UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Análisis y aplicaciones sobre letras musicales del Río de la Plata

Andrés Ferraro Paolino

Tesis de Maestría en Informática
Instituto de Computación (InCo), Facultad de Ingeniería
PEDECIBA UdelaR

Montevideo – Uruguay
Setiembre de 2018

UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Análisis y aplicaciones sobre letras musicales del Río de la Plata

Andrés Ferraro Paolino

Tesis de Maestría en Informática presentada al Programa Maestría en Informática de PEDECIBA UdelaR, como parte de los requisitos necesarios para la obtención del título de Magíster en Ingeniería en Informática.

Directores de Tesis:
Guillermo Moncecchi
Pablo Cancela

Director Académico:
Guillermo Moncecchi

Montevideo – Uruguay
Setiembre de 2018

Ferraro Paolino, Andrés

Análisis y aplicaciones sobre letras musicales del Río de la Plata / Andrés Ferraro Paolino. - Montevideo: Universidad de la República, Facultad de Ingeniería, PEDECIBA Informática, 2018.

XII, 85 p. 29, 7cm.

Directores de Tesis:

Guillermo Moncecchi

Pablo Cancela

Director académico:

Guillermo Moncecchi

Tesis de Maestría en Informática – PEDECIBA UdelaR.

Referencias bibliográficas: p. 71 – 78.

1. Procesamiento del Lenguaje Natural, 2. Aprendizaje Automático, 3. Recuperación de Información Musical, 4. Redes Neuronales Profundas, 5. Representación Distribuida. I. Moncecchi, Guillermo, Cancela, Pablo, . II. PEDECIBA UdelaR, Programa de Maestría en Informática. III. Título.

INTEGRANTES DEL TRIBUNAL DE DEFENSA DE TESIS

Dr. Horacio Saggion

Dra. Lorena Etcheverry

Dr. Martín Rocamora

Montevideo – Uruguay
Setiembre de 2018

Para Caterina y Agustina...

Agradecimientos

Es difícil resumir aquí todas las personas que hicieron posible este trabajo. En primer lugar, quisiera agradecer mis tutores: Pablo Cancela y Guillermo Moncecchi, por todos los consejos y la ayuda que me dieron durante estos dos años. También quisiera agradecer al tribunal por el valioso aporte, un especial agradecimiento a Martín Rocamora por ayudarme en distintos momentos de esta maestría y por estar siempre disponible para darme consejos muy valiosos, sin duda sin su apoyo este trabajo no hubiera sido posible. A Juan Piquinela por su dedicación y su aporte tan importante. También quisiera agradecer a Xavier Serrra por darme la oportunidad de trabajar en MTG y al resto de mis compañeros por el apoyo durante estos años. Le agradezco a Sergio Oramas por todos los aportes que hizo a este trabajo, sus consejos fueron muy importantes para desarrollar esta tesis, espero en algún momento volver a trabajando juntos. También le agradezco a Aiala Rosá, Dina Wonsever y Pablo Zinemanas por los valiosos aportes. A María Inés Sanchez por su constante ayuda en la parte administrativa. Kjell Lemstrom por permitirme realizar la pasantía en Helsinki. También le agradezco a mi esposa Agustina por acompañarme incondicionalmente, por apoyarme en las decisiones más difíciles y por ayudarme en todo momento. Este trabajo no hubiera sido posible sin la ayuda de mi familia. A Fernando, Charo y toda la familia, por apoyarnos siempre. Por último pero no menos importante gracias a Santiago, Francisco y Matías por estar siempre.

La investigación que da origen a los resultados presentados en la presente publicación recibió fondos de la Agencia Nacional de Investigación e Innovación bajo el código POS_NAC_2016_1_130162.

RESUMEN

A partir del año 2015 el consumo de música por medio de soportes digitales superó el consumo en medios físicos, en gran parte debido al creciente uso de los servicios de música *online*. Dado el tamaño de las colecciones musicales que manejan estos servicios, es importante contar con sistemas capaces de categorizar y recomendar la música de forma automática. Es por esto que desde hace varios años se viene investigando sobre la clasificación y recomendación automática de canciones en el área de la recuperación de información musical, ya sea a partir de su contenido o de su contexto.

En este trabajo se analizan distintas técnicas para representar las canciones basadas únicamente en sus letras, con el objetivo de encontrar aquella que permita obtener una representación que mantenga la similitud entre las letras de las canciones. Para evaluar el desempeño se utilizan las representaciones en distintas tareas de clasificación.

Una parte importante de esta tesis es la recolección del corpus de datos utilizados, ya que el trabajo se enfoca en algunos géneros musicales relacionados con el Río de la Plata que actualmente no cuentan con un conjunto de datos para trabajar (por ejemplo: Tango, Milonga o Candombe). No hay muchos trabajos anteriores que se enfoquen en letras de canciones en español, por lo tanto, decidimos ofrecer parte del conjunto de datos abiertamente para favorecer la investigación sobre estos géneros musicales.

Una vez obtenido el corpus de datos, es explorado mediante algunas técnicas basadas en aprendizaje no supervisado. Luego, se profundiza en técnicas basadas en redes neuronales recurrentes para obtener las representaciones de las letras. Por último, se utilizan las representaciones obtenidas para calcular similitud entre ellas, permitiendo explorar estos resultados a través de una interfaz web. Mediante la misma interfaz, también se puede navegar por el conjunto de datos recolectado, siendo un aporte para todos las personas que

deseen conocer más sobre estos géneros musicales.

Palabras claves:
Procesamiento del Lenguaje Natural, Aprendizaje Automático, Recuperación de Información Musical, Redes Neuronales Profundas, Representación Distribuida.

ABSTRACT

Since the year 2015 music consumption in digital format surpassed physical formats for the first time, this is primarily because of the increased use of streaming services. Given the size of the music collections that offers this services, is very important to count with systems that allows to automatically categorize and recommend music. This is the reason why a lot of research has been done in the previous years in the field of Music Information Retrieval, related with automatic classification and recommendation based on song content and context.

In this work, we analyze different technics for song representation only based on their lyrics, with the goal of identifying the method that allows to keep the most important aspects of the song lyrics. To evaluate the performance we use these representations in multiple classification tasks.

An important part of this work is the gatering and clean of the corpus since the focus of this work is in different musical genres from the River Plate, where currently there is not available any dataset of lyrics. Also, not many works focus on lyrics in spanish language, therefore we decided to publicly share part of the corpus in order to favor the research related to these musical genres.

First, we explore the corpus using some unsupervised technics. Later, we focus on deep learning technics to genereate representations of lyrics. Finally, we use the representations to calculate the similarity between the songs, we built a web interface that allows exploring this results. Moreover, this interface allows to navigate the collected corpus, we think that this could be a good contribution for the people interested in this music genres.

Keywords:
Natural Language Processing, Machine Learning, Music Information Retrieval, Deep Neural Networks, Representation Learning.

Tabla de contenidos

Capítulo 1

Introducción

La introducción de soportes digitales para el consumo de música significó una gran revolución en la industria. Inicialmente, la única forma de utilizar este soporte digital consistía en descargar la música desde plataformas específicas para compartir archivos de audio, por lo que cada usuario necesariamente debía contar con su propia colección de música. Más tarde, con la introducción de los servicios de música en línea o por medio de *streaming* de datos, se pudo acceder a todo el catálogo y consumirlo directamente por medio de Internet, sin ser necesario almacenar las canciones para poder consumirlas.

Como muestra el informe publicado por la Federación Internacional de la Industria Fonográfica (IFPI, 2016), a partir del año 2015 el ingreso generado por el consumo de música a través de medios digitales superó al generado a partir de medios físicos. Además, en el informe se indica una reducción del consumo de música mediante descargas y un gran incremento del 93 % en el consumo mediante *streaming* con respecto al año anterior.

A partir de los datos recién mencionados, podemos ver la importancia que toman los servicios de *streaming* en la industria y, por lo tanto, la necesidad de contar con sistemas que permitan manejar de forma eficiente su contenido, ya sea video o audio; también incluyendo a toda la información relacionada con las canciones y los artistas.

El cambio en la forma de consumo hacia servicios de *streaming* también introdujo una nueva problemática. Según Schwartz (2003), cuando se incrementa el número de opciones que se le ofrecen a una persona se puede influir negativamente en la experiencia como consumidor. Por lo tanto, es necesario acotar las opciones, por ejemplo, ofreciendo un adecuado número de recomendaciones

que sean de su agrado (Bollen et al., 2010).

Cuando se intenta dar recomendaciones al usuario se encuentran una serie de dificultades; el sistema de recomendación debe ser capaz de identificar cuáles ítems pueden ser del agrado del usuario, al mismo tiempo le debe ofrecer un balance adecuado entre los elementos que generalmente consume y elementos nuevos que nunca consumió. Además, se debe tener en cuenta que existen elementos más populares que otros. Esto quiere decir que existe un sesgo en los datos relacionados a los elementos consumidos por los usuarios: para los elementos menos populares será más difícil identificar si los usuarios desearán consumirlos o no.

1.1. Recuperación de información musical

La recuperación de información musical (MIR) es el área donde, desde hace varios años, se viene investigando sobre la clasificación y recomendación automática de canciones (entre otros temas). En un principio los enfoques utilizados eran principalmente basados en el análisis del audio, y se los conoce como basados en el contenido. Luego, se comenzó a utilizar otro tipo de información (letras de canciones, imágenes, biografías de artistas o partituras); a estos enfoques se los conoce como basados en contexto. Aprovechando la creciente cantidad de datos disponible por medio de distintas fuentes en Internet, en MIR se combinan cada vez más fuentes del contexto, con el objetivo de mejorar los resultados en las distintas aplicaciones.

En MIR el género musical se entiende como un concepto de alto nivel y es utilizado para clasificar la música según su similitud. Es decir, canciones dentro de un mismo género musical tienen una mayor similitud que canciones de distintos géneros musicales.

Uno de los temas más estudiados en MIR es la clasificación automática de géneros musicales. Distintos tipos de datos se utilizaron para detectar el género de canciones, ya sea audio (Fu et al., 2011) o también otros formatos como partituras o letras. Una importante revisión de más de 500 trabajos sobre este tema fue realizada por Sturm (2014). Como se detalla más adelante, se ha mostrado que la combinación de datos de distinta naturaleza (por ejemplo, al combinar audio con letras de canciones), permite mejorar la clasificación.

Dentro de MIR, en un principio los trabajos se enfocaron en géneros más populares como el Rock, Jazz y Blues (Tzanetakis and Cook, 2002; Mayer

et al., 2008a). Últimamente, se comenzó a investigar otros estilos de música tradicionales de algunos países. Por ejemplo, se publicaron conjuntos de datos del Flamenco (Kroher et al., 2016; Oramas et al., 2015a), o también música tradicional de China (Repetto and Serra, 2014), Turquía (Uyar et al., 2014), India (Srinivasamurthy et al., 2014). Estos trabajos dentro de MIR pueden tener un impacto en la musicología, ya que se puede utilizar esta información para analizar automáticamente los distintos estilos musicales. Por ejemplo, se puede utilizar la información disponible en Internet para analizar relaciones y similitudes entre distintos géneros a una escala que no es posible realizar manualmente.

1.2. Objetivos del trabajo

Aunque Internet ofrece la posibilidad de acceder a muchas fuentes y tipos de información, este trabajo se enfoca únicamente en letras de canciones ya que entendemos que es un problema lo suficientemente amplio en sí mismo.

El objetivo de este trabajo es la investigación de algunas representaciones de letras en espacios de baja dimensión que mantengan la similitud entre las canciones. Como forma de evaluar la calidad de las representaciones, se evalúa su desempeño en diferentes tareas de clasificación.

Se entiende que una representación de las letras que mantenga la similitud entre las canciones podría ser utilizada en combinación con otras representaciones obtenidas a partir de otros datos y de esta forma se podría obtener una representación más completa, es decir, que contemple más aspectos de las canciones.

No existen muchos trabajos que tengan como foco géneros musicales relacionados con el Río de la Plata, debido a la falta de datos disponibles que permitan realizar la investigación. Entonces, este trabajo tiene como objetivo realizar un aporte a la investigación de la música rioplatense.

En esta tesis, cuando se refiere a géneros relacionados al Río de la Plata principalmente se refiere al tango, pero también en menor proporción a otros como la milonga o el candombe.

1.3. Estructura de la tesis

En el siguiente capítulo se describen algunos trabajos relacionados, ya sea porque se enfocan en el uso de letras de canciones o porque aplican técnicas del procesamiento del lenguaje sobre otros textos relacionados a la música. También se describen algunas técnicas para obtener representaciones a partir del texto basadas en aprendizaje profundo, en algunos casos relacionadas a la música y otros no.

En el Capítulo 3 se describe el conjunto de datos construido a partir de fuentes de Internet, incluyendo la descripción del proceso por el cual se obtuvo. También allí se realizan distintos análisis sobre el corpus de letras. En primer lugar, se analiza la estructura de las letras. Luego, se utiliza una representación simple para la visualización y el agrupamiento de las letras mediante un enfoque no supervisado.

A continuación, el Capítulo 4 describe la aplicación de una técnica basada en redes neuronales recurrentes para representar las letras de canciones. Como resultado adicional, al ser modelos generativos estas redes también permiten generar nuevas letras.

Luego, en el Capítulo 5 se evalúan los modelos construidos mediante diferentes tareas de clasificación.

El Capítulo 6 muestra una aplicación para las representaciones obtenidas. Con esta aplicación se puede navegar por el conjunto de datos recolectado y se puede interactuar con las representaciones a partir de las canciones más similares.

Por último, en el Capítulo 7 se mencionan las conclusiones principales y el trabajo a futuro.

1.4. Contribuciones principales

En resumen, este trabajo se enfoca en identificar las representaciones más adecuadas para las letras de las canciones en el contexto antes mencionado. Una de las contribuciones es el desarrollo de un método para obtener las representaciones de las letras a partir de redes recurrentes. Las representaciones obtenidas se podrían combinar con otros datos para mejorar distintas tareas. Por ejemplo, sistemas de recomendación, recuperación de información en una base de datos musical, la navegación en una base de datos musical, entre otras.

Se contribuye al área del MIR mediante la publicación de un conjunto de datos que facilita la investigación sobre géneros musicales del Río de la Plata, dentro de lo permitido por restricciones en los derechos de distribución de las letras.

También se realiza una contribución mediante el desarrollo y publicación de un sistema web que permite, de forma intuitiva, acceder a contenidos de estos géneros musicales, teniendo un impacto directo en la preservación del bien inmaterial.

Capítulo 2

Trabajos relacionados

En este capítulo se realiza una revisión de algunas publicaciones que están relacionadas de alguna forma con los objetivos de este trabajo.

En primer lugar, se describen algunas técnicas básicas para representar documentos de texto en forma vectorial ya que serán mencionadas a lo largo de toda la tesis.

Luego se realiza un repaso sobre los conjuntos de datos disponibles relacionados con música del Río de la Plata. A continuación, se detallan algunos trabajos enfocados en la clasificación de canciones basada en sus letras. También se mencionan trabajos que utilizan técnicas del procesamiento del lenguaje natural sobre texto extraído de Internet relacionado a los artistas y las canciones.

Por último, se describe un conjunto de trabajos enfocados en aprendizaje profundo y sobre el uso de representaciones distribuidas en el área del procesamiento del lenguaje natural.

2.1. Técnicas básicas para representación vectorial de documentos

Existen distintas formas de obtener una representación vectorial de documentos que permita compararlos. Un ejemplo es el método basado en bolsa de palabras, el cual consiste en representar cada documento como un vector a partir de la importancia de cada palabra presente en el documento, sin tener en cuenta el orden en que aparecen (Jurafsky and Martin, 2009).

La forma de determinar la importancia de cada palabra en el documento se

puede calcular a partir de la cantidad de ocurrencias de cada palabra. También se pueden utilizar otras medidas para determinar el peso como *Term Frequency Inverse Document Frequency* (TF-IDF):

$$\text{tf-idf}_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t} \tag{2.1}$$

Donde $\text{tf}_{t,d}$ es el número de veces en que el término t aparece en el documento d y df_t es el número de documentos en los que el término t ocurre.

Además de las palabras del texto, en ocasiones se utiliza el etiquetado gramatical de cada palabra; de esta forma se puede desambiguar diferentes usos de una misma palabra.

También es frecuente el uso de *Stemming* y de Lematización con el objetivo de normalizar textos. *Stemming* se refiere a la reducción de cada palabra a su raíz, mientras que Lematización se refiere a la transformación de cada palabra en su forma canónica (por ejemplo, las palabras "perros", "perra" y "perro" tienen como forma canónica el lema "perro"). Se puede aplicar bolsa de palabras sobre los lemas o las raíces, lo que permite unificar diferentes palabras en una misma, permitiendo identificar documentos que están relacionados.

También vale la pena mencionar la combinación de representaciones distribuidas de las palabras (*word embeddings*). A diferencia de las representaciones antes mencionadas, *word embeddings* es una representación densa de menor dimensión para las palabras, mediante vectores de números reales. Los *word embeddings* se pueden combinar para representar un documento aplicando alguna operación sobre los vectores, comúnmente se utiliza la concatenación o también la suma y el promedio.

En la Figura 2.1 se comparan dos representaciones que se podrían obtener para un documento, una a partir de *word embeddings* y otra basada en bolsa de palabras (obtenida de la frecuencia de las palabras en el documento). Por lo tanto, se puede ver que es posible representar el mismo documento en una menor dimensión. La desventaja de tener una representación de menor dimensión es que no es posible, de forma intuitiva, identificar las palabras que componen el documento a partir de la representación. En la Sección 2.5 se profundiza sobre la forma en que se calculan los *word embeddings* de las palabras, también se mencionan distintas formas de extender el concepto de *word embeddings* para entrenar representaciones de oraciones o documentos.

Una vez que los documentos se encuentran en una representación vectorial

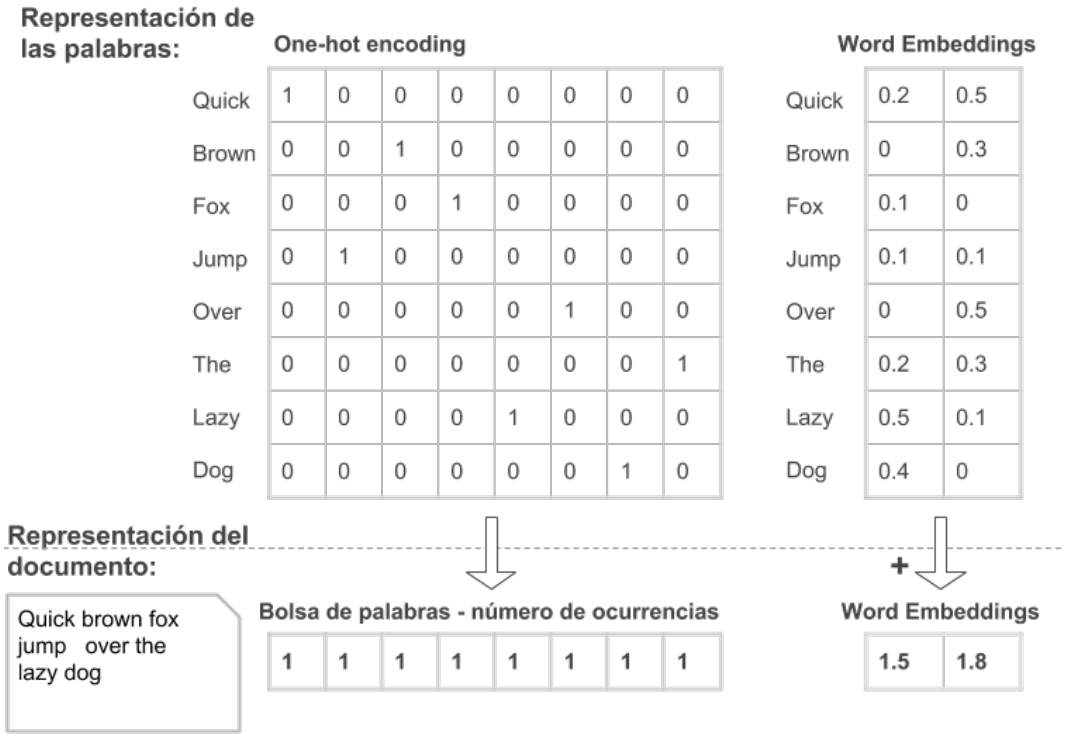

Figura 2.1: Representación de las palabras mediante *one-hot-encoding* en comparación a *word embeddings*. Se combinan los vectores de las palabras para obtener una representación del documento mediante la operación suma.

es posible compararlos. Por lo tanto, utilizando las representaciones vectoriales de los documentos es posible identificar los más similares o también agruparlos en función de los temas que tratan.

2.2. Conjuntos de datos sobre música de América Latina

Para poder realizar este trabajo es necesario contar con un conjunto de datos que contenga letras de canciones relacionadas al problema antes mencionado. Por lo tanto, en esta sección se realiza un repaso por los conjuntos de datos disponibles relacionados con música del Río de la Plata, ya sea que su contenido es audio o texto, debido a que en ambos casos también contienen metadatos referentes a las canciones o sus artistas.

Al detallar los conjuntos de datos disponibles más relevantes para esta tesis, se puede ver que no hay un conjunto de datos que sea lo suficientemente grande para poder aplicar las técnicas deseadas y al mismo tiempo representar

la realidad del problema antes mencionado. Es por esto que se debió conformar un conjunto de datos propio. Por lo tanto, en esta sección también se mencionan algunas publicaciones que describen este proceso.

Tal vez el conjunto de datos más relevante para esta tesis (debido a que está relacionado a música latinoamericana), se llama The Latin Music Database (Silla Jr et al., 2008). Este conjunto de datos contiene un total de 3.227 grabaciones en MP3 manualmente anotadas en los siguientes géneros : axe, bachata, bolero, forro, gaucha, merengue, pagode, salsa, sertaneja y tango. De tango contiene 408 canciones de un total de 19 horas correspondientes a siete artistas distintos.

Con respecto al conjunto de datos The Latin Music Database, vale la pena destacar algunas características que son mencionadas en un trabajo que lo utiliza para clasificación de género (Esparza et al., 2015). Los autores mencionan que en este conjunto de datos el tango tiene muy pocos artistas (solamente siete). También mencionan que la mayoría de las grabaciones de tango son de Gardel, lo que quiere decir que son grabaciones de 1930 aproximadamente. Esto hace que la tarea de clasificar tango sea más fácil en comparación con los otros géneros, ya que la representación de tango es más homogénea (por ejemplo en el rango de frecuencias usado). A partir del problema recién mencionado, podemos ver que al momento de conformar un conjunto de datos es necesario tener en cuenta una mayor variedad de canciones y artistas de forma de representar mejor la realidad.

The Latin Music Mood Database (LMMD) (dos Santos and Silla, 2015) es una extensión del conjunto de datos The Latin Music Database con anotaciones de emociones sobre las canciones, agregando información de 6 emociones a cada grabación. En LMMD, el tango cuenta con el 56 % de anotaciones dentro de la categoría *decepción*. Por otro lado, dentro de las anotaciones de tango el sentimiento más seleccionado es *pasión*(23 %) y luego *tristeza*(22 %).

Otro conjunto de datos muy utilizado en el área es el Ballroom Dataset, creado por Gouyon et al. (2004). Este conjunto de datos contiene 698 archivos de audio de los géneros: *cha cha*, *jive*, *quickstep*, *rumba*, *samba*, *tango*, *viennese waltz* y *slow waltz*, contando con 86 grabaciones de tango. Este conjunto de datos fue extendido por Marchand and Peeters (2016) con más audios y los siguientes géneros: pasodoble, salsa, *slow waltz* y *wcswing*. El conjunto de datos extendido contiene un total de 4180 archivos de audio de canciones donde 464 corresponden al género tango.

Howard et al. (2011) describe la creación de un conjunto de letras de canciones asociadas a The Latin Music Database. El conjunto presenta un total de 500 letras, en donde 50 corresponden a tango. Por lo tanto, el tamaño no es suficientemente grande para poder realizar las tareas de clasificación que se plantean en esta tesis.

En general, es difícil encontrar conjuntos de datos abiertos con letras de canciones debido a limitaciones relacionadas a los derechos para distribuirlas. En la web del conjunto de datos The musiXmatch Dataset [1] mencionan este problema; la forma en que los autores logran evadirlo es ofreciendo las letras en formato bolsa de palabras (sobre *stems* de las palabras). El conjunto de datos The musiXmatch Dataset contiene un total de 237.662 letras relacionadas con el conjunto de datos Million Song Dataset (Bertin-Mahieux et al., 2011).

Otro conjunto de letras es el LyricFind Corpus (Ellis et al., 2015) el cual ofrece un total de 275.905 letras también en formato Bolsa de palabras.

El trabajo de Ribeiro et al. (2014) describe un mecanismo para detectar y extraer letras de canciones a partir de la web. Para evaluar el sistema se utiliza el conjunto de datos The Latin Music Database con el objetivo de encontrar las letras relacionadas a las canciones.

2.3. Clasificación de canciones basadas en las letras

En esta sección se detallan algunos trabajos enfocados en la clasificación de canciones utilizando sus letras. Estas publicaciones son relevantes para nuestro trabajo, en primer lugar, porque una parte de esta tesis se trata de clasificar las canciones; en segundo lugar, para mostrar la forma en que estos trabajos extraen una representación de las letras.

También se mencionan algunos trabajos que tienen como objetivo la detección de autor, lo cual es un caso particular de clasificación.

Es importante destacar que la tareas de detección de autores y clasificación de textos se aplican también en otros contextos (por ejemplo: en poesía y libros de texto). En algunos casos las técnicas pueden resultar útiles para el contexto de la música, pero en otros casos no. Por lo tanto, es necesario evaluar cada método en el contexto deseado.

[1] https://labrosa.ee.columbia.edu/millionsong/musixmatch

Li and Ogihara (2004) realizan uno de los primeros trabajos en que se aplican técnicas de procesamiento del lenguaje sobre letras de canciones con el objetivo de identificar el autor. Este trabajo solo utiliza la música de 43 artistas, todos ellos intérpretes en inglés. En este trabajo, se compara el enfoque basado en las letras con otro basado en el audio de las canciones y proponen una combinación de ambos métodos. Algunos de los atributos basados en las letras utilizados son: presencia de ciertas palabras (seleccionadas a partir de la medida tf-idf) y estadísticas de las letras (por ejemplo, largo de las palabras). También utilizan otros atributos extraídos a partir del etiquetado gramatical (artículos, preposiciones y pronombres), dado que había sido utilizado en otros contextos. Obtienen una precisión del 64 % solamente utilizando las letras y 78 % combinando las letras con el audio.

También se puede considerar dentro de los primeros trabajos en que se aplican técnicas de procesamiento del lenguaje sobre letras de canciones el de Mahedero et al. (2005), en donde los autores hacen experimentos en 4 tareas distintas:

Identificación del lenguaje Para identificar el lenguaje prueban hacerlo solo con los títulos y obtienen un 75 % de acierto, mientras que utilizando todas las letras obtienen un 92 %.

Extracción de estructura Para identificar la estructura dividen las letras en secciones y calculan la similitud entre ellas; a continuación se identifica la parte más repetida de la letra y finalmente se desambiguan los casos más dudosos. Obtienen un 75 % de resultados correctos.

Categorización por temática Para categorizar según temáticas utilizan un clasificador basado en Naive Bayes, las categorías son: *Love*, *Violent*, *Protest (antiwar)*, *Christian* y *Drugs*, donde obtienen un 82 % de precisión.

Búsqueda por similitud Para calcular la similitud utilizan distancia coseno, representado por medio de un vector cada letra basado en tf-idf. Para evaluar los resultados hacen tres pruebas: una es buscar canciones que sean una versión de la misma, la segunda prueba consiste en buscar otra versión pero del mismo cantante y la tercera es buscar una canción que no esté. Para la primera prueba obtienen un 98 % de relevancia en los resultados, en la segunda prueba obtienen 82 % y en la tercera 62 %.

Además, en el trabajo de Mahedero et al. (2005) se mencionan algunas dificultades encontradas según las distintas tareas. Por ejemplo, uno de los problemas

que detectan para el caso de la Identificación del lenguaje es que si hay palabras inventadas puede confundir el clasificador. Otro problema sucede para la tarea Categorización por temática: cuando hay frases o palabras que se usan en muchos contextos distintos, se dificulta la tarea del clasificador.

Aunque con un objetivo diferente, Mayer et al. (2008b) utiliza varios de los atributos ya mencionados en el trabajo de Li and Ogihara (2004). Mayer et al. (2008b) utiliza la combinación de un conjunto de atributos extraídos de la rima de las letras, estadísticas de las letras (por ejemplo: cantidad de caracteres por palabra), medidas tf-idf de las palabras y otras extraídas a partir del etiquetado gramatical. En este trabajo se estudia la clasificación en 10 géneros de 397 canciones en inglés seleccionadas a partir de un conjunto de 12.000 letras extraídas de Internet. El mejor resultado obtenido tiene una precisión del 33 % en donde un clasificador aleatorio obtendría 11 % de precisión.

También en el trabajo de Hirjee and Brown (2010) se evalúa el uso de distintos atributos basados en la rima con varios objetivos. Algunos de los objetivos son: clasificación de autores y clasificación de género. Para el caso de clasificación de autores, se obtiene un 56 % de precisión al clasificar las canciones para el caso del *rap*. Aplicando el mismo método sobre un conjunto de canciones de *pop* obtienen un 26 %, por lo que muestran que los atributos seleccionados funcionan mejor en el caso del *rap*.

Basados en algunas de las publicaciones mencionadas anteriormente, Fell and Sporleder (2014) realizan tres experimentos a partir de las letras. El primer experimento se trata de clasificar las letras según su género; en el segundo experimento se intenta predecir la valoración de los usuarios sobre la canción y en el tercer experimento se clasifican las canciones en tres categorías según su fecha de publicación. Para evaluar las distintas tareas, los autores construyen un conjunto de datos propio con 400.000 canciones en inglés. En todas las tareas utilizan un clasificador basado en SVM sobre un conjunto de 13 atributos extraídos de las letras en combinación con n-gramas a nivel de palabras (n = 1 y 2). Los autores concluyen que en todos los experimentos se mejoran los resultados al combinar los n-gramas con los atributos originales, en comparación a utilizar únicamente los n-gramas.

2.3.1. Clasificación basada en letras de canciones en combinación con audio

A continuación, se mencionan algunos trabajos similares a los anteriores ya que se basan en letras para realizar clasificación, con la diferencia que además utilizan el audio de las canciones.

Mayer et al. (2008a) realizan clasificación por género de canciones en inglés a partir de la combinación del audio y las letras. Los atributos utilizados en este trabajo se encuentran dentro de los mencionados previamente, como estadísticas sobre el etiquetado gramatical o de ciertas palabras y caracteres particulares, además de la identificación de estructuras rítmicas y atributos de las palabras. En este trabajo, los autores concluyen que la combinación de atributos extraídos de audio y letras permite obtener mejores resultados que utilizando solamente atributos de audio o del texto.

El trabajo de McKay et al. (2010) combina atributos extraídos de audio, letras de canciones, contexto cultural y partituras con el objetivo de clasificar canciones en 10 géneros. En este trabajo se menciona que los atributos extraídos de las letras fueron los que menos contribuyeron a la clasificación, aunque esto es posiblemente debido a que un porcentaje importante de las canciones corresponde a música instrumental.

Los siguientes cuatro trabajos están enfocados en la detección de sentimiento en canciones a partir de la combinación de audio y letras, en conjuntos de datos en inglés. Laurier et al. (2008) utiliza 3 representaciones distintas a partir de las letras. En primer lugar, utilizando una representación basada en bolsa de palabras. En segundo lugar, basada en una representación a partir del método Latent Semantic Analysis (LSA) (Dumais et al., 1988). Por último, basado en las diferencias a partir de distintos modelos del lenguaje. Los trabajos de Hu (Hu et al., 2009; Hu and Downie, 2010a,b) utilizan atributos seleccionados manualmente, algunos ya propuestos por Mayer et al. (2008a) y otros nuevos, como *features* extraídos a partir de un lexicón que previamente se demostró que eran útiles a partir de estudios psicolingüísticos.

En todas las publicaciones anteriormente mencionadas que comparan el uso de audio con letras de canciones en tareas de clasificación, se concluye que la combinación de ambas fuentes produce mejores resultados que el uso únicamente del audio.

Un enfoque diferente es utilizado por Logan et al. (2004), en donde se aplica

LSA probabilístico (Hofmann, 1999) sobre una colección de 15.589 letras de canciones correspondientes a 399 artistas. Esta técnica permite obtener una representación vectorial de cada letra, donde cada dimensión corresponde a la probabilidad de pertenecer a un tópico preentrenado con otro conjunto de datos; en este caso, los tópicos se corresponden con géneros musicales. Las representaciones son evaluadas a nivel de artistas comparándolas contra los resultados de encuestas a usuarios. También se utiliza un método basado en audio para calcular las similitud entre artistas. Para la mayoría de los géneros funciona mejor el método basado en audio, excepto para el caso de música latina, donde funciona mejor el basado en las letras. Los autores mencionan que es probable que las respuestas de los usuarios estén sesgadas a identificar como similares canciones que se escuchan parecidas y no tanto según el contenido de las letras.

2.3.2. Agrupamiento no supervisado de letras

Vale mencionar algunos trabajos en los que se realiza clasificación de canciones sobre las letras basado en métodos de aprendizaje no supervisado

El trabajo realizado por Parra and León (2013) tiene como objetivo organizar y etiquetar un conjunto canciones a partir de su letra. Para esto se utiliza una representación similar a la utilizada por Mayer et al. (2008b) en combinación con el uso de etiquetado gramatical. A partir de las representaciones se agrupan las canciones utilizando K-means. Para definir el número de grupos, Parra and León (2013) se basa en distintas medidas de calidad, como *Davies Bouldin Index* (Davies and Bouldin, 1979). Finalmente, se asigna un conjunto de etiquetas a cada grupo de canciones a partir de las palabras más frecuentes dentro de cada grupo. La publicación de Parra and León (2013) es la única encontrada en la que se utiliza exclusivamente un conjunto de letras de canciones en Castellano.

Uno de los problemas que presentan los métodos que asignan categorías a las letras de forma no supervisada es que es muy difícil de interpretar los resultados. Sterckx et al. (2014) trata este problema buscando aquella medida que ofrece la mayor correlación entre las categorías asignadas automáticamente con categorías anotadas manualmente.

2.4. Procesamiento del lenguaje aplicado a otros textos relacionados a la música

En la sección anterior se mencionaron publicaciones que utilizan letras de canciones. En esta sección se analizan trabajos en los que se utilizan otros textos relacionados a la música con objetivos similares (por ejemplo: clasificación, cálculo de similitud o recomendación de canciones o artistas). Si bien los datos utilizados por estas publicaciones son distintos a los utilizados en nuestro trabajo, todas aplican técnicas del procesamiento del lenguaje que también se podrían aplicar sobre nuestro problema.

El trabajo de Whitman and Lawrence (2002) fue uno de los primeros en aplicar técnicas del procesamiento del lenguaje natural sobre texto relacionado a la música. Allí se muestra cómo, utilizando texto no estructurado extraído de Internet, se puede obtener una representación de los artistas que permite mantener la similitud. Los autores utilizan n-gramas, etiquetado gramatical y sintagmas nominales para seleccionar los términos que mejor identifican a cada artista. Luego, calculan la similitud entre los artistas en función de los términos que tienen en común.

También en el año 2002, el trabajo de Whitman and Smaragdis (2002) combina información del audio con información cultural extraída de un catálogo *online* de información musical (Allmusic[1]), con el objetivo de clasificar artistas por estilo musical. Los autores muestran que al utilizar solo el audio hay algunos estilos que no se clasifican bien y cuando usan solo el texto son otros estilos distintos los que no se clasifican bien. Luego, cuando combinan audio y texto obtienen una precisión alta. El método utilizado sobre el texto consiste en seleccionar los términos que mejor identifican a los artistas, luego se agrupan los artistas de forma no supervisada y los grupos definen los estilos musicales.

Un enfoque similar a los trabajos anteriores es utilizado por Knees et al. (2004b,a). En este trabajo, se extrae el texto de todas las web retornadas por los buscadores Google y Yahoo a partir de la consulta de los artistas (con algunos términos particulares). Aplicando tf-idf sobre texto obtenido, se construye un clasificador de género. En uno de estos trabajos se muestra que el método es superior al de Whitman and Smaragdis (2002) y en el otro trabajo se muestra

[1] http://allmusic.com

una forma de medir la similitud a partir de los valores obtenidos mediante Tf-Idf.

Schedl et al. (2005) mide la similitud entre los artistas a partir de la cantidad de documentos retornados por motores de búsqueda. En este trabajo, se evalúan distintas formas de conformar las consultas. Para la evaluación, se mide el desempeño obtenido al clasificar los artistas con respecto a un conjunto de anotaciones de género musical.

En el trabajo de Pohle et al. (2007), primero se obtiene un vector a partir de la descripción de los artistas utilizando tf-idf. Luego, se reduce la dimensión de los vectores a 16 mediante el método Non-negative matrix factorization (NMF). De esta forma, cada una de las 16 dimensiones están asociadas a un conjunto de palabras originales en el vector obtenido con tf-idf. Por lo tanto, para cada artista se obtiene una representación indicando en cada dimensión si presenta o no la característica correspondiente. Al analizar los grupos de palabras que están asociados a cada una de las 16 categorías, los autores concluyen que son similares a agrupaciones por género.

Schedl (2010) aplica un enfoque similar a los trabajos basados en motores de búsqueda, con la diferencia que realiza búsqueda de *tweets* relacionados a los artistas. Los autores aplican tf-idf sobre los *tweets* y utilizan K-NN para agrupar los artistas. Luego, comparan contra un conjunto de anotaciones de géneros los grupos obtenidos. Además, se evalúa este método tomando las palabras más relevantes para cada artista y comparándolas contra un conjunto de etiquetas obtenidas a partir de un servicio llamado Last.fm[1], las cuales son ingresadas por usuarios.

2.4.1. Extracción de información y reconocimiento de entidades

También es relevante mencionar algunos trabajos que aplican métodos similares a los anteriores, con el objetivo de identificar relaciones entre las diferentes entidades musicales (por ejemplo: artistas, instrumentos, bandas, lugares, etc). Una vez identificadas estas relaciones, se pueden construir bases de conocimiento que permiten mejorar los sistemas de recuperación de información o también sistemas recomendación.

Los primeros trabajos relacionados a la música que aplican técnicas de

[1] http://last.fm

reconocimiento de entidades sobre textos extraídos de Internet, tienen como objetivo extraer cierta información específica, por ejemplo :

- Extraer el lugar de origen de un artista (Govaerts and Duval, 2009).
- Identificar aquellos artistas que son miembros de la banda (Schedl and Widmer, 2008).
- Identificar los miembros de una banda y la discografía de los artistas (Knees and Schedl, 2011).

Otras publicaciones posteriores procesan el texto con el objetivo de obtener un grafo con relaciones entre las entidades. Luego, este grafo es utilizado para calcular la similitud entre los artistas (Oramas et al., 2015b) o para generar recomendaciones (Sordo et al., 2015).

Oramas et al. (2016b) construye una base de conocimiento de la música a partir de textos extraídos de una *web* con información sobre las canciones (Songfacts[1]). Esta base de conocimiento fue publicada por el autor para ser utilizada por otros investigadores.

Por último, en el trabajo de Oramas et al. (2016a) se utiliza la base de conocimiento recién mencionada para enriquecer un corpus de comentarios de usuarios de Amazon[2] sobre álbumes de música. El objetivo del trabajo de Oramas et al. (2016a) es clasificar los álbumes en 13 géneros musicales. Evalúan las combinaciones de los siguientes enfoques:

- Tf-idf sobre el texto de los comentarios.
- Información semántica que se obtiene de las categorías de wikipedia asociadas a los elementos identificados en los comentarios.
- *features* asociados al sentimiento extraídos de los comentarios.

Los autores destacan que se obtiene el mejor resultado combinando la información semántica con la información obtenida mediante tf-idf.

[1] http://www.songfacts.com/
[2] https://www.amazon.com/

2.5. Aprendizaje profundo y representación distribuida aplicada al procesamiento del lenguaje

En las secciones anteriores se analizaron diferentes trabajos sobre letras de canciones y otros textos relacionados a la música. En esta sección, se mencionan enfoques más recientes que podrían ser aplicados al problema planteado en esta tesis. Hasta donde sabemos, algunos de estos enfoques aún no han sido aplicados sobre letras de canciones.

Últimamente los métodos basados en aprendizaje profundo tomaron mayor interés en múltiples comunidades impulsado por resultados alentadores. Desde hace algunos años la investigación relacionada al procesamiento del lenguaje no es ajena a este fenómeno. Por lo tanto, a continuación se describen algunas técnicas más recientes del procesamiento del lenguaje natural que se podrían aplicar a nuestro problema.

La representación distribuida es un concepto introducido por Hinton (1986) el cual se refiere al mapeo de los datos originales a una representación más adecuada para aplicación de aprendizaje automático. En el procesamiento del lenguaje natural, Bengio et al. (2003) introduce la idea de usar redes neuronales con el fin de obtener representaciones distribuidas de las palabras. Mikolov et al. (2010, 2011) desarrolla un modelo del lenguaje basado en redes neuronales recurrentes, las cuales son utilizadas más tarde por el mismo autor con el fin de obtener una representación distribuida de las palabras (Mikolov et al., 2013a).

En el trabajo de Mikolov et al. (2013a) se muestra una característica interesante de las representaciones obtenidas. Los autores muestran que las representaciones mantienen ciertas relaciones semánticas de las palabras, a las que se refieren como analogías. Por ejemplo, el vector diferencia entre la representación de Montevideo y de Uruguay es similar al vector diferencia de París con Francia.

Otra representación vectorial de palabras muy utilizada es Glove (Pennington et al., 2014), la cual se calcula a partir de contar coocurrencia de palabras en un corpus.

Previo al trabajo de Mikolov, Collobert et al. (2011) presenta una arquitectura de red neuronal que permite obtener buenos resultados en varias tareas del procesamiento del lenguaje. Por ejemplo: etiquetado gramatical, reconoci-

miento de entidades con nombre y etiquetado de roles semántico.

Inspirado en los trabajos recién mencionados, algunas publicaciones recientes exploran la idea de obtener representaciones a nivel de frases, párrafos o documentos. Un ejemplo es el trabajo de Kiros et al. (2015), el cual entrena el modelo intentando predecir la próxima frase a partir de la anterior y la siguiente. De todas formas, mediante estos métodos aún no se supera el estado del arte; una posible razón es que el corpus utilizado para entrenar los modelos está compuesto por libros, por lo que el corpus utilizado en la tarea que se lo evalúa puede ser muy distinto. Otros autores, en lugar de entrenar un modelo general y obtener una representación genérica, realizan un afinamiento para la tarea en la que se quiere evaluar. En el trabajo de Hill et al. (2016) se detallan varios métodos relacionados con estos conceptos y se compara el desempeño al representar frases.

En un trabajo muy reciente, Radford et al. (2017) propone utilizar redes neuronales recurrentes para modelar el lenguaje a partir de un gran corpus de comentarios de usuarios sobre productos de Amazon. Una vez entrenada la red para generar nuevos comentarios, los autores la utilizan para obtener representaciones distribuidas de los comentarios. Luego, utilizan las representaciones en distintas tareas de clasificación. Mediante este método, no superan el estado del arte en análisis de sentimiento, pero sí muestran que su modelo se comporta mejor que otros cuando se tienen menos datos. Además, muestra que en este caso se puede detectar el sentimiento con una buena precisión solamente utilizando la salida de una neurona, lo cual también le permite a los autores, manipulando el valor de la neurona, generar comentarios positivos o negativos.

Dado que estos trabajos son muy recientes, no fue posible encontrar muchas de estas técnicas aplicadas sobre letras de canciones. Sin embargo, vale la pena mencionar algunos trabajos recientes que utilizan técnicas similares sobre textos relacionados a la música.

En el trabajo de Espinosa-Anke et al. (2017) se obtiene una representación de artistas y álbumes a partir de un corpus de biografías ya desambiguado. La representación se entrena combinando las frases relacionadas con las entidades y un grafo con las relaciones de las entidades. Se utiliza el corpus de biografías aplicando el método de Mikolov et al. (2013b), el cual tiene en cuenta información semántica a la hora de entrenar las representaciones.

Luego, Oramas et al. (2017b) utiliza redes neuronales profundas con el ob-

jetivo de recomendar artistas y canciones combinando texto con audio. Con respecto al texto, se prueban varias alternativas para representar a los artistas; unas basadas en un método similar al presentado por Espinosa-Anke et al. (2017) y la otra alternativa basada en representaciones de palabras preentrenadas en un conjunto de datos de noticias. Al final, se compara el desempeño de las diferentes combinaciones.

Por último, Oramas et al. (2017a) utiliza redes profundas con el objetivo de clasificar en 250 géneros diferentes. En este trabajo se utiliza datos de texto (comentarios de usuarios sobre álbumes), imágenes (portadas de álbumes) y audio de las canciones. Con respecto al texto, prueban dos enfoques distintos. En primer lugar, utilizan una representación vectorial para cada álbum a partir de aplicar tf-idf sobre el conjunto de documentos concatenados. En segundo lugar, utilizan un enfoque similar concatenando cada texto las categorías identificadas luego de aplicar reconocimiento de entidades y obtener las categorías de Wikipedia[1]. Finalmente, se muestra cómo el mejor desempeño se obtiene combinando todas las fuentes de información.

2.6. Discusión

A partir del análisis anterior, se puede ver que las letras de canciones son comúnmente utilizadas para las tareas de clasificación y que existen un gran número de publicaciones que evalúan distintas técnicas. De todas formas, aún no se han comparado los resultados de utilizar algunas técnicas recientes, pensadas para obtener una representación vectorial a partir del texto, sobre las letras de canciones.

También es interesante destacar que no hay muchos trabajos que utilicen letras en español. Esto puede implicar una diferencia muy importante en los resultados, debido a que en ocasiones no se aplican exactamente las mismas técnicas o no es posible encontrar un corpus tan grande para entrenar los modelos. Tal vez el conjunto de datos más similar al que se necesita en este trabajo sea el creado por Howard et al. (2011), pero el tamaño del conjunto presentado por lo autores no es muy grande y tampoco es accesible públicamente.

A partir de las conclusiones recién mencionadas, se decidió que este trabajo se basara en el método detallado por Radford et al. (2017) mediante el uso

[1] https://www.wikipedia.org/

de una red neuronal recurrente. Con la diferencia que en nuestro trabajo se utilizan letras de canciones, mientras Radford et al. (2017) utiliza reseñas de usuarios sobre productos con el objetivo final de obtener representaciones que sean útiles para distintas tareas de clasificación. Entendemos que este método es novedoso, por lo tanto, el principal objetivo de este trabajo es aplicar un enfoque nuevo para el problema planteado y compararlo con técnicas más tradicionales.

Algunas de las técnicas que se utilizan inicialmente en este trabajo están basadas en los métodos presentados por Mayer et al. (2008b) utilizando bolsa de palabras para representar las letras. Ademas, se aplican algunas técnicas de agrupamiento no supervisado de letras, las publicaciones más similares son las de Parra and León (2013) y Sterckx et al. (2014), debido al uso de letras de canciones para su agrupamiento y también debido a la discusión planteada sobre cómo evaluar la calidad de estos métodos.

Capítulo 3

Corpus de letras de canciones

En el capítulo anterior, se mencionaron los conjuntos de datos que de alguna forma están relacionados con música del Río de la Plata, mostrando que actualmente no hay ninguno disponible que contenga la información necesaria y sea lo suficientemente grande para realizar el trabajo planteado en esta tesis. Por lo tanto, fue necesaria la creación de un corpus de datos que contenga letras de canciones relacionados con el Río de la Plata. Además, es necesario que el corpus esté anotado con las etiquetas necesarias para una posterior clasificación (por ejemplo: a partir de su año, autor o género).

Algunos trabajos mencionados en el capítulo anterior, tienen como objetivo la recolección de letras de canciones de Internet (Ribeiro et al., 2014). Siguiendo un enfoque similar, se identificaron las canciones que conformarían el conjunto de datos y se obtuvieron los metadatos relacionados a partir de la web Todotango[1], para luego obtener las letras combinando distintas fuentes.

En las siguientes secciones se describe en detalle el proceso para la creación del corpus, incluyendo análisis preliminares sobre las letras con el objetivo de explorar el corpus.

En primer lugar, mediante el análisis de similitud, se muestra cómo es posible identificar repeticiones dentro de las letras de las canciones.

El segundo análisis consiste en agrupar las letras de forma no supervisada, a partir de una representación simple de las letras dentro de las analizadas en el Capítulo 2. Luego, se reduce la dimensión de las representaciones para obtener una visualización de todo el corpus de letras y los grupos calculados de forma no supervisada.

[1] http://www.todotango.com/

3.1. Metadatos y letras extraídas de Todotango

Se decidió extraer los metadatos de las canciones del sitio web Todotango. En primer lugar, debido a la cantidad y la calidad del contenido que ofrece relacionado con música del Río de la Plata. En segundo lugar, debido al tamaño de la comunidad de aficionados que lo utiliza, siendo el sitio con más tráfico[1] relacionado al género tango.

A continuación, se describe el modelo de datos definido para estructurar la información y luego se menciona la cantidad de datos obtenidos a partir de Todotango.

3.1.1. Modelo de datos

En primer lugar se definió una estructura para los datos que permitiera, por un lado, clasificar las canciones según distintos criterios, y por otro identificar relaciones entre las canciones en la aplicación web.

Los datos extraídos se estructuraron en tres entidades que son: Canción, Artista y Grabación. En la Figura 3.1 se muestra un diagrama del modelo de datos. En la Tabla 3.1 se muestra los atributos de cada entidad.

Un Artista puede ser ser una persona o una agrupación de personas (por ejemplo una orquesta). El atributo Tipo de Artista indica si se corresponde a una persona o una agrupación. La entidad Grabación se refiere a una versión de una Canción, ya que una misma Canción puede estar grabada múltiples veces por distintos artistas. Por lo tanto, la entidad Grabación también tiene un atributo que indica su género.

Tabla 3.1: Composición de los metadatos

Entidad	Atributos
Artista	Nombre, Alias, Fecha de Nacimiento, Lugar de Nacimiento, Fecha de Fallecimiento, Género, Descripción y Tipo
Grabación	Artista, Descripción, Duración, Nombre y Género
Canción	Compositores, Letristas, Fecha, Título y Género

[1] https://www.alexa.com/topsites/category/Arts/Music/Styles/R/Regional_and_Ethnic/Latin/Tango

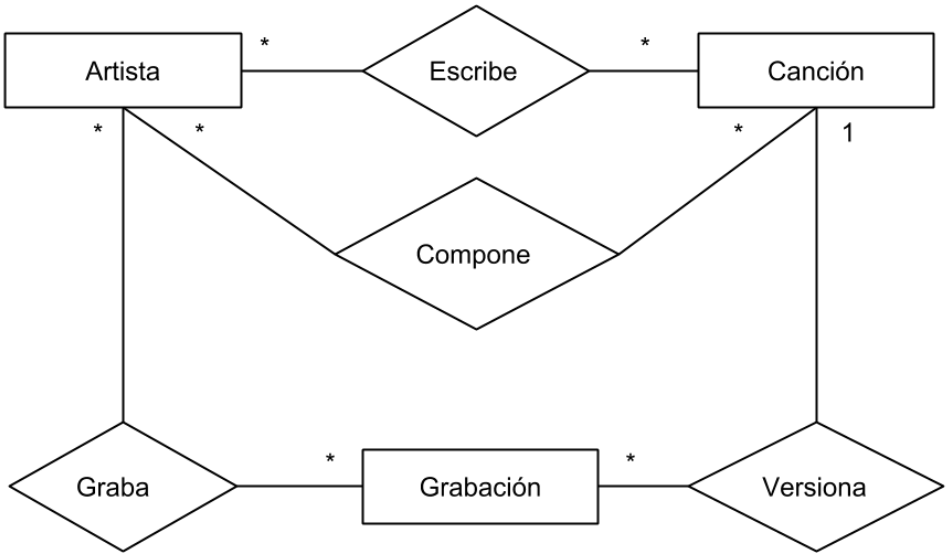

Figura 3.1: Diagrama del modelo de datos.

3.1.2. Cantidad de metadatos obtenidos

A partir de Todotango se obtuvieron un total de 9660 canciones, de las cuales 5647 contaban con su respectiva letra. No se esperaba contar con letras para todas las 9660 canciones ya que muchas son instrumentales. En total se obtuvieron 4419 Artistas; en la Tabla 3.2 se indica el número correspondiente a cada tipo. Además, se obtuvieron 7044 Grabaciones; en la Tabla 3.3 se indican los 10 géneros de canciones con mayor presencia en el conjunto de datos, siendo el género tango aquel con mayor cantidad de canciones, con un total de 6929.

Tabla 3.2: Número de Artistas según tipo.

Tipo de Artista	Cantidad de Entidades
Persona	3985
Sin tipo	377
Orquesta	26
Grupo	31

Como ya fue mencionado anteriormente, una misma canción puede estar grabada varias veces por distintos artistas y en distintos géneros. En la Tabla

Tabla 3.3: 10 Géneros de canciones con mayor presencia.

Género de Canción	Cantidad de Canciones
Tango	6929
Vals	869
Milonga	692
Canción	207
Poema lunfardo	273
Ranchera	71
Zamba	58
Candombe	53
Foxtrot	38
Estilo	33

3.4 se muestran los cinco géneros con mayor cantidad de grabaciones.

Tabla 3.4: 5 Géneros de grabaciones con mayor presencia.

Género de Grabación	Cantidad de Grabaciones
Tango	5729
Vals	511
Milonga	414
Canción	90
Arr. en tango	50
Ranchera	49

3.2. Enriquecimiento del corpus

Al analizar las letras inicialmente extraídas de Todotango, se pudo ver que había canciones sin letras y además algunas de las letras obtenidas estaban incompletas. Por ejemplo, en algunos casos secciones que se debían repetir no estaban presentes. En la Tabla 3.5 se muestra como ejemplo la letra de la canción "Volver" de Alfredo Le Pera, como fue extraída originalmente de Todotango. En este caso falta la repetición del estribillo al final de la canción.

Con el objetivo de mitigar estos problemas, se recolectaron varias versiones de letras a partir de distintas fuentes. Luego, se definió un orden de prioridad entre las fuentes, a partir de cuáles ofrecían letras más completas. Se entiende que aquella fuente que presenta las letras más completas es la de mayor calidad.

El proceso de buscar y extraer las letras de las canciones de distintas fuentes se lo conoce comúnmente como *scraping*. Algunas de estas fuentes ofrecen una

Tabla 3.5: Letra del tango "Volver" de Alfredo Le Pera extraída de Todotango

Yo adivino el parpadeo
de las luces que a lo lejos,
van marcando mi retorno.
Son las mismas que alumbraron,
con sus pálidos reflejos,
hondas horas de dolor.

Y aunque no quise el regreso,
siempre se vuelve al primer amor.
La quieta calle donde el eco dijo:
"Tuya es su vida, tuyo es su querer",
bajo el burlón mirar de las estrellas
que con indiferencia hoy me ven volver.

Volver,
con la frente marchita,
las nieves del tiempo
platearon mi sien.
Sentir, que es un soplo la vida,
que veinte años no es nada,
que febril la mirada
errante en las sombras
te busca y te nombra.

Vivir,
con el alma aferrada
a un dulce recuerdo,
que lloro otra vez.
Tengo miedo del encuentro
con el pasado que vuelve
a enfrentarse con mi vida.
Tengo miedo de las noches
que, pobladas de recuerdos,
encadenen mi soñar.

Pero el viajero que huye,
tarde o temprano detiene su andar.
Y aunque el olvido que todo destruye,
haya matado mi vieja ilusión,
guarda escondida una esperanza humilde,
que es toda la fortuna de mi corazón.

interfaz para poder consultar y obtener las letras, mientras que para el resto directamente se procesa el código HTML para automatizar esta tarea.

En resumen, a continuación se lista la información provista por las distintas fuentes:

- **Todotango:** para esta fuente se tiene una relación directa entre los metadatos y las letras, contando a lo sumo con una única letra para cada canción. Inicialmente este conjunto comprendía un total de 5647 letras distintas
- **Lyricswikia:** se encontraron un total de 577 letras de canciones en esta fuente y cada una de ellas se relaciona con la canción a partir de la cual se hizo la consulta
- **Musixmatch:** solo 62 letras de canciones fueron encontradas, por lo tanto esta fuente se descartó
- **Minilyrics:** usando la interfaz de búsqueda se obtuvieron múltiples resultados para cada canción: un total de 2914 canciones contaron con al menos una letra. Para esta fuente se obtuvo un total de 40927 letras. Un gran porcentaje de estos resultados eran canciones con el mismo nombre, con un nombre muy similar, o también podían ser distintas versiones de otros músicos de la misma canción (los cuales no queríamos incluir). Para esta fuente algunas letras contenían intervalos de tiempo, indicando el momento en que cada línea se presenta en la canción. La información de tiempo se removió usando una expresión regular. Se muestra un ejemplo de esta fuente en la Tabla 3.6.
- **Chartlyrics:** para esta fuente también se utilizó la interfaz de búsqueda, obteniéndose múltiples resultados para cada canción. Un total de 2740 canciones tuvo una o más letras, sumando un total de 3253 letras.
- **Songlyrics:** no se obtuvo ningún resultado con esta fuente
- **Lyricsmania:** no se obtuvo ningún resultado con esta fuente
- **12K:** a diferencia de las anteriores, esta fuente no ofrece ninguna interfaz de búsqueda y además todas las letras son de tango. Por lo tanto, a partir de los títulos de las canciones se pudieron relacionar solamente 2926 canciones. Para relacionar los títulos se utilizó distancia Jaccard (Jaccard, 1901). Al comienzo de las letras de esta fuente había un texto introductorio que fue removido comparando la letra con la correspondiente en Todotango en el caso que era posible. El texto introductorio

para la canción Malevaje se muestra en la Tabla 3.7.

Tabla 3.6: Letra del tango "Malevaje"de Enrique Santos Discépolo. Versión de la fuente Minilyrics.

```
[id: fq_pilcsiqn]
[ar:Carlos Gardel]
[ti:Malevaje]

[00:05.33][00:14.82][00:29.84][00:37.13][00:57.71][01:13.51][01:20.36][01:34.63]
[00:07.94]¡Decí, por Dios, que me has dao,
[00:10.67][01:38.22]que estoy tan cambiao!...
[00:12.96]¡No sé más quién soy!...
[00:15.79]El malevaje extrañao
[00:19.49]me mira sin comprender;
[00:23.30]me ve perdiendo el cartel
[00:25.91]de guapo que ayer
[00:27.55]brillaba en la acción.
[00:30.81]No ven que estoy embretao
[00:33.54]vencido y maniao
[00:35.18]en tu corazón.


[00:44.33][01:50.00]Te vi pasar tangueando, altanera,
[00:48.13][01:53.37]con un compás tan hondo y sensual,
[00:51.61][01:56.86]que no hice más que verte y perder
[00:54.78][02:00.23]la fe, el coraje, el ansia'e guapear...
[00:58.91][02:04.02]No me has dejao ni el pucho en la oreja
[01:02.50][02:07.72]de aquel pasao malevo y feroz.
[01:05.88][02:11.31]Ya no me falta pa completar
[01:09.48][02:14.90]más que ir a misa e hincarme a rezar.
[01:14.16]Ayer, de miedo a matar,
[01:16.88]en vez de pelear,
[01:18.41]me puse a correr...
[01:21.19]Me vi en la sombra o finao,
[01:24.51]pensé en no verte y temblé.
[01:27.88]Si yo {que nunca aflojé-
[01:30.50]de noche angustiao
[01:32.24]me pongo a llorar...
[01:35.28]¡Decí por Dios que me has dao
[01:40.07]¡No sé más quien soy!
```

Tabla 3.7: Sección inicial de la letra del tango "Malevaje"de Enrique Santos Discépolo. Versión de la fuente 12K.

```
MALEVAJE
Letra de Enrique Santos Discépolo
Musica de Juan de Dios Filiberto

Estrenado por Azucena Maizani en la Fiesta del Tango en el Teatro
Astral de Buenos Aires, el 21-9-1928. Grabado por Ignacio Corsini;
Carlos Gardel. Malevaje fue también interpretado por Roberto
Goyeneche en Todo Goyeneche de FM Tango Para Usted, 1992, ECD 50608

Al parecer llegó Enrique Santos Discépolo y se acabaron los malevos,
al menos esos torvos personajes de "la secta del cuchillo y el
coraje".  Se cuenta que Carlos Gardel le dijo a Discépolo- "Tu malevo,
maneado por el amor, es capaz de confesar su angustia ante el miedo de
perder a la mujer que quiere, llorarla si es preciso, y no a ponerle
el cuchillo al cuello para que se doblegue".

No sólo los cambios de los valores en la ciudad, la quiebra del "pacto
solidario" ente los grupos, la desconfianza en los valores religiosos,
sino una renovación de costumbres...  En ese devenir desconocido el
protagonista dice:  "No sé más quien soy".

Fue concebido en el barro de La Boca, adonde iba Discépolo a escuchar
tangos interpretados por el músico Juan de Dios Filiberto y a trabajar
con él.  Es el barrio en el que nació Buenos Aires, de la obra de Juan
de Garay, y donde tres siglos después se instaló abundante, reunida,
la inmigración italiana y española, sumándose a la población negra y
criolla preexistente.  Es famoso este barrio como una de las
indudables cunas del tango.  Algunos próceres literarios sospecharon
del torrente inmigratorio que se extasiaba dionisiaco en el tango y
que jugaba con proyectos de "Repúblicas federativas".  Ricardo Rojas
deseó que se trasladara una estatua de Mazzini "que no podía seguir en
las mismas puertas de Buenos Aires".  Aconsejó:  "Si se hace la
traslación no ha de ser desde luego a La Boca, pues tal cosa
importaría consagrar oficialmente esa población como pedazo da Italia"
(sic).  No se preocuparon, sin embargo, de que ondeara la bandera
británica en las estancias de la Patagonia que comprendían miles de
kilómetros cuadrados.

Casualidad y desarrollo influyeron en la creación de este tema.  No
tiene antecedentes pero sí tuvo extraordinaria consecuencia posterior.
Carlos Gardel entendió e!  tono del mensaje; ahí está en su voz para
desmentir el machismo absoluto adjudicado al tango canción.

¡Decí, por Dios, que me has dao, [.....]
```

3.3. Selección de un candidato por fuente

Como ya fue mencionado, para las fuentes Minilyrics y Chartlyrics se contaba con múltiples letras candidatas para cada canción. Por lo tanto, a continuación se seleccionó un único candidato por canción para cada fuente.

Para el caso de Chartlyrics, el nombre del artista en cada letra candidata se comparó con el artista correspondiente en los datos estructurados extraídos de Todotango (ya sea el letrista, el compositor o algún artista que grabó la canción).

Para el caso de Minilyrics, se seleccionó la letra que tuviera una mayor similitud con la correspondiente en Todotango usando indice de Jaccard. En el caso que no hubiera una letra en Todotango para la canción o si la similitud era demasiado lejana, se comparaban los nombres de los artistas de la misma forma que para Chartlyrics.

Luego de realizar este proceso, el número de canciones con letras para estas fuentes se redujo a 1005 para Minilyrics y 60 para Chartlyrics.

3.4. Corpus de letras final

A continuación se seleccionó una única versión de letra para cada canción a partir de las distintas fuentes. La forma en que se definió la prioridad entre las fuentes fue analizando la similitud y el largo de las letras. Se consideró el largo de las letras ya que se deseaba obtener la fuente que contara con las versiones más completas, de forma que estuvieran presente las repeticiones de las secciones en el caso que corresponda. Para calcular la similitud entre las letras, se utilizó nuevamente Jaccard y luego se calculó el promedio para cada fuente.

Es importante notar que cuando se utiliza Jaccard, se aplica a nivel de palabras y no de caracteres. También se aplicó un proceso de estandarización antes de calcular Jaccard, reemplazando mayúsculas por minúsculas, removiendo tildes de las palabras y también removiendo los siguientes caracteres antes de comparar las letras:

```
! ¡ ? ¿ . , * ’ " ‘ ( ) : ;  \
```

Tabla 3.8: Distancia Jaccard promedio entre fuentes.

	Minilyrics	**Todotango**	**12k**	**Lyricswikia**	**Chartlyrics**
Minilyrics	-	0.859525	0.732773	0.871103	0.364506
Todotango	-	-	0.775914	0.990372	0.236051
12k	-	-	-	0.829158	0.249605
Lyricswikia	-	-	-	-	0.631715

Tabla 3.9: Largo en cantidad de palabras de las letras para cada fuente

	Min	**Max**	**Average**
Minilyrics	8	396	180
Todotango	29	452	161
12k	12	2040	178
Lyricswikia	1	402	166
Chartlyrics	1	462	258

En la Tabla 3.8, se muestra la distancia Jaccard media entre las fuentes. Los valores se encuentran entre cero y uno; valores más cercanos a uno indican que las letras son similares a partir de las palabras que contienen. En la Tabla 3.9, se muestra el largo de las letras para cada fuente medido en cantidad de palabras.

A partir de las tablas 3.8 y 3.9, se puede ver claramente algunos problemas que presentan las distintas fuentes. El número máximo de palabras para la fuente 12K es el más grande por una amplia diferencia en comparación con el resto de las fuentes, lo cual indica que incluso luego de haber removido la parte inicial de algunas letras persisten algunos casos en los cuales no había otra letra para comparar. Esto se tuvo en cuenta para definir el orden de prioridad entre las fuentes.

También es claro que las letras de Todotango son más cortas en promedio que el resto de las fuentes, lo cual confirma la teoría planteada anteriormente. Para el caso de Minilyrics se tienen letras más largas que Todotango y la distancia Jaccard es bastante alta entre ellas, lo cual se puede entender como que Minilyrics presenta letras más completas que Todotango.

Notar también que Minilyrics tiene la menor diferencia entre mínimo y máximo en el largo de letras, lo cual contribuye a pensar que presenta una buena calidad de letras.

En contrario a lo que sucede para Minilyrics, es posible ver que la distancia Jaccard es extremadamente grande entre Todotango y Lyricswikia pero que el

largo es muy similar, lo cual puede significar que contienen la misma información. Una posible explicación es que el contenido de una fuente fue generado a partir de la otra.

A partir del número de canciones con letras, es claro ver que las fuentes más importantes son 12K, Todotango y Minilyrics. Combinando estas tres fuentes podemos obtener un total de 6300 canciones.

A partir del análisis realizado en esta sección, se pudo ver que aquella que presenta las letras más completas es Minilyrics y debe ser la primera opción a utilizar al momento de seleccionar la versión para cada canción. Luego, no hay una opción que sea mejor que Todotango, por lo tanto, esta se utilizó como la siguiente opción y finalmente 12K siendo la tercera alternativa.

3.5. Información de año de las canciones

A los efectos de la clasificación, es importante tener información del año de las canciones. Del conjunto de metadatos extraídos inicialmente de Todotango, solo 2678 canciones contaban con año del total de 9660 canciones. Por lo tanto, se debió completar esta información para las canciones restantes utilizando otra fuente de información.

Para completar el año de las canciones, se recolectó información extra de la web tango.info[1]. A partir de la web tango.info, para cada canción se obtuvo una referencia a su página en Todotango y además un identificador de SADAIC (Sociedad Argentina de Autores y Compositores de Música)[2]. Luego, a partir de la página *web* de SADAIC se obtuvo información adicional sobre la fecha en que fue registrada.

Para cada canción que contaba con letra, se completó el año utilizando la información en el siguiente orden de prioridad:

1. Todotango: En caso de contar con la fecha para la canción en los metadatos extraídos de Todotango, se utilizó la información.
2. Sadaic: Para cada canción sin año se utilizó la información extraída de SADAIC en caso de estar disponible.
3. Grabación: Para las canciones que aún restaban sin año se seleccionaron las grabaciones relacionadas y se tomó el año de la primera grabación.

[1] https://tango.info/work/
[2] https://www.sadaic.org.ar/

4. Fecha de nacimiento de autores: Para aquellas canciones que aún no tenían año se calculó el intervalo en el que coincidían todos los artistas, ya sean compositores o letristas y se hizo un promedio.

Luego de realizar este procedimiento para las 6300 canciones con letra, un total de 5965 canciones se completó el valor del año correspondiente. En la Figura 3.2 se puede ver por cada año el número de canciones con letra presentes; aquí se puede ver que el año 1940 es el que tiene más canciones con un total de 185.

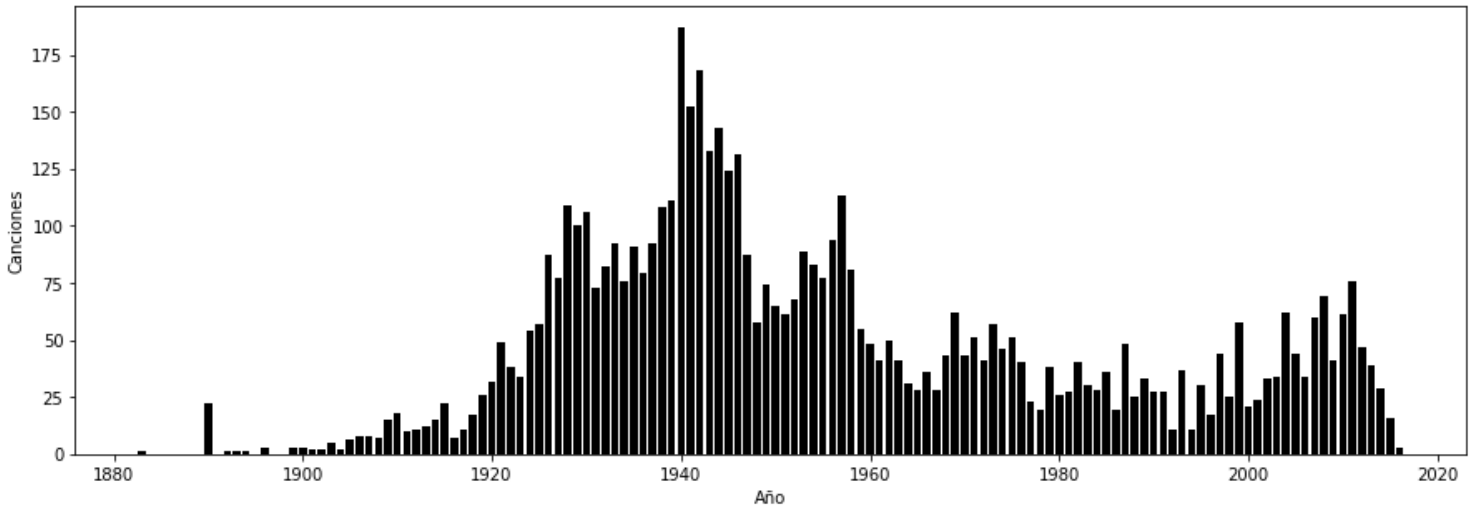

Figura 3.2: Cantidad de canciones por año.

3.6. Análisis de repetición en letras

En la Sección 3.4 se menciona que es importante contar con un corpus con las letras completas de las canciones para el trabajo que se desea realizar. Por lo tanto, las letras deben incluir las secciones repetidas de cada canción.

El primer análisis realizado sobre el corpus consistió en calcular una matriz de similitud entre cada línea de una letra, utilizando la distancia de Levenshtein (Levenshtein, 1966) entre las líneas. A partir de la matriz de similitud es posible identificar las secciones repetidas de una canción. A continuación, se muestran dos canciones del corpus que presentan secciones repetidas.

En la Figura 3.3 se muestra la matriz de similitud para la canción "Volver" de Alfredo Le Pera. Se puede ver una diagonal en color negro entre las líneas 39 y 43 que indican la repetición de una sección.

En la Figura 3.4 se muestra la matriz de similitud para la canción "Malevaje" de Enrique Santos Discépolo. En este caso se puede ver una diagonal mucho más marcada que en el caso anterior, entre las líneas 34 y 42.

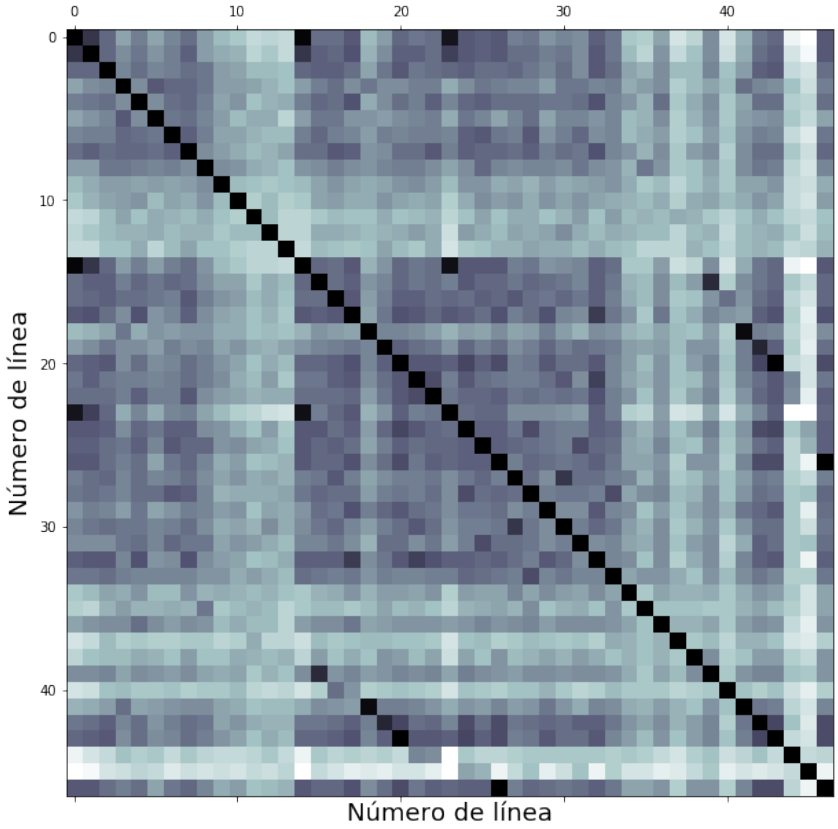

Figura 3.3: Matriz de similitud de la letra "Volver" de Alfredo Le Pera.

Mediante la matriz de similitud de cada letra, se podrían identificar automáticamente las secciones y repeticiones en cada canción. Luego, esta información podría ser utilizada para categorizar las canciones o también para calcular estadísticas dentro de cada género.

3.7. Agrupamiento y visualización de las letras

El segundo análisis sobre el corpus consistió en el agrupamiento automático y la visualización del corpus a partir de la reducción de la dimensión de las letras. Para esto, primero fue necesario obtener una representación de las letras sobre la cual aplicar los algoritmos. En esta sección se describe la representación

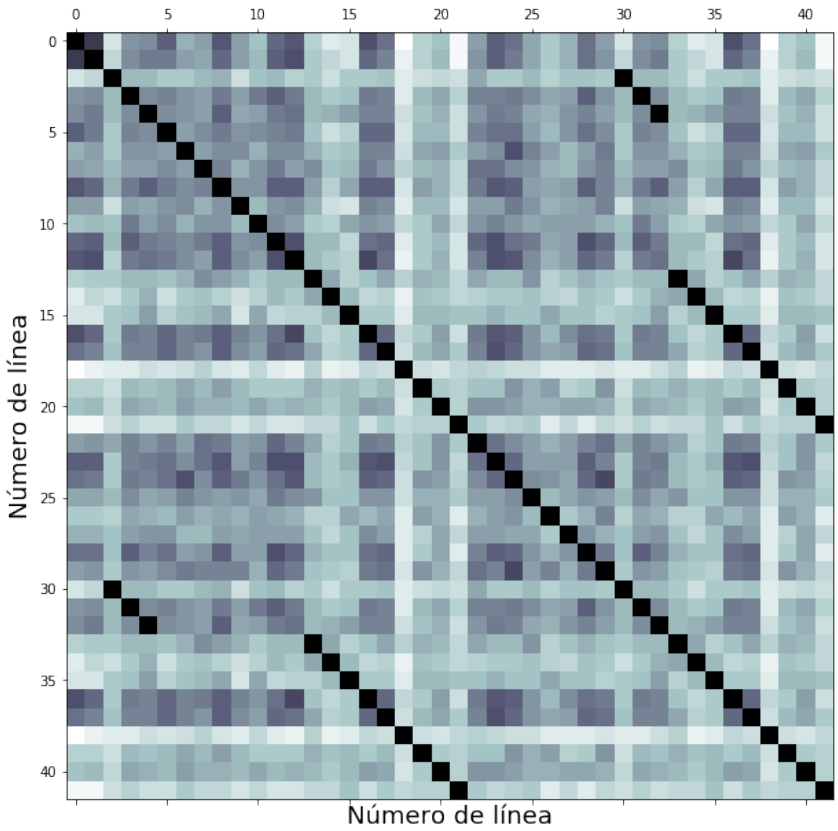

Figura 3.4: Matriz de similitud de la letra "Malevaje" de Enrique Santos Discépolo.

utilizada, también se detalla cómo se realizó el agrupamiento automático y la reducción de la dimensión para la visualización del corpus.

3.7.1. Primera representación de letras

Para los análisis siguientes se utilizó una representación simple, común en varias publicaciones mencionadas en el Capítulo 2. Para cada letra del corpus se obtuvo una representación vectorial utilizando el método de bolsa de palabras con la medida tf-idf. A cada palabra, se le agregó la etiqueta gramatical correspondiente y, en caso de ser un verbo, se utilizó su raíz.

3.7.2. Agrupamiento automático de las letras

Utilizando las representaciones vectoriales calculadas en la sección anterior, se entrenaron modelos de agrupamiento no supervisado utilizando los algoritmos Spectral Clustering (Von Luxburg, 2007) y K-means (Lloyd, 1982).

Se buscó el valor óptimo de *clusters* entre 2 y 16, utilizando dos medidas

para evaluar la calidad de los modelos a partir de qué tan cercanos son los elementos dentro de cada *cluster* y qué tan diversos son los *clusters*. Las medidas utilizadas fueron Calinski-Harabasz (CH) (Caliński and Harabasz, 1974) y silhouette (Rousseeuw, 1987).

Cuanto más altos son los valores de CH y silhouette, significa que la forma en que fueron agrupados es mejor. Esto significa que los *clusters* presentan una mayor distancia entre ellos y las canciones una menor distancia dentro de cada *clusters*. La Figura 3.5 muestra los valores correspondientes a silhouette para cada número de *clusters*. Por otro lado, la Figura 3.6 muestra el valor correspondiente a CH.

En ambas figuras se muestran los valores para Spectral Clustering y para K-means utilizando los vectores obtenidos en la parte anterior. Podemos ver que en ambos casos lo mejores valores se encuentran entre dos y cuatro *clusters*, pero debemos tener en cuenta que utilizar un número muy reducido de *clusters* no aporta mucho. Por lo tanto, entendemos que un buen balance puede ser utilizando cuatro *clusters*.

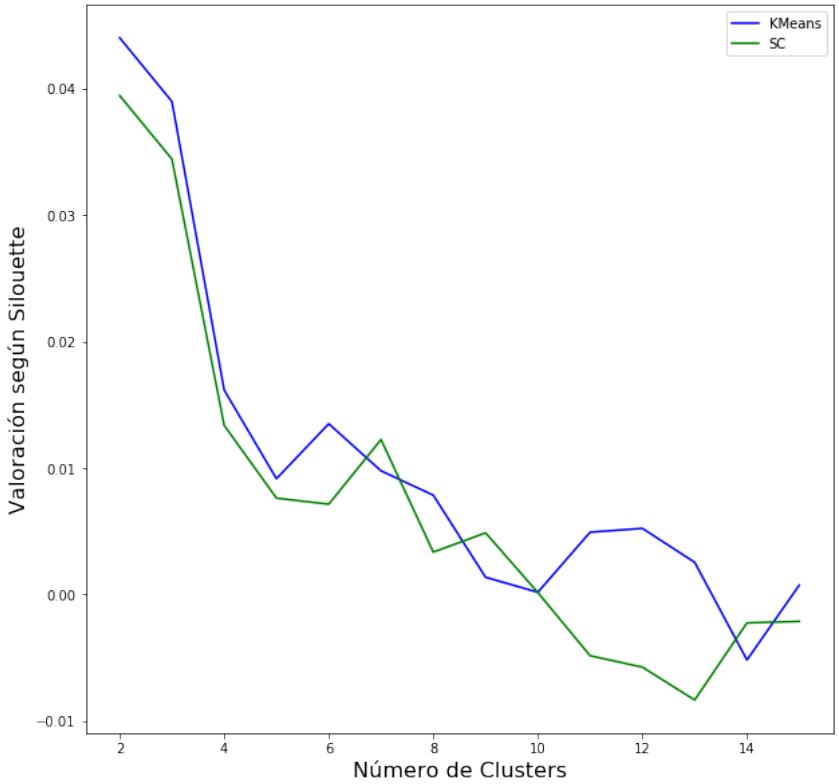

Figura 3.5: Valoración según silhouette para distintos números de clusters.

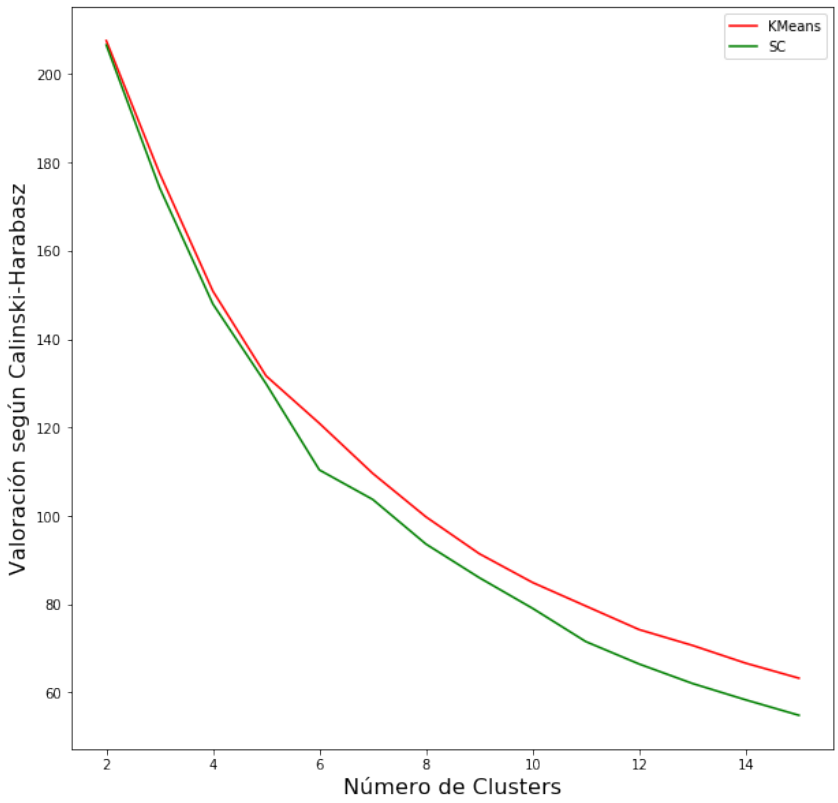

Figura 3.6: Valoración según Calinski-Harabasz para distintos números de clusters.

3.7.3. Visualización de las letras

A partir de las representaciones vectoriales obtenidas, utilizamos el algoritmo T-SNE para reducir la dimensión de los vectores a tres y así poder visualizarlos (Maaten and Hinton, 2008).

En la Figura 3.7, se muestra el resultado de aplicar la reducción de dimensión y además se muestran los *clusters* que fueron reconocidos en la sección anterior mediante Spectral Clustering. En la Figura 3.8, se aplica el mismo procedimiento pero para el algoritmo K-means.

Si bien a la hora de entrenar el modelo mediante T-SNE no fue utilizada la información de los *clusters* obtenidos mediante Spectral Clustering ni K-Means, podemos ver que en ambos casos los elementos que se encuentran en el mismo *cluster* también se encuentran más próximos en la visualización. De todas formas, esto no nos permite concluir demasiado sobre la calidad de la representación de las letras utilizada.

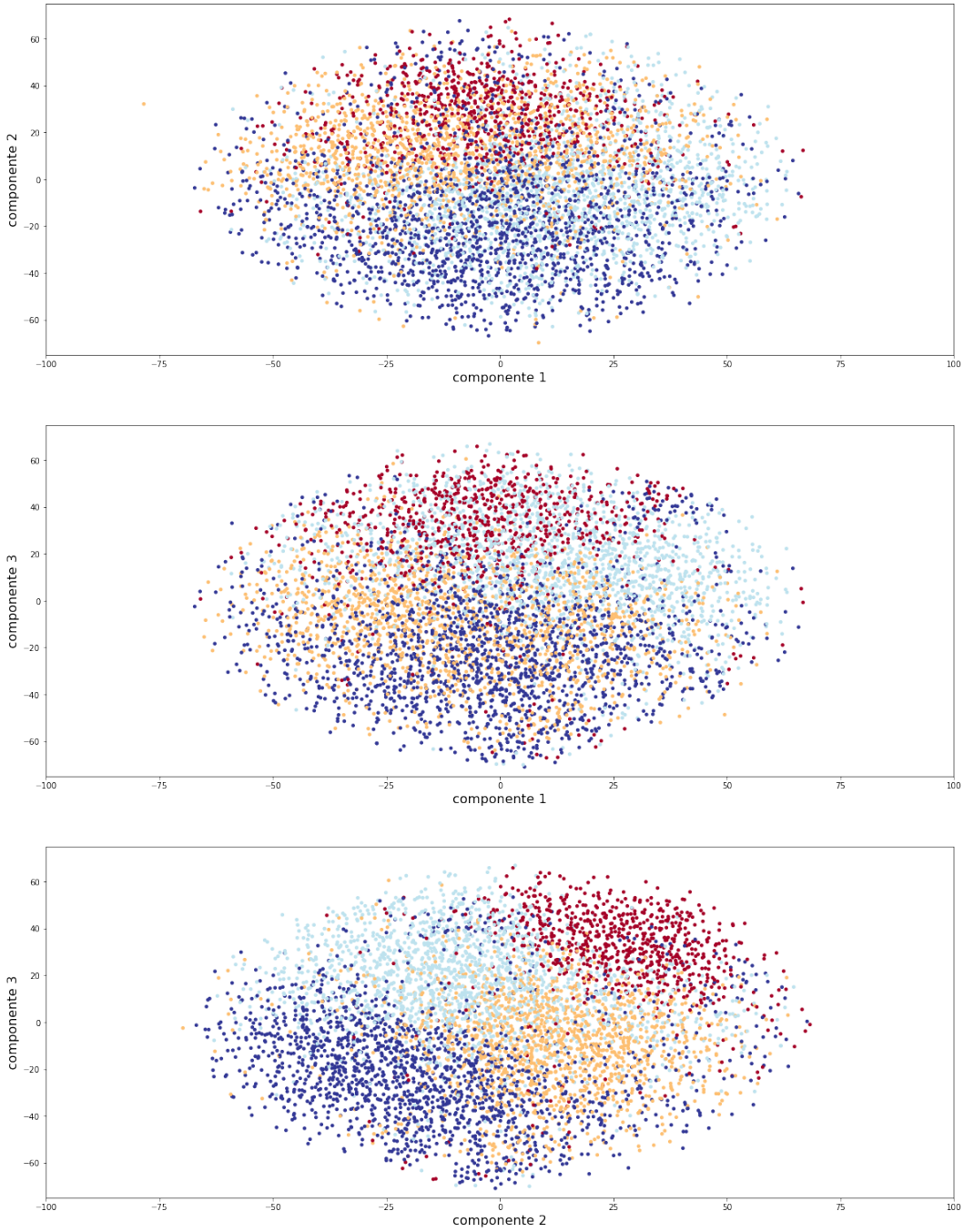

Figura 3.7: Visualización de las letras en 3 dimensiones a partir de T-SNE. Los colores corresponden a los clusters reconocidos por Spectral Clustering .

3.8. Conclusión

En este capítulo se describió el corpus de letras construido que cuenta con 6300 letras. Además de las letras, el corpus cuenta con información de año

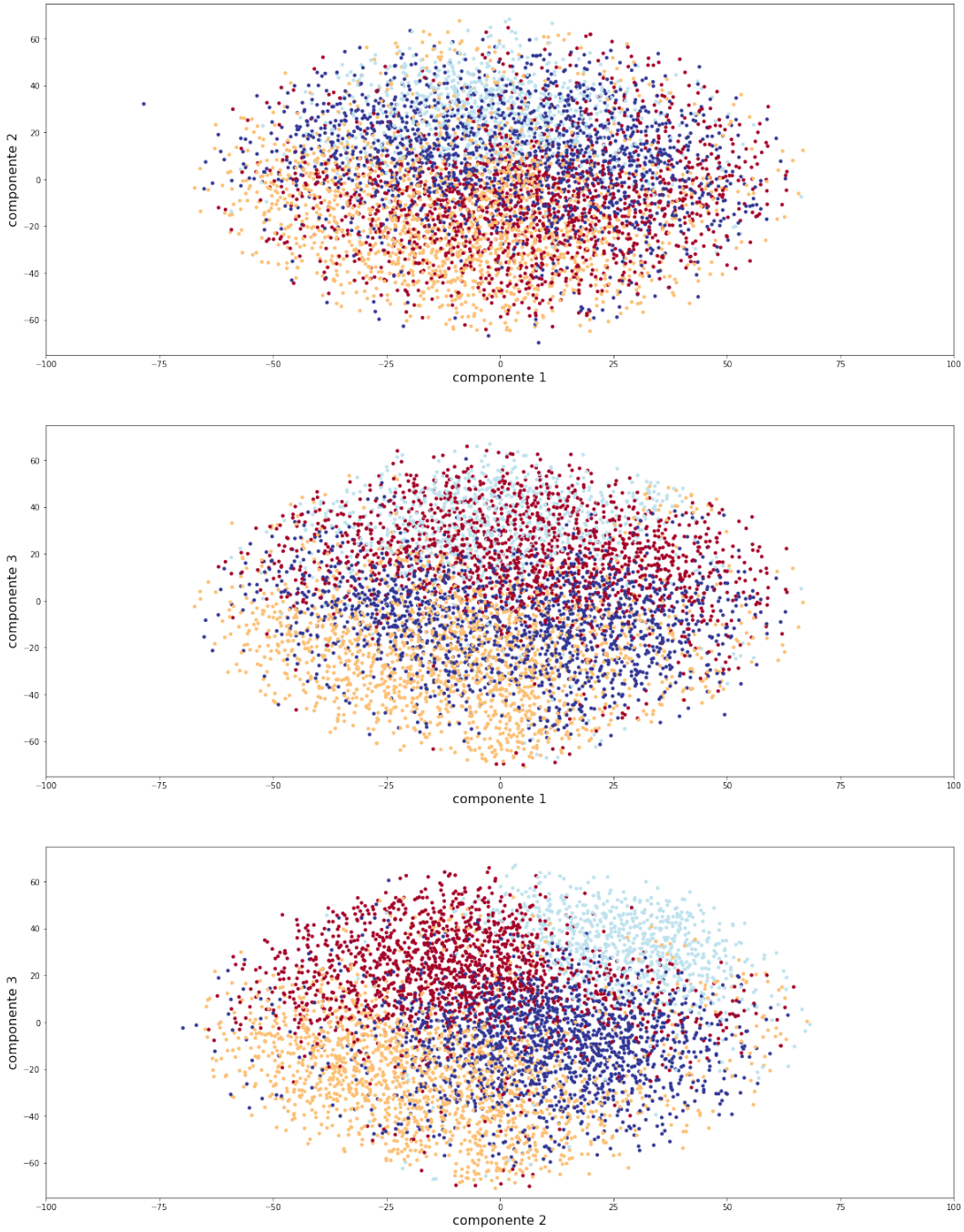

Figura 3.8: Visualización de las letras en 3 dimensiones a partir de T-SNE. Los colores corresponden a los clusters reconocidos por K-Means Clustering.

y género. Un total de 5965 canciones tienen asociado un valor de año y la totalidad de las canciones cuenta con información de género. Las canciones también cuentan con información de letrista que es utilizada para crear grupos de artistas relacionados de forma de clasificar las canciones. En el Capítulo 5

se utiliza este corpus para evaluar las distintas representaciones en múltiples tareas de clasificación.

Este conjunto de datos ofrece más información utilizada en el Capítulo 6 para desarrollar una aplicación web que le permite al usuario navegar por el conjunto de datos.

El corpus es explorado inicialmente en este capítulo. En primer lugar, se muestra cómo se puede analizar la estructura de una letra a partir de la similitud de las líneas. Mediante este análisis se muestran algunos casos en que las letras contaban con secciones repetidas, lo que indica una buena calidad de las letras. A partir del cálculo de la matriz de similitud, se podrían identificar automáticamente las secciones y repeticiones en cada letra. Luego, se podrían obtener estadísticas de cada género sobre cuáles estructuras son más características.

Para el segundo análisis, primero se definió una representación a utilizar a partir de las letras de las canciones en el corpus. Luego, se aplicaron técnicas de agrupamiento automático para identificar grupos de canciones a partir de su letra. En este análisis se utilizaron dos medidas para definir el número adecuado de grupos a partir de la calidad que ofrecen. Estos grupos de canciones identificados podrían ser utilizados para generar recomendaciones para usuarios.

También se mostró una forma de visualizar un conjunto de letras a partir de una representación, utilizando técnicas de reducción de la dimensión. Esta visualización podría ser utilizada para una aplicación para navegar por un grupo de letras.

El objetivo principal de esta tesis es analizar algunos métodos para representar las letras que mantengan la similitud de las canciones. Si bien los resultados obtenidos para la visualización y el agrupamiento de las letras están fuertemente relacionados con la representación utilizada, no nos permiten evaluarla directamente. Por lo tanto, en los próximos capítulos se plantean otras representaciones y se evalúan mediante distintas tareas de clasificación.

Capítulo 4

Modelo generativo basado en redes recurrentes

A partir de la revisión realizada en el Capítulo 2, se definió que para representar las letras se exploraría una técnica reciente presentada por Radford et al. (2017) basada en redes neuronales profundas. En este capítulo, se describe la solución utilizada, basada en un modelo generativo para modelar el lenguaje de las letras de las canciones.

El modelo generativo toma como entrada una secuencia de caracteres y produce como salida el próximo carácter. Por lo tanto, debe mantener internamente la información más importante de la entrada para generar el próximo carácter. Esta información interna puede ser utilizada como representación del texto de entrada.

Para aplicar el método de Radford et al. (2017) es necesario disponer de un mayor número de letras al recolectado en el Capítulo 3, ya que estos modelos requieren mucha cantidad de datos para ser entrenados. Por lo tanto, en este capítulo también se describe cómo se recolectaron alrededor de 800,000 canciones en español que son utilizadas para el entrenamiento de la red. El entrenamiento del modelo no requiere ninguna información complementaria a las letras; por lo tanto, no fue necesario recolectar ningún metadato de las canciones.

Este capítulo está estructurado de la siguiente manera: en la primera sección se describe el funcionamiento de las redes recurrentes y las redes LSTM; también se describe cómo es posible obtener una representación a partir de la red. Luego se describe cómo se amplió el corpus para entrenar la red. En la

Sección 3 se detallan las arquitecturas de modelos que se utilizaron y se muestran algunos resultados preliminares. Por último, en la Sección 4 se mencionan las conclusiones.

4.1. Redes neuronales recurrentes

En los últimos años vimos una creciente adopción de diferentes técnicas de inteligencia artificial que permiten la generación de contenidos tales como imágenes (Google AI, 2015), películas (Ars Technica, 2016) o música (Mann, 2017). En parte esto se debe a la capacidad de las Redes Neuronales Recurrentes (RNN) de modelar secuencias de datos, la facilidad de acceder a grandes cantidades de información y al poder de cómputo que ofrecen las tarjetas gráficas.

En las RNN cada estado depende únicamente de la entrada y el estado de la red en el paso anterior. En la Figura 4.1 se muestra un ejemplo de una RNN con una sola capa oculta. Este tipo de red toma como entrada una secuencia de datos y obtiene como salida otra secuencia de datos (Karpathy, 2015).

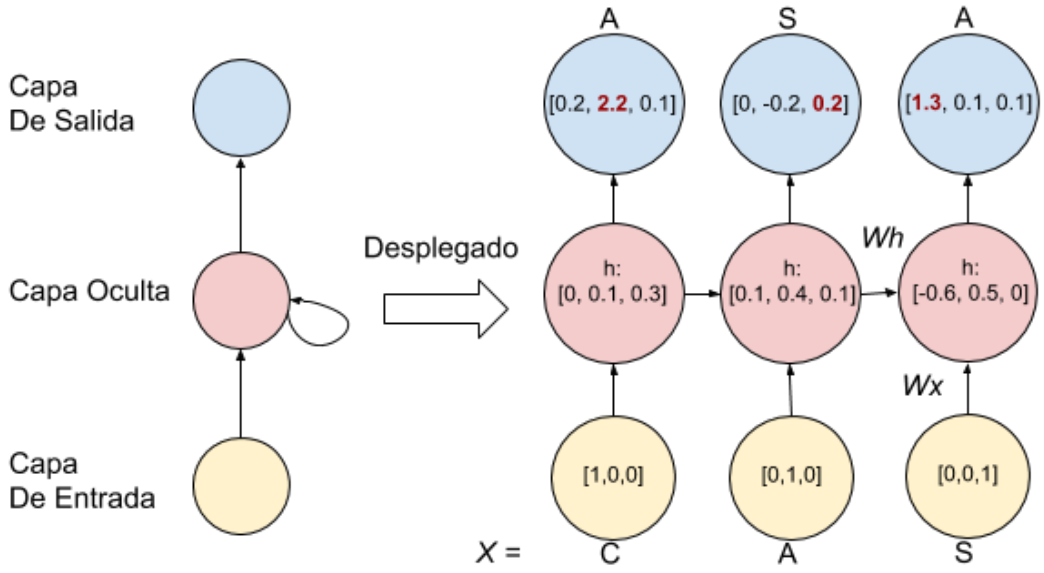

Figura 4.1: Ejemplo de RNN de tipo muchos-a-muchos con una entrada de 3 dimensiones y una capa oculta de 3 celdas. Se combina cada entrada con el estado anterior para producir una salida. La capa de salida produce una probabilidad para cada carácter estimado por la red.

Las RNN son un tipo de red en donde se procesa la entrada de forma

secuencial y se mantiene un estado interno con información relativa a los datos vistos hasta el momento. Como muestra la Figura 4.1, en las redes neuronales recurrentes se mantiene la salida anterior que es utilizada en conjunto con la próxima entrada para calcular la próxima salida.

En la RNN de la Figura 4.1 la entrada se representa mediante *one-hot-encoding* y la capa oculta está compuesta por 3 celdas. Una vez que se combina la entrada con el estado se produce una salida que corresponde a la probabilidad de cada carácter. Durante el entrenamiento se comparan las probabilidades con el valor esperado y se ajustan los pesos mediante el algoritmo de *backpropagation*

Por lo tanto, dada una entrada X, definida como una secuencia de elementos:

$$X = [x_1, x_2, ..x_t..x_N] \tag{4.1}$$

El estado interno de la red en el paso t, se calcula mediante h_t:

$$h_t = f(W_x x_t + W_h h_{t-1}) \tag{4.2}$$

Siendo W_x la matriz de pesos para cada elemento de la entrada y W_h la matriz de pesos para cada estado anterior. La función f se denomina función de activación. Los valores de las matrices se ajustan utilizando el algoritmo de *backpropagation through time* (BPTT) (Werbos, 1990).

4.1.1. Redes LSTM

Las RNN presentan el problema llamado *vanishing gradient* (Lipton et al., 2015), en donde la red no logra aprender debido a que las variaciones en los primeros elementos de la secuencia son muy pequeñas y se pierden en los siguientes pasos (cuando finalmente la salida es comparada con la esperada). Para resolver este problema surgen las redes *Long short-term memory* (LSTM) (Hochreiter and Schmidhuber, 1997).

La diferencia que se introduce con las redes LSTM es la sustitución de los nodos tradicionales por las llamadas celdas de memoria, en donde se agrega la posibilidad de hacer modificaciones (agregar o quitar información) a la memoria o pasarla a la siguiente celda como fue recibida (Olah, 2015).

La aplicación de redes LSTM sobre texto puede ser a nivel de caracteres o de

palabras. Para el caso de caracteres, el entrenamiento consiste en tomar como entrada una secuencia de caracteres e intentar predecir cuál será el próximo. La red a nivel de caracteres tiene como ventaja que no se debe definir un vocabulario fijo, ya que se permite que la red modele el lenguaje a partir de la combinación de los caracteres.

4.1.2. Función de pérdida

Para evaluar el desempeño de los modelos durante el entrenamiento es usual utilizar el valor de la función de pérdida. Esta función indica cuánto error se comete al predecir el próximo carácter. En general, se utiliza un conjunto de validación para evaluar el modelo durante el entrenamiento.

La función de pérdida está basada en el promedio de la entropía cruzada de cada cadena de caracteres. En ocasiones a esta función de pérdida se le refiere como bits por carácter o BPC por las siglas en inglés.

Dada una cadena de caracteres de largo T (tamaño de entrada de la red), siendo P_t la probabilidad correspondiente al carácter correcto en dicha posición (x_t) y $\widehat{P}_t$ la probabilidad estimada por la red para el carácter x_t, la función de pérdida se calcula como:

$$bpc(string) = \frac{1}{T}\sum_{t=1}^{T} H(P_t, \widehat{P}_t) = -\frac{1}{T}\sum_{t=1}^{T} log_2 \widehat{P}_t(x_t) \tag{4.3}$$

4.1.3. Representación vectorial a partir de redes LSTM

Como describe Radford et al. (2017) es posible utilizar la red entrenada para obtener una representación del texto. Esto es debido a que, dada una entrada, se actualiza el estado interno de la red de forma que mantenga la información relevante para generar el próximo carácter. Por lo tanto, se puede utilizar el estado como representación del texto de entrada. Nótese que la dimensión de las representaciones será igual a la cantidad de celdas en la última capa de la red.

Es necesario dividir el texto que se desea representar en secuencias de caracteres del mismo largo que la entrada de la red. En nuestro caso, se dividen las canciones en secuencias de largo fijo y se obtiene más de una representación vectorial para cada canción. Por lo tanto, estos vectores se deben combinar para producir una única representación de la canción.

Por ejemplo, si x, y y z corresponden a los vectores de una canción:

$$x = (x_1, x_2, ..., x_n) \tag{4.4}$$

$$y = (y_1, y_2, ..., y_n) \tag{4.5}$$

$$z = (z_1, z_2, ..., z_n) \tag{4.6}$$

La representación final de la canción (r) está dada de la siguiente manera:

$$r = (f(x_1, y_1, z_1), f(x_2, y_2, z_2), ..., f(x_n, y_n, z_n)) \tag{4.7}$$

Algunas posibles funciones (f) utilizadas para combinar los vectores de una canción son la desviación estándar, la suma y el promedio.

4.1.4. Generación de nuevas letras

Si bien esta tesis no tiene objetivo generar nuevas letras, el modelo una vez entrenado permite hacerlo. Una vez que la red está entrenada, la forma en que se utiliza para generar nuevas secuencias se denomina método de muestreo, el cual consiste en tomar la salida de la red y utilizarla a continuación como entrada y repetir este proceso. Por lo tanto, para generar nuevas letras inicialmente se toma un texto 'semilla' y se lo utiliza como entrada a la red, luego se le agrega a la secuencia el carácter retornado por la red y se repite este proceso las veces que sea necesario.

Es importante la forma en que se selecciona el siguiente carácter a partir de la distribución de probabilidad resultante de la red, ya que se debe elegir cuál es el carácter que será agregado a la secuencia con cierta aleatoriedad. Es decir, no necesariamente el carácter más probable retornado por la red será el utilizado, ya que de tal forma continuamente se entraría en una repetición de secuencias.

4.2. Conjunto de datos

El corpus utilizado por Radford et al. (2017) cuenta con un total de 82 millones de comentarios de productos. Esto sugiere que para entrenar una red LSTM no es suficiente con las 6300 letras de canciones que componen el corpus.

Dado que el objetivo final de nuestro trabajo es obtener representaciones

para las letras del Río de la Plata, era necesario recolectar una mayor cantidad de datos para entrenar la red que fueran lo más parecido posible a las letras del Río de la Plata. Por lo tanto, se decidió utilizar letras en castellano de cualquier género musical, ya que sería posible recolectar una gran cantidad de Internet y al mismo tiempo se trata de información similar a la del problema planteado.

Para el entrenamiento de la red únicamente se utilizan los caracteres de las letras, dado que es no supervisado. Por lo tanto, para el entrenamiento no es necesario contar con metadatos de las canciones, lo cual facilita la recolección del conjunto de letras de Internet.

Se utilizó la web musica.com[1] para extraer más letras de canciones ya que presentaba una gran cantidad de letras en castellano y además ofrecía una manera muy simple de obtener automáticamente todas las letras. Por lo tanto, se descargaron todas las letras disponibles descartando aquellas que fueran de otro idioma, utilizando la biblioteca `langdetect`[2].

Como resultado se obtuvieron un total de 821.637 letras en castellano, las cuales suman un total de 1.093.611.031 caracteres y corresponden a 208.759.352 palabras (el promedio de caracteres por palabra es 5.23).

Teniendo en cuenta los caracteres más frecuentes, se tomaron 103 para ser utilizados a la hora de entrenar la red, en la Tabla 4.1 se indican los 103 caracteres utilizados.

Para entrenar la red se utiliza un 90 % de las letras obtenidas de musica.com, seleccionadas de forma aleatoria. A este conjunto de letras lo llamaremos Conjunto de Entrenamiento. El 10 % restante se utiliza para medir el desempeño durante el entrenamiento, a este conjunto lo llamaremos Conjunto de Validación.

Tabla 4.1: Lista de los 103 caracteres utilizados en la red

a	b	c	d	e	f	g	h	i	j	k	l	m	n	ñ	o	p	q	r	s
t	u	v	w	x	y	z	A	B	C	D	E	F	G	H	I	J	K	L	M
N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	[	]	á	é	í	ó
ú	Á	É	Í	Ó	Ú	1	2	3	4	5	6	7	8	9	0	(	)	{	}
<	>		!	¡	?	¿	,	.	‘	“	*	+	-	/	_	:	;	ü	Ü
/n	/r	/t																	

[1] https://www.musica.com
[2] https://pypi.python.org/pypi/langdetect

4.3. Arquitecturas evaluadas

Trabajos previos demostraron la utilidad de redes LSTM para modelar el lenguaje, permitiendo generar texto y también obtener representaciones distribuidas de frases. En esta sección se describen las diferentes aproximaciones que se evaluaron para representar las letras musicales siguiendo un enfoque similar.

Se decidió utilizar redes LSTM ya que permiten modelar datos secuenciales; en este caso se utilizaron redes a nivel de caracteres debido a que no se debe definir un vocabulario fijo. Por lo tanto, se entrenaron redes LSTM sobre el Conjunto de Entrenamiento definido en la sección anterior, con el objetivo de predecir el siguiente carácter y se utilizó la función de pérdida (*bpc*) para evaluar el desempeño durante el entrenamiento.

Una vez terminado el entrenamiento, la red se utiliza para obtener las representaciones vectoriales de las letras de canciones (como se menciona en la sección 5.1.3), utilizando la desviación estándar para combinar los vectores de una canción.

Se evalúa el desempeño de los modelos utilizando las representaciones obtenidas para las canciones en dos tareas de clasificación binaria. La primera consiste en predecir si el año es menor o mayor a 1950. La segunda tarea consiste en predecir si la canción se encuentra dentro del conjunto de letras del Río de la Plata o no. Para medir el desempeño en las tareas de clasificación se usa la medida de precisión, calculada de la siguiente manera:

$$precisión = \frac{\text{cantidad de canciones correctamente clasificadas}}{\text{cantidad de canciones a clasificar}} \tag{4.8}$$

Partiendo de la solución propuesta por Chollet (2017), inicialmente se evaluaron algunas arquitecturas de redes LSTM detalladas en la Tabla 4.2. En todos los casos se refiere a redes de tres capas con un tamaño de lotes de 32, representando los caracteres mediante *one-hot-encoding*. Además, la última capa corresponde a una capa densa de dimensión 103, ya que es utilizada para predecir el próximo carácter de la secuencia.

En el diagrama de la Figura 4.2 se muestra cómo las redes LSTM de la Tabla 4.2 toman como entrada una secuencia de caracteres. En cada paso de la secuencia, las celdas toman su entrada y su estado anterior para dar una

Tabla 4.2: Algunos modelos utilizados para evaluar el desempeño de la red.

	Largo secuencia de entrada	**Caracteres de solapamiento entre secuencias**	**Tasa de Dropout**	**Número de neuronas en cada capa**
Modelo A	64	4	0.2	256
Modelo B	40	2	0.2	256
Modelo C	64	2	0.2	128
Modelo D	64	4	0.5	256

salida. Una vez repetido el proceso para todos los elementos de entrada, la red retorna una distribución de probabilidad correspondiente al próximo carácter.

Para el entrenamiento se utiliza la función de pérdida antes mencionada con Adam (Kingma and Ba, 2014) como algoritmo de optimización.

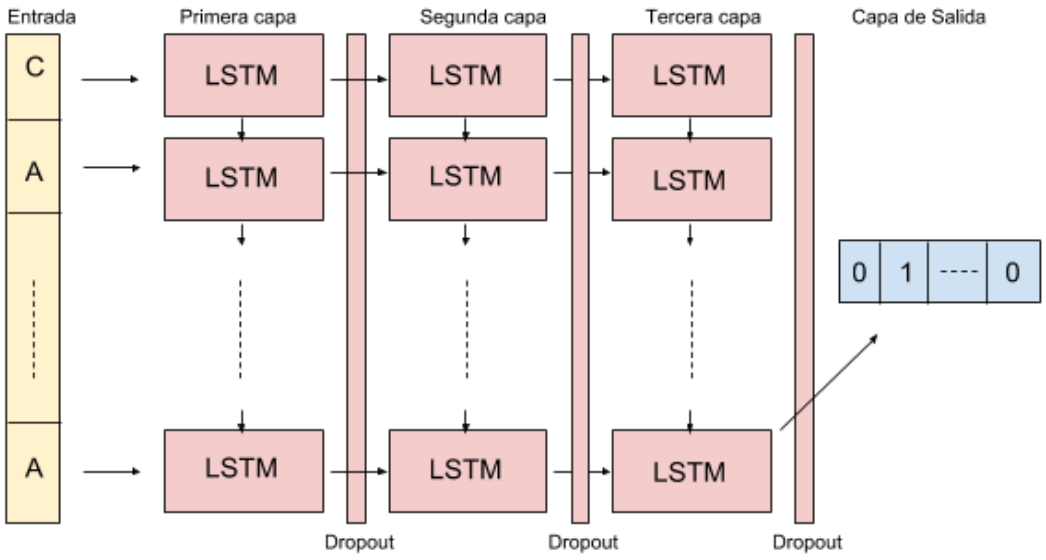

Figura 4.2: Diagrama de red LSTM con 3 capas ocultas. La red presenta una capa de *Dropout* entre las capas ocultas.

En la Tabla 4.2 se indican los cuatro modelos que fueron considerados inicialmente. Antes de entrenar los modelos sobre el Conjunto de Entrenamiento se decidió realizar una prueba solamente con 300 letras. Por lo tanto, se entrenaron los modelos con 270 letras y se validó con las 30 restantes. A partir de esta prueba preliminar, se entendió que para el Modelo A la función de pérdida medida en cada época del entrenamiento presentaba resultados más estables. Por lo tanto, se utilizó esta arquitectura para entrenar con la totalidad del Conjunto de Entrenamiento (de ahora en adelante será llamado Modelo Multicapa).

Para el entrenamiento utilizando la totalidad del Conjunto de Entrenamiento se utilizaron 100 etapas de 10000 lotes cada una. El tiempo aproximado para entrenar el modelo con todos los datos es de una semana, utilizando una GPU modelo TITAN X con 12 GB de memoria.

Para poder comparar resultados también se entrenó, utilizando todo el Conjunto de Entrenamiento, un modelo con una sola capa y 512 celdas LSTM. En la Tabla 4.3 se muestran los parámetros y características de los dos modelos utilizando todo el Conjunto de Entrenamiento. Además, en la Tabla 4.3 se indica el valor de la función de pérdida obtenido sobre el Conjunto de Validación. Aquí se puede ver que el Modelo Multicapa ofrece un menor valor de pérdida.

Tabla 4.3: Algunos modelos utilizados para evaluar el desempeño de la red.

	Capas	**Unidades**	**Parámetros**	**Caracteres**	**Tamaño lotes / épocas / iteraciones**	**Pérdida (BPC)**
Modelo Multicapa	3	256	1.445.735	64 / 4	32 / 100 / 10000 (2x cada letra)	1.39
Modelo capa-simple	1	512	1.172.295	64 / 4	4 / 20 / 20000	1.57

4.3.1. Evaluación de los primeros modelos LSTM

Como ya fue mencionado, el corpus recolectado en el Capítulo 3 fue utilizado para evaluar los modelos. A continuación, se utilizaron los modelos entrenados de la Tabla 4.3 para obtener las representaciones del corpus de letras del Río de la Plata utilizando el método ya mencionado. Luego, se aplicaron dos tareas de clasificación sobre las representaciones para comparar los modelos.

Nótese que cada modelo produce una representación de las letras con diferente dimensión, debido a que la última capa tiene distinta cantidad de celdas.

Para la primera tarea de clasificación se seleccionaron letras únicamente del corpus del Río de la Plata, con el objetivo de predecir si su año es mayor a 1950. Por lo tanto, se seleccionaron de forma aleatoria 2500 letras con el año menor a 1950 y la misma cantidad con el año mayor a 1950. A esta tarea la llamaremos "Año $>$ 1950"

Para la segunda tarea de clasificación se tomaron canciones del corpus del Río de la Plata y también canciones en castellano de musica.com, con el obje-

tivo de distinguir cuáles pertenecen a cada conjunto de datos. Se seleccionaron 6300 letras del corpus del Río de la Plata y la misma cantidad de letras en castellano extraídas de musica.com, seleccionadas de forma aleatoria. A esta tarea la llamaremos "Tango vs no tango".

Para las tareas de clasificación se utilizó un clasificador basado en SVM con función de núcleo lineal, a partir de la biblioteca `scikit-learn` de *python*. Los atributos sobre los que se aplica SVM son las representaciones obtenidas a partir de las letras como se mencionó en la Sección 4.1.3, calculadas a partir de cada modelo. A modo de comparación, también se utilizó la representación obtenida con un método basado en bolsa de palabras con los mismos tamaños de dimensión, en donde los atributos corresponden al número de ocurrencias de las palabras. Se realizó una clasificación mediante validación cruzada de 5 iteraciones.

En la Tabla 4.4 se pueden ver los resultados de las clasificaciones y también se indica la dimensión de las representaciones entre paréntesis. Es claro que la precisión de los modelos basados en redes es muy inferior al de bolsa de palabras, incluso en algún caso utilizando una dimensión menor. También es interesante ver que el Modelo Capa-simple parece tener un desempeño levemente superior que el Modelo Multicapa.

Tabla 4.4: Precisión de varios modelos en dos tareas de clasificación binaria.

Tarea	**Modelo Multicapa (256)**	**Modelo Capa-simple (512)**	**BoW (256)**	**BoW (512)**
Tango: año > 1950	0.51	0.54	0.60	**0.67**
No tango vs tango	0.64	0.74	0.84	**0.87**

4.3.2. Modelo multiplicative LSTM

Para comparar con los modelos anteriores, se entrenó otro modelo igual al presentado por Radford et al. (2017), ya que los autores muestran que su solución ofrece buenos resultados en varias tareas de clasificación. A este modelo lo llamaremos Modelo 4m-celdas.

Se utilizó una implementación del trabajo de Radford et al. (2017) basada en PyTorch[1]. En esta implementación se utiliza una variación de las celdas

[1]https://github.com/guillitte/pytorch-sentiment-neuron

LSTM llamadas *multiplicative LSTM* (mLSTM) (Krause et al., 2016). La diferencia de las celdas mLSTM respecto a las tradicionales es que mantienen varias matrices de pesos para el estado oculto (W_h) en función de la entrada. Según los autores las celdas mLSTM toman menos tiempo en aprender dando mejores resultados. La arquitectura utilizada presenta una única capa de 4096 celdas mLSTM y toma como entrada secuencias de caracteres de largo 128. En la Figura 4.3 se muestra un diagrama de esta red.

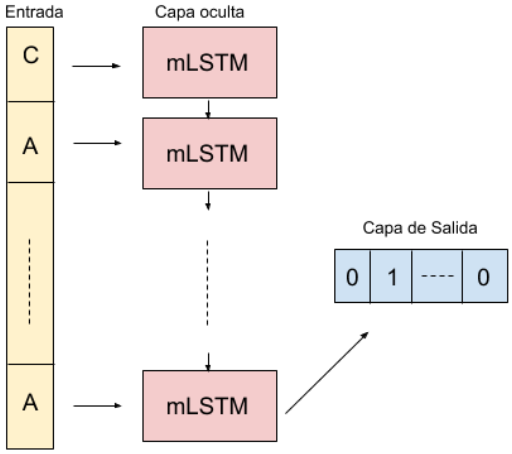

Figura 4.3: Diagrama de red multiplicative LSTM con 4096 celdas.

Al igual que para los modelos anteriores, se entrenó el Modelo 4m-celdas con todo el Conjunto de Entrenamiento y se tomó el valor de la función de perdida sobre el Conjunto de Validación. Se obtuvo una pérdida de **1.11**, dando un valor muy similar al reportado por Radford et al. (2017) por lo que se decidió compararlo con los modelos anteriores, sobre las mismas tareas de clasificación.

4.3.3. Resultados de los modelos

Al igual que para los modelos Multicapa y Capa-simple, se evaluó el Modelo 4m-celdas en las tareas de clasificación: "Año > 1950" y "Tango vs no tango". Para estas tareas se utilizaron clasificadores basados en SVM, mediante validación cruzada de 5 iteraciones. En la Tabla 4.5 se muestra el despeño del Modelo 4m-celdas y del modelo basado en bolsa de palabras.

En la Tabla 4.5 se puede ver una clara superioridad al utilizar las representaciones extraídas con el Modelo 4m-celdas con respecto al resto de los

modelos. En este caso se puede ver cómo el Modelo 4m-celdas supera también a las representaciones obtenidas mediante el método basado en Bolsa de palabras con una dimensión superior.

Tabla 4.5: Resultados preliminares de los modelos (precisión en dos tareas de clasificación).

Tarea	**Modelo Multicapa (256)**	**Modelo Capa-simple (512)**	**Modelo 4m-celdas (4096)**	**BoW (256)**	**BoW (512)**	**BoW (8192)**
Tango: año > 1950	0.51	0.54	**0.75**	0.60	0.67	0.73
No tango vs tango	0.64	0.74	**0.96**	0.84	0.87	0.92

Debido a los buenos resultados obtenidos con el Modelo 4m-celdas, en el siguiente capítulo se evalúa en varias tareas de clasificación el desempeño en comparación con el método basado en Bolsa de Palabras.

4.4. Conclusiones

En este capítulo se describió el funcionamiento de las redes recurrentes y en particular las redes LSTM. Además, se detalló cómo fue aplicado el método propuesto por Radford et al. (2017) sobre las letras de canciones. Para aplicar el método fue necesario recolectar un gran número de letras de canciones en castellano, lo cual también fue detallado en este capítulo.

Luego, se evaluaron preliminarmente distintas arquitecturas de redes, obteniéndose un resultado superior utilizando la misma arquitectura planteada por Radford et al. (2017). Por lo tanto, en el próximo capítulo se realiza una evaluación más exhaustiva, utilizando las representaciones de las letras obtenidas mediante el Modelo 4m-celdas para el corpus de letras del Río de la Plata. La evaluación consiste en utilizar las representaciones en distintas tareas de clasificación y comparar los resultados con los de otro modelo basado en Bolsa de Palabras.

Por último, vale la pena mencionar que el proceso por el cual se evaluaron diferentes arquitecturas no es para nada intuitivo, ya que es muy difícil obtener información durante el entrenamiento sobre la calidad que tendrán las representaciones. Por lo tanto, obtener la arquitectura óptima es una tarea que lleva mucho tiempo y recursos. Además, requiere tener una cierta intuición sobre el comportamiento de las redes.

Capítulo 5

Evaluación

En el Capítulo 4 se describe un modelo que permite obtener representaciones a partir de las letras basado en redes recurrentes de 4096 unidades de tipo LSTM (Modelo 4m-celdas), ofreciendo buenos resultados en algunas tareas de clasificación. En este capítulo se evalúa de forma más exhaustiva el desempeño del Modelo 4m-celdas y se compara con una representación de las letras basada en Bolsa de Palabras utilizando las ocurrencias de las palabras como atributos.

Se entiende que si las representaciones obtenidas ofrecen un buen desempeño en una tarea de clasificación, entonces mantienen la similitud de las canciones con respecto a la información utilizada para clasificar. Por lo tanto, se evalúa la precisión en cinco tareas de clasificación diferentes. Las tareas consisten en clasificar las letras a partir de su letrista, época, género y año.

Al igual que en el capítulo anterior, para cada tarea de clasificación, la precisión se calculada de la siguiente manera:

$$precisión = \frac{\text{cantidad de aciertos}}{\text{cantidad de elementos a clasificar}} \tag{5.1}$$

Para las tareas de clasificación se utilizan las 6300 letras de canciones en el corpus de letras del Río de la Plata, obtenido en el Capítulo 3 a partir de distintas fuentes. También se utilizan algunos metadatos de las canciones para generar las etiquetas (por ejemplo, género, año o letrista). En cada sección de este capítulo se describe una tarea de clasificación realizada y se menciona el número de letras y los metadatos utilizados.

En las cinco tareas de clasificación se utilizaron clasificadores basados en regresión logística con regularización de tipo L2. Para el caso del Modelo 4m-celdas, los atributos sobre los que se aplica el clasificador son las represen-

taciones obtenidas siguiendo el mismo procedimiento detallado en la Sección 4.1.3. del capítulo anterior. Mientras que para el método basado en Bolsa de Palabras se utiliza las ocurrencias de las palabras en la letra de la canción.

5.1. Grupo de letrista

La primera tarea consistió en identificar automáticamente los letristas de las canciones. Dado que en el corpus de letras del Río de la Plata los letristas tienen solamente cuatro canciones en promedio cada uno, para simplificar la tarea, se decidió agruparlos a partir de sus colaboraciones. Se agruparon los artistas si realizaron juntos una canción y se repitió este proceso uniendo los grupos hasta que no se modificaran más, luego se descartaron los grupos con menos de cinco artistas. De esta forma se obtuvieron en total seis grupos de letristas. En la Tabla 5.1 se indica la cantidad de canciones relacionadas a cada grupo de letristas, sumando un total de 1573 letras.

Para predecir el grupo correspondiente a cada canción a partir de su letra, se entrenó un clasificador basado en regresión logística, utilizando una regularización de tipo L2. Por tratarse de un conjunto de datos tan reducido, se utilizó validación cruzada de cinco iteraciones.

Dado que se cuenta con múltiples clases para clasificar, se utiliza la estrategia *one-vs-rest*. Esta estrategia se basa en entrenar un clasificador binario para cada una de las seis clases y en función de la probabilidad retornada por cada uno se predice la clase de la canción.

Los resultados para esta tarea se muestran en la Tabla 5.2. Puede verse que el modelo basado en redes neuronales ofrece una mayor precisión.

Tabla 5.1: Cantidad de canciones con letra para cada grupo de letristas

	Grupo 1	**Grupo 2**	**Grupo 3**	**Grupo 4**	**Grupo 5**	**Grupo 6**
# Canciones	264	526	349	155	152	127

Tabla 5.2: Precisión de los modelos al predecir el grupo de letristas de las canciones a partir de la letra

LSTM	**BOW**	**RND**
0.50	0.46	0.33

5.2. Épocas

Aunque entre los historiadores del tango existe un consenso en cuanto a las etapas que marcan distintos estilos, no está del todo claro los años que comprenden las etapas. En parte esto se debe a que músicos con una presencia muy marcada en un estilo siguieron componiendo en otras etapas y por lo tanto, no es posible establecer una división tan fuerte.

De todas formas, simplificando el problema, en este trabajos vamos a seguir las épocas que se marcan en el museo del tango, según la Academia Nacional del Tango de la República Argentina. Allí se definen las siguientes cuatro épocas[1]:

- Guardia Vieja (- 1925)
- Guardia Nueva (1925 - 1955)
- Vanguardia (1955 - 1970)
- Contemporáneo (1970 - 2017)

La siguiente tarea corresponde a la clasificación de las canciones según su letra a partir de los cuatro períodos anteriormente mencionados. En la Tabla 5.3 se indica la cantidad de canciones en cada período de tiempo.

Para entrenar y evaluar la clasificación, en este caso también se realizó una validación cruzada de cinco iteraciones y se utilizó el método *one-vs-rest*. Los resultados en la Tabla 5.4 muestran un desempeño levemente superior del modelo basado en redes neuronales. Los modelos basados en Bolsa de Palabras y basado en redes neuronales muestran un desempeño bastante superior a un clasificador aleatorio.

Tabla 5.3: Cantidad de canciones con letra para cada época

	Guardia Vieja	**Guardia Nueva**	**Vanguardia**	**Contemporáneo**
# Canciones	437	2953	772	1506

Tabla 5.4: Precisión de los modelos al predecir la época de las canciones a partir de la letra

LSTM	**BOW**	**RND**
0.66	0.65	0.52

En la Tabla 5.5 se muestra una matriz de confusión producida por el clasificador basado en bolsa de palabras. Mientras que en la Tabla 5.6 se muestra

[1] De acuerdo a https://es.wikipedia.org/wiki/Tango#Las_etapas_del_tango

la matriz de confusión producida por el clasificador basado en redes profundas.

Tabla 5.5: Matriz de confusión para el clasificador al predecir la época de las canciones a partir de la letra, utilizando la representación obtenida con bolsa de palabras.

real/clasificado	contemporáneo	nueva	vanguardia	vieja
contemporáneo	935	551	20	0
nueva	202	2729	21	1
vanguardia	177	573	22	0
vieja	32	403	0	2

Tabla 5.6: Matriz de confusión para el clasificador al predecir la época de las canciones a partir de la letra, utilizando la representación obtenida con la red neuronal.

real/clasificado	contemporáneo	nueva	vanguardia	vieja
contemporáneo	1011	461	31	3
nueva	240	2638	66	9
vanguardia	195	517	59	1
vieja	36	387	7	7

A partir de las matrices de confusión se puede ver que Guardia Vieja es la época que produce mayor cantidad de errores en ambos casos. Para el caso de Bolsa de palabras, solamente tres canciones se clasificaron dentro de esta época. Para el caso del modelo basado en la red neuronal, se clasificaron 20 dentro de Guardia Vieja con siete aciertos.

5.3. Géneros

Como ya fue mencionado en el Capítulo 3, el conjunto de canciones presenta distintos géneros. Mayoritariamente se encuentra presente el tango, pero también se tomaron las canciones de milonga y vals dado que son las siguientes en frecuencia. En esta tarea se intenta predecir el género a partir de la letra. Se realizó una selección de 460 canciones de cada género para realizar el entrenamiento y la evaluación de los clasificadores. Se repitió el proceso 10 veces con distintas canciones y se calculó el promedio de todos los resultados.

En la Tabla 5.7 se muestra la precisión de los modelos. En esta tarea también es levemente superior el modelo basado en redes neuronales en comparación al modelo basado en bolsa de palabras, ambos modelos se encuentran muy por encima de un clasificador aleatorio.

Tabla 5.7: Precisión de los modelos al predecir el género de las canciones a partir de la letra

LSTM	BOW	RND
0.64	0.63	0.33

5.4. Período de año binario

En esta tarea se distinguieron las canciones que presentaban un año anterior a 1945 contra las que presentaban año posterior a 1955.

Se seleccionaron un total de 4556 letras del corpus del Río de la Plata, igualmente distribuidas en cada período de tiempo. Además de utilizar las 4556 letras, en esta tarea también se quería evaluar el desempeño de los modelos al utilizar una cantidad menor de letras. Por lo tanto, se evaluó con 1000, 2000 y 4556 canciones. En la Tabla 5.8 se muestra el resultado para cada conjunto de datos. Podemos ver que en todos los casos el modelo basado en redes neuronales tiene una precisión mayor que el basado en bolsa de palabras. Ambos muestran un desempeño muy superior a un clasificador aleatorio.

Tabla 5.8: Precisión de los modelos al predecir el año de las canciones a partir de la letra

cantidad	LSTM	BOW	RND
4556	**0.77**	0.76	0.50
2000	**0.75**	0.74	0.50
1000	**0.72**	0.71	0.50

A continuación, se seleccionaron los cinco *features* más relevantes del modelo mediante el método basado en el análisis de varianza (ANOVA) de la medida F. Luego, se utilizaron únicamente los mejores cinco *features* para clasificar el período del año mediante un clasificador basado en regresión logística. En la Tabla 5.9 se muestran los resultados para esta tarea, en donde se puede ver que la diferencia no es tan grande con respecto a un clasificador aleatorio. De todas formas, el modelo basado en redes neuronales tiene una mayor precisión.

Tabla 5.9: Precisión de los modelos al predecir el año de las canciones a partir de la letra utilizando solo 5 features

cantidad	LSTM (5 feat)	BOW (5 feat)	RND
4556	**0.61**	0.58	0.50

5.5. Tango o no tango

Por último, se seleccionaron letras de canciones por fuera del conjunto de datos de forma que se puedan comparar las letras de tango con otros géneros más diversos. El resto de las canciones fueron seleccionadas de forma aleatoria de musica.com, todas son canciones es Castellano pero de distintos géneros.

Para esta tarea también se quiso evaluar el desempeño de los modelos utilizando conjuntos de datos de distintos tamaños. Por lo tanto, se utilizaron conjuntos de 1000 letras, 2000 y 8738 letras, en todos los casos distribuidas de forma equitativa entre las dos clases. Para este caso se utilizó un clasificador basado en regresión logística realizando el mismo procedimiento que en las tareas anteriores.

En la Tabla 5.10, se puede ver que en todos los casos el desempeño del modelo basado en redes neuronales es superior al modelo basado en bolsa de palabras. Los dos modelos obtienen una precisión cercana al 95 % cuando se usan 8738 letras, muy por encima del clasificador aleatorio.

Tabla 5.10: Desempeño de los modelos al distinguir canciones de tango a partir de la letra, utilizando distintos tamaños de conjuntos. Se indica la cantidad de letras utilizadas y la medida de precisión de cada modelo.

cantidad	LSTM	BOW	RND
8738	**0.96**	0.93	0.50
2000	**0.94**	0.92	0.50
1000	**0.92**	0.90	0.50

Finalmente, también se realizó la evaluación con los cinco features más relevantes según el análisis de varianza. Los resultados se muestran en la Tabla 5.11. En este caso se puede ver que la precisión del modelo basado en redes neuronales es muy superior a la del modelo basado en bolsa de palabras y al clasificador aleatorio.

Tabla 5.11: Precisión de los modelos al distinguir canciones de tango a partir de la letra utilizando solo 5 features

cantidad	LSTM (5 feat)	BOW (5 feat)	RND
8738	**0.81**	0.70	0.50

5.6. Conclusiones

Se realizaron cinco tareas de clasificación utilizando las representaciones obtenidas con el Modelo 4m-celdas y se comparó con representaciones obtenidas mediante el método Bolsa de Palabras.

Si bien no es claro con qué precisión se pueden realizar las tareas planteadas por un humano, algunas parecen más simples que otras. Por ejemplo, la tarea Tango vs no-tango parece ser la más fácil, mientras que identificar el género parece la más difícil. Es posible que las tareas más complejas tengan un techo de precisión más cercano a un clasificador aleatorio.

En todas las tareas planteadas, aunque la diferencia entre los modelos en general no es muy considerable, en todos los casos el modelo basado en redes neuronales presenta una mayor precisión que el modelo basado en bolsa de palabras. Por lo tanto, entendemos que la representación obtenida a partir de dicho modelo conserva la similitud de las canciones y podemos concluir que permite ser utilizado en distintas aplicaciones con objetivos diversos.

En el siguiente capítulo se muestra una aplicación de dichas representaciones, con el objetivo de ofrecer una interfaz intuitiva de navegación sobre artistas y canciones basado en el aprendizaje automático realizado sobre el corpus del Río de la Plata.

Capítulo 6

Aplicación

En este capítulo se describe una aplicación desarrollada con el objetivo de mostrar la utilidad de las representaciones de las letras que fueron obtenidas con el Modelo 4m-celdas. La aplicación permite recorrer el corpus del Río de la Plata recolectado en el Capítulo 3, en base a diferentes criterios de búsqueda y de similitud entre las canciones o los artistas.

Se implementó una interfaz web que ofrece estas funcionalidades de forma intuitiva para el usuario. A partir de la interfaz, es posible comprobar la similitud entre las canciones a partir de distintos aspectos, ya sea a partir de sus datos (año, artista, época o género) o de las representaciones calculadas. En la Figura 6.1 se muestra la pantalla principal de la interfaz web.

6.1. Búsqueda de similitud entre las canciones

Se entiende que si las canciones tienen un atributo en común (por ejemplo, el género o la época) entonces son similares con respecto a ese atributo. En esta sección, se describe la forma de recorrer el corpus a partir de la similitud de las canciones con respecto a distintos atributos. De esta forma, es posible comprobar la utilidad de las representaciones obtenidas por medio de una aplicación.

6.1.1. Año o época

Se entiende que uno de los aspectos que puede mostrar similitudes entre las canciones es la cercanía en el tiempo. En el caso del tango, los historiadores definen cuatro épocas que marcan distintos estilos dentro del género. Por lo

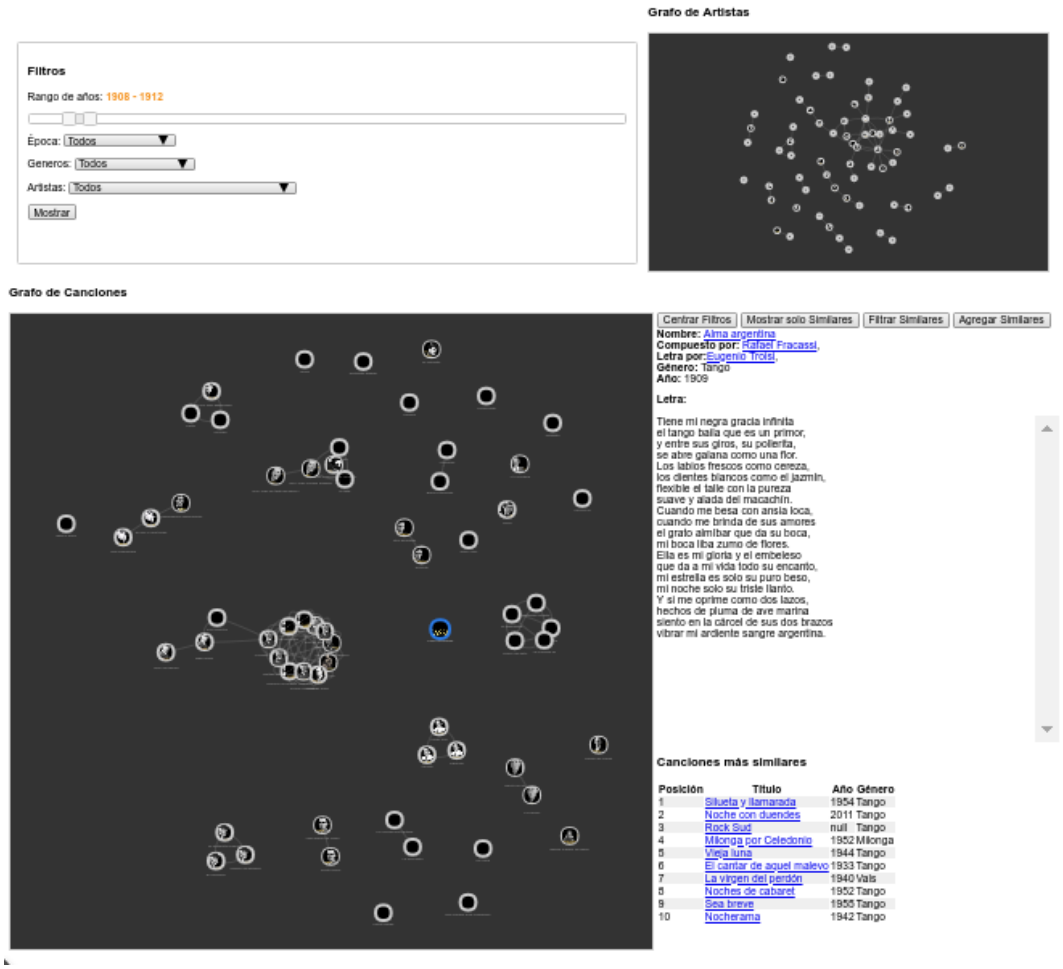

Figura 6.1: Pantalla principal de la interfaz web completa.

tanto, es interesante explorar todas las canciones del corpus que fueron creadas dentro de los mismos años o dentro de la misma época.

Con este objetivo, en la interfaz web se permite filtrar las canciones del corpus a partir de su año o época. En la Figura 6.2 se muestran las opciones de filtros en la interfaz web.

Como se muestra en la Figura 6.1, una vez aplicado el filtro de año, se muestran las canciones que cumplen con estos atributos y los artistas correspondientes, ambos en forma de grafo.

6.1.2. Artista

Otro aspecto que se puede utilizar para encontrar relaciones entre las canciones son los artistas.

Por un lado, se pueden explorar todas las canciones en las que un artista

Figura 6.2: Sección de la aplicación web que permite seleccionar los filtros de resultados.

participó; ya sea como compositor o letrista. En la interfaz web se permite obtener todas las canciones de un artista aplicando el filtro correspondiente que se muestra en la Figura 6.2.

Por otro lado, dado un conjunto de canciones, se pueden ver las relaciones entre las canciones a partir de los artistas que tienen en común. Esta información también se puede utilizar en sentido inverso, es decir, se pueden ver las relaciones entre los artistas a partir de las canciones que tienen en común.

En la interfaz web, dado un conjunto de canciones previamente filtrados, se muestra un grafo en donde las conexiones están dadas por artistas que tienen en común. En la Figura 6.3 se muestra el grafo de canciones a partir del filtro de año. Por lo tanto, allí se puede ver la relación entre las canciones a partir de artistas y año.

Además, en la interfaz web se puede ver un grafo con los artistas correspondientes a las canciones previamente filtradas. En la Figura 6.4 se muestra el grafo de artistas en donde las relaciones están determinadas por todas las canciones que tienen en común.

6.1.3. Género

Para explorar el corpus se puede utilizar la información del género. Como ya fue mencionado, el género musical es un aspecto de alto nivel que define una relación de similitud entre las canciones.

En la interfaz web se permite obtener todas las canciones de un cierto género aplicando el filtro correspondiente que se muestra en la Figura 6.2. Por

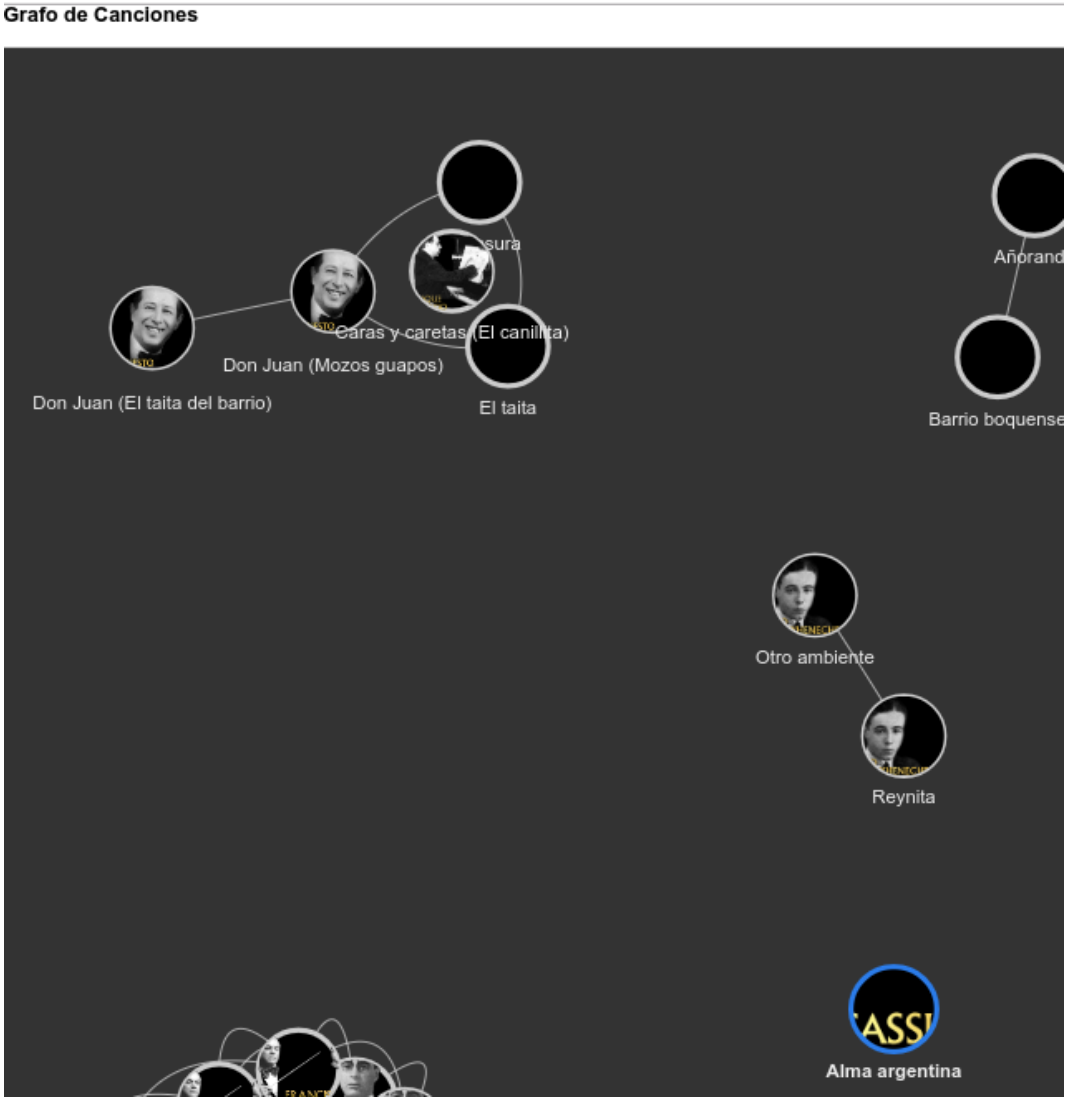

Figura 6.3: Sección Grafo de canciones.

ejemplo, mediante los filtros se pueden obtener todas las canciones de un año específico y del género milonga.

6.1.4. Letra

Basado únicamente en las representaciones obtenidas utilizando del Modelo 4m-celdas, es posible obtener las canciones más similares a partir de la letra. Por lo tanto, una vez identificada una canción, se puede analizar su letra y compararla con la letra de sus similares, de esta forma es posible ver la utilidad de las representaciones obtenidas. Además, es posible comparar los datos de las canciones más similares (año, género y título) para identificar aspectos en

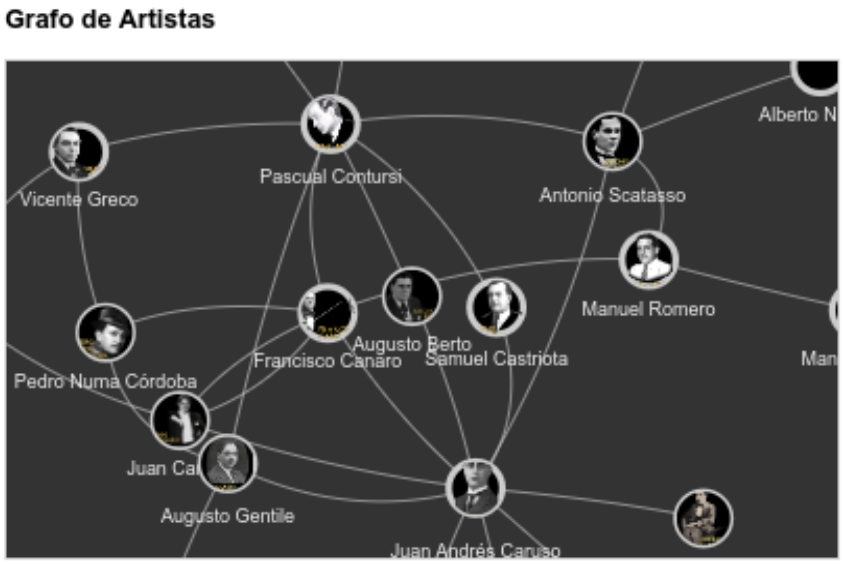

Figura 6.4: Sección Grafo de artistas.

común entre las canciones.

Utilizando la interfaz web se puede seleccionar una canción en el grafo y se muestran las 10 canciones más similares calculadas mediante la distancia coseno de las representaciones obtenidas. En la Figura 6.5 se muestran las canciones más similares para la canción "Alma argentina". También se muestra: título, año y género de la canción, de esta forma es posible ver si hay alguna relación entre los datos del elemento seleccionado y las canciones más similares.

6.1.5. Relaciones entre varios aspectos

El principal objetivo de este trabajo es obtener una representación de las letras que mantenga la similitud de las canciones. Con la motivación de mostrar la utilidad de las representaciones obtenidas, es posible buscar relaciones entre las canciones comparando varios de los atributos (por ejemplo: artista, año, época o género) y las letras.

Mediante la interfaz web, dada una canción, se permite combinar la lista de sus 30 canciones más similares con los resultados de la búsqueda. Para esto se utiliza el menú que se muestra en la Figura 6.5, de las dos formas que se describen a continuación.

Una vez seleccionada una canción en el grafo y haciendo *click* sobre la opción "Mostrar solo similares", se mostrarán en la sección "Grafo de can-

Figura 6.5: Sección correspondiente a los datos de la canción seleccionada.

ciones" solamente las 30 canciones más similares a la seleccionada (utilizando las representaciones calculadas a partir de la red). Mediante esta opción es

posible visualizar las relaciones entre las canciones más similares a partir de las conexiones entre los nodos. En la Figura 6.6 se muestran las canciones más similares para la canción "Alma argentina", allí se puede ver una relación dada por los artistas de tres canciones dentro las similares.

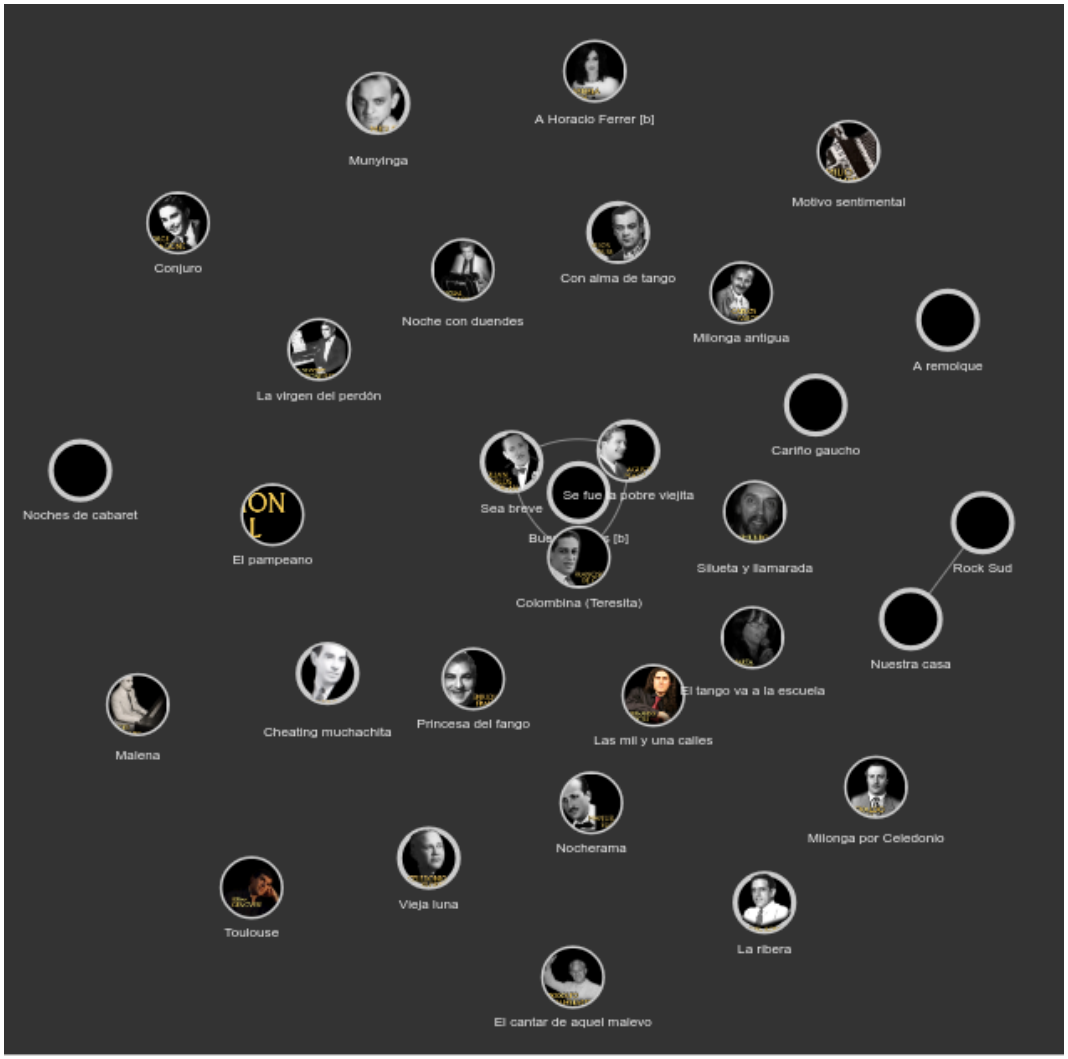

Figura 6.6: Sección correspondiente al grafo de canciones, mostrando las 30 canciones más similares a "Alma Argentina".

Otra forma de ver relaciones es utilizando la opción "Filtrar similares". Utilizando esta opción se muestra en la sección "Grafo de canciones" solamente la intersección entre las 30 canciones más similares a la canción seleccionada y las canciones resultantes de aplicar los filtros. De esta forma se pueden comparar distintos aspectos. Por ejemplo, se puede aplicar un filtro para una cierto año y luego visualizar el grafo con las canciones más similares del año indicado.

6.2. Conclusiones

En este capítulo se muestra el uso de las representaciones de las letras obtenidas a partir del Modelo 4m-celdas, con el objetivo de permitir una búsqueda en la base de datos teniendo en cuenta la similitud entre las canciones. También se muestran distintas formas de identificar relaciones entre los aspectos de las canciones, a partir de su artista, género, año, época y letra. Por lo tanto, se puede comprobar la utilidad de los métodos aplicados al comparar las relaciones entre los diferentes atributos de las canciones más similares.

En este capítulo también se describe la interfaz web que permite realizar estas consultas de forma intuitiva sobre el corpus de canciones del Río de la Plata. Por lo tanto, esta interfaz es un aporte para la comunidad ya que permite difundir y preservar la tradición musical de los géneros del Río de la Plata.

Capítulo 7

Conclusiones y trabajo a futuro

En este trabajo, luego de analizar algunas técnicas del procesamiento del lenguaje natural y cómo se pueden aplicar a las letras de música, se profundizó sobre algunas técnicas enfocadas en redes neuronales profundas para extraer representaciones que permitan conservar la similitud entre las canciones.

Para evaluar la utilidad de las representaciones, se las utilizó en varias tareas de clasificación y se comparó el desempeño con otra técnica clásica del procesamiento de lenguaje natural. Si bien no se apreció una diferencia significativa en los resultados entre los métodos, en todos los casos el modelo basado en redes neuronales presentó un mejor desempeño que el modelo basado en bolsa de palabras. Mediante ambas representaciones se superó por un buen margen a un clasificador aleatorio en todas las tareas analizadas. Por lo tanto, se puede afirmar que la representación obtenida a partir del modelo basado en redes neuronales conserva información de similitud entre las letras y permite ser utilizado en distintas aplicaciones.

Este trabajo se enfocó en letras de canciones del Río de la Plata. Por lo tanto, fue necesario recolectar un corpus de letras lo suficientemente completo para las tareas que se debían realizar. Uno de las principales aportes de este trabajo es la publicación de los datos utilizados. La publicación de estos datos permitirá a otros investigadores realizar trabajos sobre los géneros antes mencionados, lo cual es positivo para poder difundir y conservar la tradición musical del Río de la Plata.

También se presentó una aplicación web que permite navegar por el conjunto de datos obtenido e interactuar con las representaciones obtenidas. Las

representaciones fueron utilizadas para medir la similitud entre las canciones y para seleccionar un conjunto de canciones recomendadas para cada canción. Se entiende que la herramienta será un aporte interesante para difundir y conservar estas canciones.

Por último, muchas de las técnicas detalladas en este trabajo no habían sido aplicadas en letras de música en español, por lo que los resultados reportados en esta tesis son novedosos en este sentido. Las distintas técnicas de representación basadas en el aprendizaje profundo se encuentran en una etapa de desarrollo y este tipo de trabajos permiten compararlos con otras técnicas más exploradas.

7.1. Trabajo a futuro

Como trabajo a futuro, sería interesante profundizar sobre algunas técnicas analizadas en el Capítulo 2. Por ejemplo, siguiendo el método utilizado por Oramas et al. (2016a,b), sería interesante aplicar técnicas para extraer información sobre texto relacionado con los artistas de los géneros del Río de la Plata, por ejemplo:

- texto extraído de artículos disponible en Todotango relacionados con los artistas
- texto extraído de Wikipedia sobre los artistas en el corpus
- sobre libros digitalizados relacionados con estos estilos musicales

A partir de la información extraída se construiría un grafo de conocimiento, en donde se podrían identificar relaciones entre los artistas y se podrían obtener representaciones con mayor información semántica.

También parece interesante profundizar en distintas formas de combinar audio o partituras con las letras para obtener representaciones más ricas de las canciones. Muchos trabajos mencionados en el Capítulo 2 combinan distintas fuentes y siempre se logra mejorar los resultados.

En esta tesis, las redes neuronales recurrentes entrenadas para generar texto fueron utilizadas con el objetivo de obtener las representaciones, pero este es un producto interesante que podría ser explotado. Por ejemplo, se podría utilizar como herramienta para asistir a letristas en la creación de nuevas letras.

Por último, vale la pena mencionar que este trabajo se enfocó en el análisis de las técnicas que tienen como objetivo obtener una representación de las letras, pero existen muchas aplicaciones específicas para estas representaciones

que requieren un estudio particular. Por ejemplo, en el caso de la recomendación, actualmente las técnicas de filtrado colaborativo son las que producen los mejores resultados, pero presentan el problema de estar muy influenciados por los artistas o canciones más populares. Por lo tanto, las técnicas que se basan en el contenido de la música para recomendar pueden mitigar este problema. La combinación de las representaciones obtenidas con filtrado colaborativo, es un enfoque para evaluar con el objetivo de mejorar la recomendación de música.

Referencias bibliográficas

Ars Technica (2016). Sunspring. the first film ever written entirely by an artificial intelligence. http://www.thereforefilms.com/sunspring.html. Accessed: 2016-06-09.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. (2011). The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

Bollen, D., Knijnenburg, B. P., Willemsen, M. C., and Graus, M. (2010). Understanding choice overload in recommender systems. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 63–70. ACM.

Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

Chollet, F. (2017). *Deep Learning with Python*. Manning Publications Co., Greenwich, CT, USA, 1st edition.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227.

dos Santos, C. L. and Silla, C. N. (2015). The latin music mood database. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):23.

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., and Harshman, R. (1988). Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285. Acm.

Ellis, R. J., Xing, Z., Fang, J., and Wang, Y. (2015). Quantifying lexical novelty in song lyrics. In *ISMIR*, pages 694–700.

Esparza, T. M., Bello, J. P., and Humphrey, E. J. (2015). From genre classification to rhythm similarity: Computational and musicological insights. *Journal of New Music Research*, 44(1):39–57.

Espinosa-Anke, L., Oramas, S., Saggion, H., and Serra, X. (2017). Elmdist: A vector space model with words and musicbrainz entities. In *European Semantic Web Conference*, pages 355–366. Springer.

Fell, M. and Sporleder, C. (2014). Lyrics-based analysis and classification of music. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 620–631.

Fu, Z., Lu, G., Ting, K. M., and Zhang, D. (2011). A survey of audio-based music classification and annotation. *IEEE Transactions on Multimedia*, 13(2):303–319.

Google AI (2015). Inceptionism: Going deeper into neural networks. https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html. Accessed: 2015-06-17.

Gouyon, F., Dixon, S., Pampalk, E., and Widmer, G. (2004). Evaluating rhythmic descriptors for musical genre classification. In *Proceedings of the AES 25th International Conference*, pages 196–204.

Govaerts, S. and Duval, E. (2009). A web-based approach to determine the origin of an artist. In *Proceedings of ISMIR2009: 10th International Society for Music Information Retrieval Conference*, pages 261–266. ISMIR-The International Society for Music Information Retrieval.

Hill, F., Cho, K., and Korhonen, A. (2016). Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.

Hinton, G. E. (1986). Learning distributed representations of concepts.

Hirjee, H. and Brown, D. (2010). Using automated rhyme detection to characterize rhyming style in rap music.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc.

Howard, S., Jr, C. N. S., and Johnson, C. G. (2011). Automatic lyrics-based music genre classification in a multilingual setting. In *Thirteenth Brazilian Symposium on Computer Music*.

Hu, X. and Downie, J. S. (2010a). Improving mood classification in music digital libraries by combining lyrics and audio. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, JCDL '10, pages 159–168, New York, NY, USA. ACM.

Hu, X. and Downie, J. S. (2010b). When lyrics outperform audio for music mood classification: A feature analysis. In *ISMIR*, pages 619–624.

Hu, X., Downie, J. S., and Ehmann, A. F. (2009). Lyric text mining in music mood classification. *American music*, 183(5,049):2–209.

IFPI (2016). Ifpi global music report 2016. http://www.ifpi.org/news/IFPI-GLOBAL-MUSIC-REPORT-2016. Accessed: 2016-04-12.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579.

Jurafsky, D. and Martin, J. H. (2009). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.

Karpathy, K. (2015). The unreasonable effectiveness of recurrent neural networks. http://karpathy.github.io/2015/05/21/rnn-effectiveness/. Accessed: 2015-05-21.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Knees, P., Pampalk, E., and Widmer, G. (2004a). Artist classification with web-based data. In *ISMIR*.

Knees, P., Pampalk, E., and Widmer, G. (2004b). Automatic classification of musical artists based on web-data. *Oesterreichische Gesellschaft fuer Artificial Intelligence*, 24(1).

Knees, P. and Schedl, M. (2011). Towards semantic music information extraction from the web using rule patterns and supervised learning. In *Workshop on Music Recommendation and Discovery*, pages 18–25.

Krause, B., Lu, L., Murray, I., and Renals, S. (2016). Multiplicative lstm for sequence modelling. *arXiv preprint arXiv:1609.07959*.

Kroher, N., Díaz-Báñez, J.-M., Mora, J., and Gómez, E. (2016). Corpus cofla: A research corpus for the computational study of flamenco music. *J. Comput. Cult. Herit.*, 9(2):10:1–10:21.

Laurier, C., Grivolla, J., and Herrera, P. (2008). Multimodal music mood classification using audio and lyrics. In *2008 Seventh International Conference on Machine Learning and Applications*, pages 688–693.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Li, T. and Ogihara, M. (2004). Music artist style identification by semi-supervised learning from both lyrics and content. In *ACM Multimedia*.

Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

Logan, B., Kositsky, A., and Moreno, P. (2004). Semantic analysis of song lyrics. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 2, pages 827–830. IEEE.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Mahedero, J. P., MartÍnez, Á., Cano, P., Koppenberger, M., and Gouyon, F. (2005). Natural language processing of lyrics. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 475–478. ACM.

Mann, Y. (2017). Ai duet. https://experiments.withgoogle.com/ai/ai-duet. Accessed: 2017-05-09.

Marchand, U. and Peeters, G. (2016). The Extended Ballroom Dataset. Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conf.. 2016.

Mayer, R., Neumayer, R., and Rauber, A. (2008a). Combination of audio and lyrics features for genre classification in digital audio collections. In *Proceedings of the 16th ACM International Conference on Multimedia*, MM '08, pages 159–168, New York, NY, USA. ACM.

Mayer, R., Neumayer, R., and Rauber, A. (2008b). Rhyme and style features for musical genre classification by song lyrics.

McKay, C., Burgoyne, J. A., Hockman, J., Smith, J. B., Vigliensoni, G., and Fujinaga, I. (2010). Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features. In *ISMIR*, pages 213–218.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Mikolov, T., Kombrink, S., Burget, L., Černockỳ, J., and Khudanpur, S. (2011). Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Olah, C. (2015). Understanding lstm networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed: 2015-08-27.

Oramas, S., Espinosa-Anke, L., Lawlor, A., et al. (2016a). Exploring customer reviews for music genre classification and evolutionary studies. In *The 17th International Society for Music Information Retrieval Conference (ISMIR 2016), New York City, United States of America, 7-11 August 2016*.

Oramas, S., Espinosa-Anke, L., Sordo, M., Saggion, H., and Serra, X. (2016b). Information extraction for knowledge base construction in the music domain. *Data and Knowledge Engineering*, 106:70 – 83.

Oramas, S., Gómez, F., Gómez Gutiérrez, E., and Mora, J. (2015a). Flabase: Towards the creation of a flamenco music knowledge base. In *Müller M, Wiering F, editors. ISMIR 2015. 16th International Society for Music Information Retrieval Conference; 2015 Oct 26-30; Málaga, Spain. Canada: ISMIR; 2015.* International Society for Music Information Retrieval (ISMIR).

Oramas, S., Nieto, O., Barbieri, F., and Serra, X. (2017a). Multi-label music genre classification from audio, text, and images using deep features. *arXiv preprint arXiv:1707.04916*.

Oramas, S., Nieto, O., Sordo, M., and Serra, X. (2017b). A deep multimodal approach for cold-start music recommendation. *arXiv preprint arXiv:1706.09739*.

Oramas, S., Sordo, M., Anke, L. E., and Serra, X. (2015b). A semantic-based approach for artist similarity. In *ISMIR*, pages 100–106.

Parra, F. L. and León, E. (2013). Unsupervised tagging of spanish lyrics dataset using clustering. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 130–143. Springer.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Pohle, T., Knees, P., Schedl, M., and Widmer, G. (2007). Building an interactive next-generation artist recommender based on automatically derived high-level concepts. In *2007 International Workshop on Content-Based Multimedia Indexing*, pages 336–343.

Radford, A., Jozefowicz, R., and Sutskever, I. (2017). Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

Repetto, R. C. and Serra, X. (2014). Creating a corpus of jingju (beijing opera) music and possibilities for melodic analysis. In *15th International Society for Music Information Retrieval Conference*, pages 313–318, Taipei, Taiwan.

Ribeiro, R. P., Almeida, M. A., and Silla Jr, C. N. (2014). The ethnic lyrics fetcher tool. *EURASIP Journal on Audio, Speech, and Music Processing*, 2014(1):27.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Schedl, M. (2010). On the use of microblogging posts for similarity estimation and artist labeling. In *ISMIR*, pages 447–452. Citeseer.

Schedl, M., Knees, P., and Widmer, G. (2005). A web-based approach to assessing artist similarity using co-occurrences. In *Proceedings of the Fourth International Workshop on Content-Based Multimedia Indexing (CBMI'05)*.

Schedl, M. and Widmer, G. (2008). Automatically detecting members and instrumentation of music bands via web content mining. In Boujemaa, N., Detyniecki, M., and Nürnberger, A., editors, *Adaptive Multimedia Retrieval: Retrieval, User, and Semantics*, pages 122–133, Berlin, Heidelberg. Springer Berlin Heidelberg.

Schwartz, B. (2003). *The Paradox of Choice: Why More Is Less*. Harper Perennial. HarperCollins.

Silla Jr, C. N., Koerich, A. L., and Kaestner, C. A. (2008). The latin music database. In *ISMIR*, pages 451–456.

Sordo, M., Oramas, S., and Espinosa-Anke, L. (2015). Extracting relations from unstructured text sources for music recommendation. In Biemann,

C., Handschuh, S., Freitas, A., Meziane, F., and Métais, E., editors, *Natural Language Processing and Information Systems*, pages 369–382, Cham. Springer International Publishing.

Srinivasamurthy, A., Koduri, G. K., Gulati, S., Ishwar, V., and Serra, X. (2014). Corpora for music information research in indian art music. In *International Computer Music Conference/Sound and Music Computing Conference*, pages 1029–1036, Athens, Greece.

Sterckx, L., Demeester, T., Deleu, J., Mertens, L., and Develder, C. (2014). Assessing quality of unsupervised topics in song lyrics. In *European Conference on Information Retrieval*, pages 547–552. Springer.

Sturm, B. L. (2014). A survey of evaluation in music genre recognition. In Nürnberger, A., Stober, S., Larsen, B., and Detyniecki, M., editors, *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*, pages 29–66, Cham. Springer International Publishing.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302.

Uyar, B., Atlı, H. S., Şentürk, S., Bozkurt, B., and Serra, X. (2014). A corpus for computational research of turkish makam music. In *1st International Workshop on Digital Libraries for Musicology*, pages 1–7, London, UK.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Whitman, B. and Lawrence, S. (2002). Inferring descriptions and similarity for music from community metadata. In *ICMC*.

Whitman, B. and Smaragdis, P. (2002). Combining musical and cultural features for intelligent style detection.

ANEXOS

Anexo 1

Encuesta

En el Capítulo 3 se utiliza una primera representación de las letras para distintas tareas con el objetivo de analizar el corpus. En esta primera instancia exploratoria se deseaba tener una idea sobre la calidad de la representación. Por lo tanto, se decidió evaluar la representación utilizada consultando un experto del dominio.

Se le solicitó al experto que indicara un conjunto de canciones que considerara buenas, otro conjunto que considerara malas y otro conjunto que considerara como muy populares. Las canciones seleccionadas por el experto están detalladas en las tablas 1.1, 1.2 y 1.3. Luego, partir de estas canciones se buscaron las dos letras más similares para cada una y se le entregó al experto sin indicarle a cuales correspondían, en la Tabla 1.4 se detallan las canciones enviadas al experto para evaluar.

Tabla 1.1: Canciones seleccionadas inicialmente como buenas por el experto.

Título	Música	Letra
De barro	Sebastián Piana	Homero Manzi
Sur	Aníbal Troilo	Homero Manzi
Mano a mano	Gardel/Razzano	Celedonio Flores
Cafetín de Buenos Aires	Mariano Mores	Enrique S. Discépolo
Las cuarenta	Roberto Grela	Francisco Gorrindo
Garúa	Aníbal Troilo	Enrique Cadícamo
Balada para mi muerte	Astor Piazzolla	Horacio Ferrer
Bocha	Astor Piazzolla	Horacio Ferrer
Esta ciudad	Osvaldo Avena	Héctor Negro
Sexto piso	Roberto Nievas Blanco	Homero Expósito

Se le solicitó al experto que valorara cada canción entre Muy Bueno, Bueno,

Tabla 1.2: Canciones seleccionadas inicialmente como malas por el experto.

Título	Música	Letra
La cieguita	Keppler Lais	Ramuncho
La cumparsita [versión de Matos Rodríguez]	Matos Rodríguez	Matos Rodríguez
Cucusita	Alberto Castillo	Carlos Lucero
Gurisa	Aminto Vidal	Enrique García Satur
El bazar de los juguetes	Roberto Rufino	Reinaldo Yiso
El tarta	José Rizzuti	Emilio Fresedo

Tabla 1.3: Canciones seleccionadas inicialmente como muy populares por el experto.

Título	Música	Letra
Naranjo en flor	Virgilio Expósito	Homero Expósito
Los mareados	Juan Carlos Cobián	Enrique Cadícamo
Malena	Lucio Demare	Homero Manzi
El choclo	Ángel Villoldo	Enrique Santos Discépolo y J.C. Marambio
Nostalgias	Juan Carlos Cobián	Enrique Cadícamo
Balada para un loco	Astor Piazzolla	Horacio Ferrer

Neutro, Malo y Muy Malo, además se le solicitó que indicara que tan familiarizado está con la canción en la siguiente escala: “No la conocía”, “alguna vez la escuchó”, “lo escucha en ocasiones”, “lo escucha frecuentemente”, “es un tema muy gastado”.

1.1. Resultados de la encuesta

Se evaluaron los resultados de la encuesta realizada al experto, en la Figura 1.1 se muestran los resultados para las canciones obtenidas a partir del grupo de canciones seleccionadas originalmente como buenas. En la Figura 1.2 se muestran los resultados para las canciones obtenidas a partir del grupo de canciones seleccionadas originalmente como malas. Finalmente en la Figura 1.3 se muestran los resultados para las canciones obtenidas a partir del grupo de canciones seleccionadas originalmente como muy populares.

A partir de dichos resultados se puede ver que en todos los grupos un porcentaje importante de las canciones eran desconocidas para el experto. También se puede ver que el porcentaje de canciones valoradas como buenas y muy buenas es más alto en el grupo de canciones obtenidas a partir de las valoradas originalmente como malas.

Tabla 1.4: Canciones seleccionadas automáticamente a partir de las elegidas inicialmente por el experto.

Título	Música	Letra
Lejana tierra mía	Carlos Gardel	Alfredo Le Pera
Como un sueño	Juan Carlos Cobián	Enrique Cadícamo
Matufias (O el arte de vivir)	Ángel Villoldo	Ángel Villoldo
La marcha nupcial	Venancio Clauso	Armando Tagini
Sangra el vino		
Justo el 31	Enrique Santos Discépolo	Enrique Santos Discépolo
Copen la banca	Juan Maglio	Enrique Dizeo
Barra de oro	Eduardo Moreno	Eduardo Moreno
Yo era un muchacho bueno	Juan Mercorelli	Alfredo Bigeschi
No te arrepientas	Ángel Condercuri	Abel Aznar
La cantina	Aníbal Troilo	Cátulo Castillo
Aquella cantina de la ribera	Cátulo Castillo	José González Castillo
Los tres silencios	Marcelo Saraceni	Enrique Martín
Volver a querer	Germán Teisseire	Nolo López
Pena de luna	Sebastián Piana	Federico Silva
Noches provincianas	Sebastián Piana	Homero Manzi
Sentencia	Pedro Maffia	Celedonio Flores
Milonguita (Esthercita)	Enrique Delfino	Samuel Linnig
Porteñesa a Cachorrín	Daniel Piazzolla	Horacio Ferrer
Yo	Juan José Guichandut	Juan José Guichandut
Milonga fina	José Servidio	Celedonio Flores
Mala racha	Juan Rezzano	Lito Bayardo
Melenita de oro [Pesce]	Carlos Vicente Geroni Flores	Carlos Pesce
No cantes victoria	Juan Epumer	Juan Fulginiti
Campana de plata	Carlos Vicente Geroni Flores	Samuel Linnig
Sos de la Quema	Cátulo Castillo	Edelmiro Garrido
Espiantá Gregorio	Emilio Sola	Juan Fulginiti
Arrabalera	Sebastián Piana	Cátulo Castillo
A Homero	Aníbal Troilo	Cátulo Castillo
Margot	José Ricardo / Carlos Gardel	Celedonio Flores
A vos te arrancaron verde	Carmelo Taverna	Francisco Lío
Balada para un loco	Astor Piazzolla	Horacio Ferrer
De mi tierra [b]	Eduardo Manella	Francisco Lozano

1.2. Conclusión

A partir de la encuesta realizada se puede ver que hay un mayor porcentaje de canciones valoradas como buenas en el conjunto generado a partir de las canciones seleccionadas originalmente como malas. Por lo tanto, no pode-

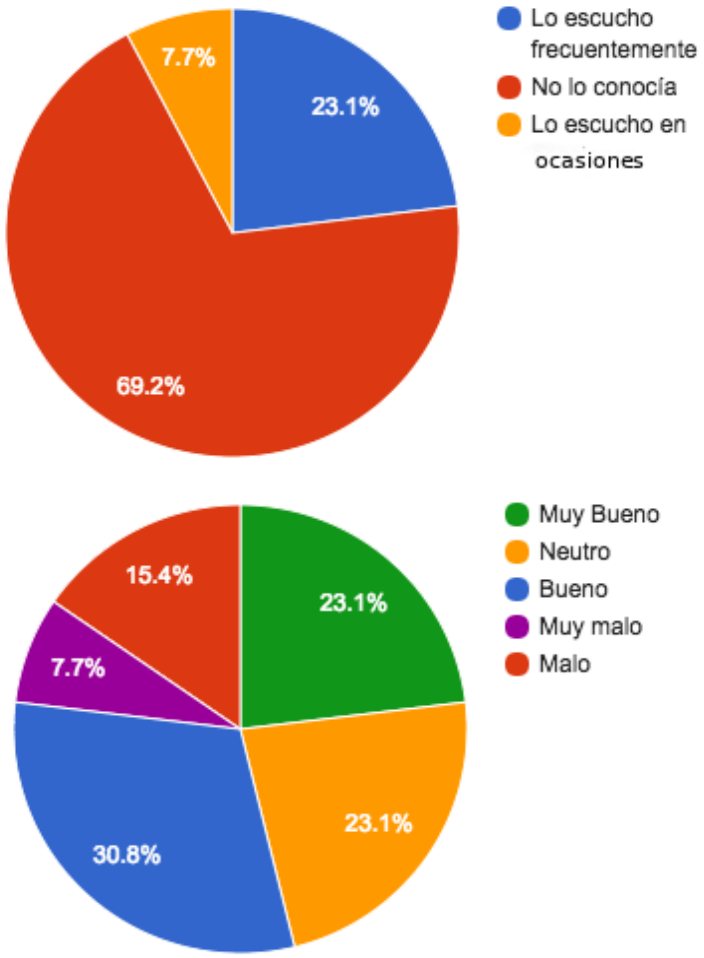

Figura 1.1: Resultados de encuesta a partir de grupo de canciones valoradas como buenas.

mos afirmar que la representación mantiene la relación sobre la calidad de las canciones.

Además, es claro que esta encuesta es muy subjetiva y como evaluación no aporta mucho valor ya que solamente se consulta a una persona. De todas formas, en una primera fase exploratoria fue útil para aproximarse al tema y plantear distintas alternativas.

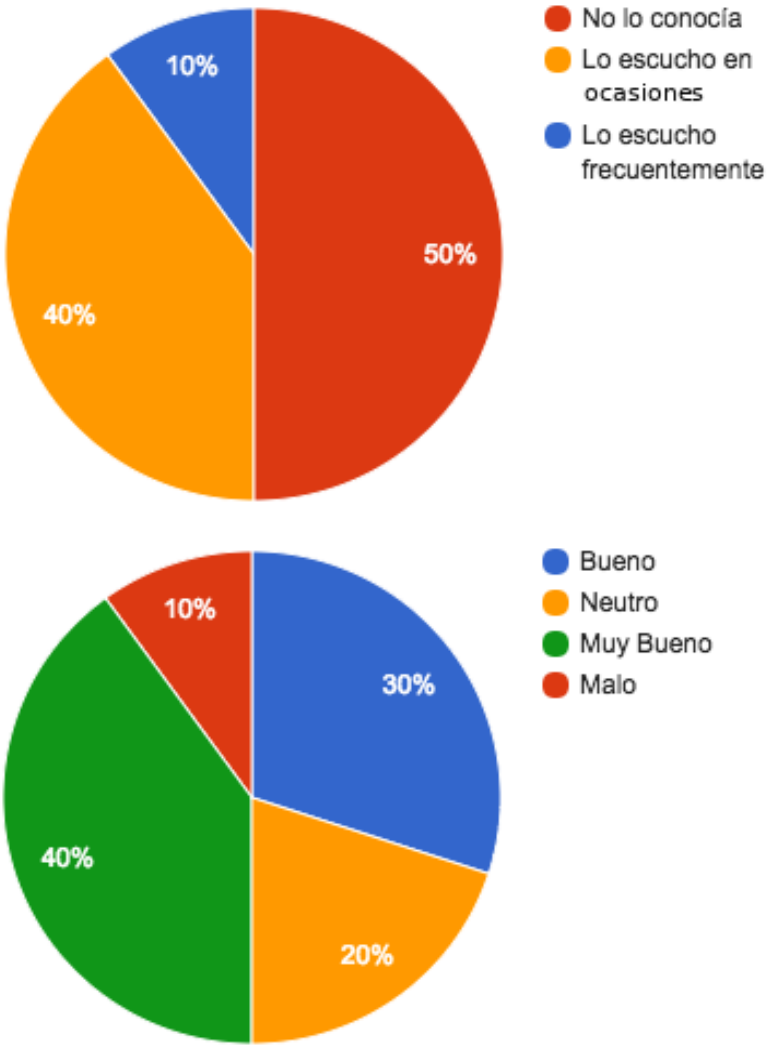

Figura 1.2: Resultados de encuesta a partir de grupo de canciones valoradas como malas.

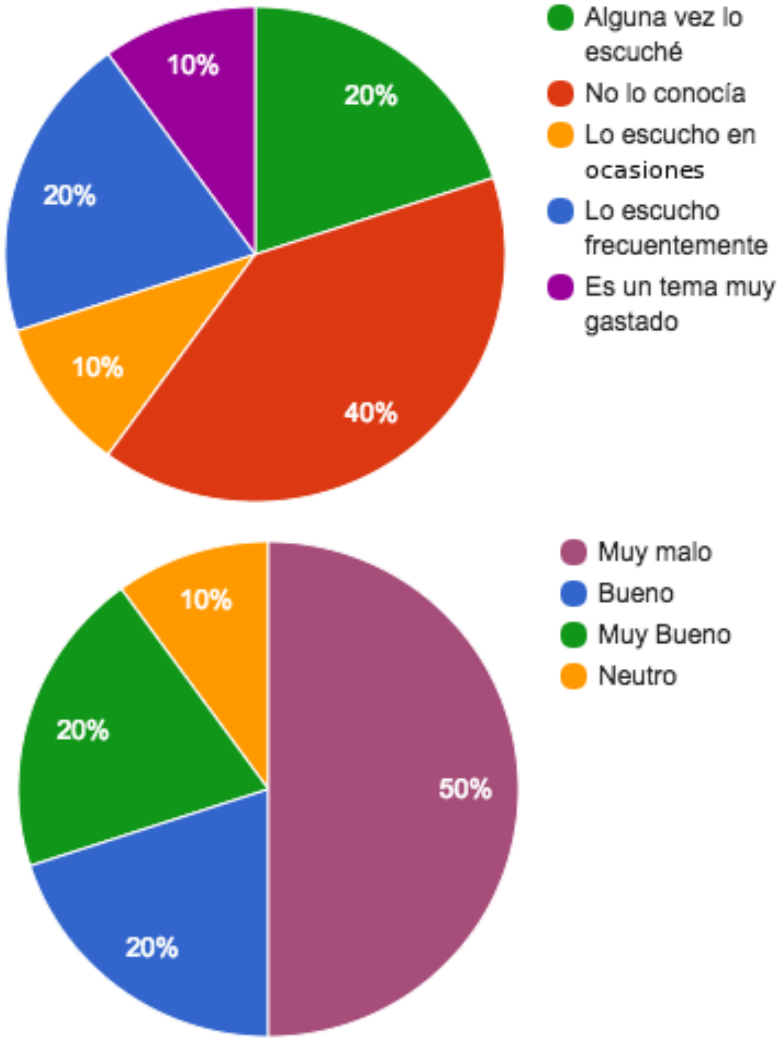

Figura 1.3: Resultados de encuesta a partir de grupo de canciones valoradas como muy populares.