Universitat Pompeu Fabra
Barcelona

Knowledge-based and Multimodal Deep Learning Approaches for Music Recommendation and Classification

Sergio Oramas Martín

TESI DOCTORAL UPF / 2015

Director de la tesi:

Dr. Xavier Serra Casals
Dept. of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona, Spain

Dissertation submitted to the Deptartment of Information and Communication Technologies of Universitat Pompeu Fabra in partial fulfillment of the requirements for the degree of

DOCTOR PER LA UNIVERSITAT POMPEU FABRA

Music Technology Group (http://mtg.upf.edu), Dept. of Information and Communication Technologies (http://www.upf.edu/dtic), Universitat Pompeu Fabra (http://www.upf.edu), Barcelona, Spain.

Acknowledgements

I can say now, after four and a half years working on this thesis, that so far these have been the greatest and most challenging years of my life. I learned a lot of things during these years, and surely not only about *tags*. There are many people I want to thank for having contributed, in one way or another, to make this happen. First of all, I would like to thank Xavier Serra, not only for giving me the opportunity to join the MTG and supervising this thesis, but also for having lead the MTG for more than 20 years and always being enthusiastic about new projects and ideas. I remember, when I was about to apply for the grant that then supported my research, Xavier told me that pursuing a PhD is a "way of life". He was right, and has helped me since then. However, there is someone else to whom I feel much obliged for helping me out. After I did my first presentation on what my thesis was going to be about, Joan Serrà told me that he was interested and, if I wanted, he could help me. I was not aware at the time of how important that collaboration would become, nor about how much I would learn from him. Fortunately I knew enough to say *yes*, and so Joan became co-supervisor of the thesis. Since then, his help at all stages has been invaluable.

A very important element of this thesis has been Freesound. Working with Freesound has been a great motivation throughout the thesis, and it has turned me into a developer that now reads programming books and Python blogs. All because I have been lucky enough to be part of the Freesound team, and to work with people like Gerard Roma, Alastair Porter, Bram de Jong, and former Freesound team members Vincent Akkermans, Stelios Togias and Jordi Funollet. To all of you, I sincerely thank you for teaching me so many *geeky* things, research and programming philosophy. Also related to Freesound, I want to particularly thank the Freesound moderators and all Freesound users that participated in my online experiments and that contribute everyday to make Freesound such an amazing site.

There is many other people at the MTG without whom this journey would not have been half as enjoyable. My lunch mates from the "Lunch time conversations (official thread)" Skype group: Panos Papiotis, Sergio Oramas, Sebastian Mealla, Oriol Romaní, Giuseppe Bandeira, Dara Dabiri, Álvaro Sarasúa, Juanjo Bosch, Martí Umbert, Carles F. Julià, and everyone else with whom I shared thoughts on the thesis or about any other professional or personal aspects: Sankalp Gulati, Sertan Şentürk, Mohamed Sordo, Gopala Krishna Koduri, Dmitry Bogdanov, Rafael Caro, Agustín Martorell , Nadine Kroher, Ajay Srinivasamurthy, Justin Salamon, and those that I'm missing! From the MTG, I would also particularly like to thank Perfecto Herrera for his input when designing user experiments, and Alba Rosado for making me feel I could do much more than research at the MTG. Furthermore, I want to thank György Fazekas for his help and collaboration during my stay at Centre for Digital Music, Queen Mary University of London, and Tamsin Porter for proofreading this thesis.

Last but not least, I want to thank all my friends and family and, with a particular emphasis, I would like to thank Anna for always being there, accompanying me throughout the whole process, and letting me accompany her on her own. Thank you!

This thesis has been carried out at the Music Technology Group of Universitat Pompeu Fabra (UPF) in Barcelona, Spain, from October 2010 to February 2014 and from May 2014 to March 2015, and at the Centre for Digital Music of Queen Mary University of London (QMUL), United Kingdom, from March 2014 to April 2014. This work has been supported by the Spanish Ministry of Science and Innovation (BES-2010-037309 FPI grant and TIN-2009-14247-C02-01 DRIMS project), and by the European Research Council (FP7-2007-2013 / ERC grant agreement 267583). The research stay at QMUL has been also funded by the Spanish Ministry of Science and Innovation (EEBB-I-14-08838).

Abstract

Online sharing platforms host a vast amount of multimedia content generated by their own users. Such content is typically not uniformly annotated and can not be straightforwardly indexed. Therefore, making it accessible to other users poses a real challenge which is not specific of online sharing platforms. In general, content annotation is a common problem in all kinds of information systems. In this thesis, we focus on this problem and propose methods for helping users to annotate the resources they create in a more comprehensive and uniform way. Specifically, we work with tagging systems and propose methods for recommending tags to the content creators during the annotation process. To this end, we exploit information gathered from previous resource annotations in the same sharing platform, the so called *folksonomy*. Tag recommendation is evaluated using several methodologies, with and without the intervention of users, and in the context of large-scale tagging systems. We focus on the case of tag recommendation for sound sharing platforms. Besides studying the performance of several methods in this scenario, we analyse the impact of one of our proposed methods on the tagging system of a real-world and large-scale sound sharing site. As an outcome of this thesis, one of the proposed tag recommendation methods is now being daily used by hundreds of users in this sound sharing site. In addition, we explore a new perspective for tag recommendation which, besides taking advantage of information from the folksonomy, employs a sound-specific ontology to guide users during the annotation process. Overall, this thesis contributes to the advancement of the state of the art in tagging systems and folksonomy-based tag recommendation, and explores interesting directions for future research. Even though our research is motivated by the particular challenges of sound sharing platforms and mainly carried out in that context, we believe our methodologies can be easily generalised and thus be of use to other information sharing platforms.

Resum

Les plataformes d'intercanvi de recursos multimèdia a Internet contenen grans quantitats de contingut creat pels seus usuaris. Habitualment, aquest contingut no està ben anotat, i això fa que la seva indexació no sigui una tasca fàcil. Aconseguir que aquest contingut sigui accessible pels altres usuaris suposa un repte important, el qual no és només específic d'aquest tipus de plataformes. En general, l'anotació de contingut és un problema comú en molts tipus de sistemes d'informació. En aquesta tesi, ens focalitzem en aquest problema i proposem mètodes per ajudar els usuaris a anotar, d'una manera més completa i uniforme, el contingut creat per ells mateixos. Concretament, treballem amb sistemes d'etiquetatge – *tagging* – i proposem mètodes per recomanar etiquetes – *tags* – durant el procés d'anotació del contingut. Per aconseguir això, analitzem la manera com els altres continguts de la plataforma d'intercanvi han estat etiquetats prèviament. Aquesta informació s'anomena *folksonomia*. Avaluem la tasca de recomanar tags utilitzant diverses metodologies, amb o sense la participació d'usuaris, i en el context de sistemes de tagging a gran escala. Particularment, ens focalitzem en el cas de la recomanació de tags en plataformes d'intercanvi de sons i, a part de testar el funcionament de diferents mètodes en aquest escenari, també analitzem l'impacte d'un d'aquests mètodes en el sistema de tagging d'una plataforma d'intercanvi de sons real. De fet, de resultes d'aquesta tesi, centenars d'usuaris fan servir diàriament un dels sistemes proposats de recomanació de tags en aquesta plataforma d'intercanvi. A més a més, també explorem un nou enfocament per als sistemes de recomanació de tags que, a part de nodrir-se de la informació de la folksonomia, incorpora una ontologia amb informació sobre l'àmbit del so que serveix per guiar els usuaris durant el procés d'anotació de contingut. En general, aquesta tesi contribueix a l'avenç de l'estat de l'art dels sistemes de tagging i de recomanació de tags basats en folksonomies, i explora direccions interessants per continuar investigant. Tot i que la nostra recerca està motivada pels reptes particulars que proposen les plataformes d'intercanvi de sons i està avaluada principalment en aquest context, creiem que les metodologies que proposem poden ser generalitzades fàcilment i utilitzades en altres plataformes d'intercanvi.

Contents

List of figures

List of tables

List of mathematical symbols

General

Example	Symbol type	Description
a, b, γ	Lowercase letters	Indices, variables, vector, set and matrix elements.
A, B, Γ	Uppercase letters	Constants, functions and evaluation metrics.
$\mathbf{A}, \mathbf{B}, \mathbf{C}$	Bold uppercase letters	Vectors and sets.
$\mathcal{A}, \mathcal{B}, \mathcal{C}$	Calligraphy letters	Graphs, matrices and other complex data structures.

Specific

Symbol	Description
$\mathbf{A}$	Set of annotation sessions
a	Element of $\mathbf{A}$ (a particular annotation session)
$\mathbf{C}$	Set of audio classes
$\mathcal{D}$	Association matrix
d	Element of $\mathcal{D}$
$\mathbf{E}$	Vector of tag applications (edges of the folksonomy hypergraph)
F	F-measure evaluation metric
$\mathcal{F}$	Folksonomy
I	(In)coherence in annotations evaluation metric
M	Percentage of misspelled tag applications evaluation metric
$\mathcal{O}$	Ontology
P	Precision evaluation metric
p	p-value in statistical tests
Q	Subjective annotation quality evaluation metric
$\mathbf{Q}$	Union of all qualitative judgements of sound annotations
q	Qualitative judgement for the annotation of a sound
$\mathcal{R}$	Sound-sound graph
R	Recall evaluation metric
$\mathbf{R}$	Set of resources (typically of sounds)
r	A particular resource (typically a sound)
$\mathcal{S}$	Tag-tag similarity matrix

Symbol	**Description**
s	Element of $\mathcal{S}$
$\mathbf{T}$	Set of tags
$\mathbf{T}_{\mathrm{A}}$	Set of aggregated candidate tags
$\mathbf{T}_{\mathrm{C}}$	Set of candidate tags
$\mathbf{T}_{\mathrm{D}}$	Set of deleted tags
$\mathbf{T}_{\mathrm{I}}$	Set of input tags
$\mathbf{T}_{\mathrm{R}}$	Set of recommended tags
$\mathbf{T}^{r}$	Set of tags assigned to a resource r (tagline of the resource)
$\mathbf{T}_{\mathrm{T}}$	Set of attribute-tags
$\mathbf{T}_{\mathrm{Z}}$	Set of tags populated under a tag category
t	A particular tag
$\mathcal{TR}$	Bipartite graph relating tags and resources
$\mathcal{U}$	User-user graph
$\mathbf{U}$	Set of users
u	A particular user
W_I	Analysis window of interest
$\mathbf{W}$	Vector of reference analysis windows
w	Edge weight for user-user and sound-sound graphs
$\mathbf{Z}$	Set of tag categories
$\mathbf{Z}_{\mathrm{R}}$	Set of recommended tag categories
z	Element of $\mathbf{Z}$ (a particular tag category)
α	Percentage parameter of Percentage Strategy
β	Percentage parameter of Kernel Percentage Strategy
Γ	Average tagline length evaluation metric
ε	Score threshold for candidate tags
Θ	Average percentage of new tags evaluation metric
θ	Number of candidate tags per input tag
κ	Fixed number of recommended tags
Λ	Annotation comprehensiveness evaluation metric
λ	Duration of an annotation session
Ψ_{u}	User vocabulary sharing evaluation metric
Ψ_{r}	Sound vocabulary sharing evaluation metric
Φ_{e}	Average tag application time evaluation metric
Φ_{r}	Average time per sound evaluation metric
ϱ	Number of repeated tags in Repeated aggregation and selection strategy
Υ	Average user vocabulary size evaluation metric
υ	Tag frequency of occurrence
ϕ	Score of a candidate tag
Π	Average percentage of attribute-tags evaluation metric
Ω	Average number of correctly predicted tags evaluation metric
ω	Tag frequency threhsold

CHAPTER 1

Introduction

1.1 Motivation

CHAPTER 2

Literature review

2.1 Introduction

The literature review presented in this chapter is divided into five parts. Firstly, we summarise existing work on information extraction, with special focus on its application to the music domain. We describe what Entity Linking is, the different state-of-the-art systems. Additionally, we breafly descrive the different approaches for relation extraction. discuss problems that are typically found in tagging systems and highlight some of the solutions that are commonly proposed. Secondly, we define what a Knowledge Base (KB) is and the different types of KBs. Moreover, we deepen into the available KBs with music related information. Thirdly, we focus on existing literature about music genre classification, with a special focus in text-based approaches. Finally, we outline the different apporaches that have been proposed about semantic-based recommender systems and cold-start music recommendation.

2.2 Information Extraction

Information Extraction (IE) can be defined as the task of automatically extracting structured information from unstructured or semi-structured text sources. It is a widely studied technique within the Natural Language Processing (NLP) research community, whose major challenge is to understand natural language. A major step towards understanding language is the extraction of meaningful terms (entities) from text as well as relationships between those entities. This statement involves two different tasks. The former is to determine the identity and category of entity mentions present in text. This task is often called Named Entity Recognition (NER). However, when this task involves a latter step of disambiguation of entities against a KB it is often called Named Entity Disambiguation (NED) or Entity Linking (EL). The second task is to identify and annotate relevant semantic relations between entities in text. This task is called Rellation Extraction, and is an established task in NLP.

2.2.1 Entity Linking

The advent of large knowledge repositories and collaborative resources has contributed to the emergence of Entity Linking (EL), i.e. the task of discoveing mentions of entities in text and link them to a suitable knowledge repository (Moro et al., 2014c). It encompasses similar subtasks such as Named Entity Disambiguation (Bunescu & Pasca, 2006), which is precisely linking mentions to entities to a KB, or Wikification (Mihalcea & Csomai, 2007), specifically using Wikipedia as KB. There have been a great development of EL systems that perform well in general purpose domains. Among these systems we focus on three of them in this thesis:

DBpedia Spotlight (Mendes et al., 2011) is a system for automatically annotating text documents with DBpedia URIs, finding and disambiguating natural language mentions of DBpedia resources. DBpedia Spotlight is shared as open source and deployed as a Web service freely available for public use[1]. DBpedia Spotlight gives as a result the DBpedia uri, start and end char positions, the value of the rdf:type property, and a confidence score.

TagMe (Ferragina & Scaiella, 2012) is an EL system that matches terms with Wikipedia link texts and disambiguates them using the in-link graph and the page dataset. Then, it performs a pruning process by looking at the entity context. TagMe is available as a web service [2]. Tagme output provides the start and end char position, the Wikipedia page id, the Wikipedia categories and a confidence score.

Babelfy (Moro et al., 2014a) is an EL and WSD based on non-strict identification of candidate meanings (i.e. not necessarily exact string matching), together with a graph based algorithm that traverses the BabelNet graph and selects the most appropriate semantic interpretation for each candidate. Babelfy output provides the BabelNet synset and the word index. If the synset references to a Wikipedia page, it returns the Wikipedia url, the DBpedia uri and the Wikipedia categories. If it points to WordNet, it yields the name of the equivalent WordNet synset.

In the context of Open Data, the need for benchmarking datasets and evaluation frameworks for EL is clear. However, while general purpose datasets exist (Usbeck et al., 2015), dealing with highly specific domains (e.g. chemistry) or ever-evolving areas (e.g. videogames or music) poses a greater challenge due to linguistic idiosincrasies or under-representation in general purpose knowledge-bases. This is true in the music domain as well, where available data is scarce (Gruhl et al., 2009).

Among the few works on EL for the music domain, let us refer to (Gruhl et al.,

[1] lhttps://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki/Web-service
[2] lhttp://tagme.di.unipi.it/

2009), who describe an approach for detecting musical entities in informal text. In addition, (Zhang et al., 2009) describe a system for musical EL in the Chinese language based on Hidden Markov Models. Finally, (Oramas et al., 2015b) describe an EL system for recognizing musical entities in a relation extraction pipeline.

Regarding unification frameworks for EL, (Cornolti et al.) put forward a benchmarking framework for comparing EL systems, leveraging Wikipedia, and a hierarchy of EL problems together with a set of novel measures. (Rizzo et al., 2014) divide the process into NER and EL, evaluating each stage individually. They subsequently describe a system aimed at combining the output of the different NER systems. Finally (Usbeck et al., 2015) present GERBIL, an evaluation framework for semantic EL based on (Cornolti et al.). It is an open-source and extensible framework that allows evaluating tools against different datasets.

2.2.2 Relation Extraction

A large portion of the knowledge contained in the web is stored in unstructured natural language text. In order to acquire and formalize this heterogeneous knowledge, methods that automatically process this information are in demand. Extracting semantic relations between entities is an important step towards this formalization (Wang, 2008). Relation Extraction is an established task in Natural Language Processing (Bach & Badaskar, 2007). It has been defined as the process of identifying and annotating relevant semantic relations between entities in text (Jiang & Zhai, 2007).

Relation Extraction (RE) approaches are often classified according to the level of supervision involved. Supervised learning is a core-component of a vast number of RE systems, as they offer high precision and recall. However, the need of hand labeled training sets makes these methods not scalable to the thousands of relations found on the Web (Hoffmann et al., 2011). More promising approaches, called semi-supervised approaches, bootstrapping approaches, or distant supervision approaches do not need big hand labeled corpus, and often rely on existent knowledge base to heuristically label a text corpus (e.g., (Carlson et al., 2010; Hoffmann et al., 2011)) Open Information Extraction methods do not require a pre-specified vocabulary, as they aim to discover all possible relations in the text (Banko et al., 2007b). However, these methods have to deal with uninformative and incoherent extractions. In ReVerb (Fader et al., 2011) part-of-speech based regular expressions are introduced to reduce the number of these incoherent extractions. Less restrictive pattern templates based on dependency paths are learned in OLLIE (Mausam et al., 2012) to increase the number of possible extracted relations. Unsupervised approaches do not need any annotated corpus. In (Eichler et al., 2008) verb relations involving a subject and an object are extracted, using simplified dependency trees in sentences with at least two named entities. These approaches can process very large amounts of data, however, the resulting relations are hard to map to ontologies (Isabelle Augenstein & Ciravegna, 2014).

In our work we use an NLP task called Dependency Parsing in the relation extraction process. Dependency Parsing provides a tree-like syntactic structure of a sentence based on the linguistic theory of Dependency Grammar (Tesnière, 1959). One of the outstanding features of Dependency Grammar is that it represents binary relations between words (Ballesteros & Nivre, 2013), where there is a unique edge joining a node and its parent node (see Fig. 4.1 for the full parsing of an example sentence). Dependency relations have been successfully incorporated to RE systems. For example, (Bunescu & Mooney, 2005) describe and evaluate a RE system based on shortest paths among named entities. (Culotta & Sorensen, 2004) focus on the smallest dependency subtree in the sentence that captures the entities involved in a relation, and (Gamallo et al., 2012) propose a rule-based dependency-parsing Open IE system. Moreover, in (Nakashole et al., 2012; Moro & Navigli, 2012; Bovi et al., 2015b) syntactyc and semantic information is combined by means of the combination of Dependency Parsing and Entity Linking techniques.

2.3 Knowledge Bases

The work described in this thesis strongly focuses on the exploitation of linguistic and semantic properties of text collections for the automatic learning of Music Knowledge Bases MKBs. For this reason, we deem relevant to cover related work in the following areas: (1) KB learning and curation, with special focus on RE methods; and (2) KB learning and its applications in the music domain as well as Music Information Retrieval (MIR).

2.3.1 Knowledge Base Construction

We understand language by making sense of the connections between words, concepts, phrases and thoughts (Havasi et al., 2007). KBs constitute a resource for encapsulating this knowledge. Previous efforts on KB construction may be characterized as: (1) Hand-crafted KBs; (2) Integrative projects (automatic in design, but reliant on manually validated data); and (3) Fully automatic, also in the RE process.

Among the first group, the best known is probably WordNet (Miller, 1995), a lexical database which groups concepts in "synonym sets", and encodes predefined relations among them such as *hyponymy/hypernymy*, *meronymy*, *holonymy*, or *instantiation*. Manually constructed KBs, however, are mostly developed in specific domains, where the degree of ambiguity is lower and there is more availability of trained knowledge engineers.

Next, integrative projects are probably the most productive, as they are the most ambitious attempts in terms of content coverage and community involvement, not only users, but also contributors. Examples of these include YAGO (Suchanek et al., 2007), an automatically created KB derived from integrating WIKIPEDIA and WORDNET; DBPEDIA (Lehmann et al., 2014), a collaboratively maintained project aimed at exploiting information present in WIKIPEDIA, both structured and in free text; FREEBASE (Bollacker et al., 2008), also a collaborative effort mainly based on extracting structured knowledge from WIKIPEDIA; or BABELNET (Navigli & Ponzetto, 2012), a semantic network which started as a seamless integration of WIKIPEDIA and WordNet, and today constitutes the largest multilingual repository of words and senses.

With regard to the third group we refer to approaches where knowledge is obtained automatically. Usually, these are framed within the *Open Information Extraction* (OIE) paradigm (Banko et al., 2007a), which can be (roughly) summarized as (1) reading the web, (2) learning facts, (3) scoring them; and (4) structuring them according to predefined semantic criteria. Endeavours in this area include TEXTRUNNER (Banko et al., 2007a), widely regarded as the first OIE system; REVERB (Fader et al., 2011), particularly designed to reduce noise while keeping a wide coverage, thanks in part to a set of syntactic and lexical constraints; NELL (Carlson et al., 2010), which incorporates semantic knowledge in the form of a hand-crafted taxonomy of entities and relations; PATTY (Nakashole et al., 2012) and WISENET (Moro & Navigli, 2012, 2013), in which a shared vision to integrate semantics is applied both at the entity and relation level; DEFIE (Bovi et al., 2015b), a recent development in OIE tested on the whole set of BABELNET glosses; and KB-UNIFY (Bovi et al., 2015a), not an actual OIE implementation, but rather a unification framework for OIE systems.

2.3.2 Music Knowledge Bases

MUSICBRAINZ and DISCOGS are two paramount examples of manually curated MKBS. They are open music encyclopedias of music metadata built collaboratively and openly available. MUSICBRAINZ, in addition, is regularly published as Linked Data by the LINKEDBRAINZ project[3].

As for generic KBS based on WIKIPEDIA, such as the ones described earlier, these include a remarkable amount of music data, such as artist, album and song biographies, definitions of musical concepts and genres, or articles about music institutions and venues. However, their coverage is biased towards the best known artists, and towards products from Western culture. Finally, let us refer to the notable case of GROVE MUSIC ONLINE[4], a music encyclopedia

[3] http://linkedbrainz.org/
[4] http://www.oxfordmusiconline.com

containing over 60k articles written by music scholars. However, it has the drawback of not being freely open, as it runs by subscription.

Other than the aforementioned curated repositories, to the best of our knowledge, there is not a single automatically learned open MKB. A first step in this direction was taken in (Sordo et al., 2015; Oramas et al., 2014), applying RE techniques to big datasets of music related texts extracted from the web. Moreover, in (Oramas et al., 2015a), a Flamenco MKB is created by combining data from curated KBs and information extracted from blogs and websites.

Despite their scarcity, MKBs are becoming increasingly popular in MIR applications, such as artist similarity and music recommendation (Celma & Serra, 2008; Oramas et al., 2015c; Leal et al., 2012; Ostuni et al., 2015). MKBs have also been exploited as sources of explanations in music recommender systems. According to (Celma & Herrera, 2008), giving explanations of the recommendations provides transparency to the recommendation process and increases the confidence of the user in the system. In (Passant, 2010), explanations of recommendations are created by exploiting DBpedia's structured information, whilst in (Sordo et al., 2015), explanations are based on an automatically learned MKB.

2.4 Music Genre Classification

Music genre labels are useful categories to organize and classify songs, albums and artists into broader groups that share similar musical characteristics. They have been widely used for music classification, from physical music stores to streaming services. Music genre classification thus is a widely explored topic within the MIR community (Sturm, 2012).

Most published music genre classification approaches rely on audio sources (Sturm, 2012; Bogdanov et al., 2016). Traditional techniques typically use handcrafted audio features, such as Mel Frequency Cepstral Coefficients (MFCCs) (Logan & Others, 2000), as input of a machine learning classifier (e.g., SVM, k-NN) (Tzanetakis & Cook, 2002; Seyerlehner et al., 2010). More recent deep learning approaches take advantage of visual representations of the audio signal in form of spectrograms. These visual representations of audio are used as input to Convolutional Neural Networks (CNNs) (Dieleman et al., 2011; Dieleman & Schrauwen, 2014; Pons et al., 2016; Choi et al., 2016a,b), following approaches similar to those used for image classification.

Text-based approaches have also been explored for this task. For instance, one of the earliest attempts on genre classification of music reviews is described in (Hu et al., 2005), where experiments on multiclass genre classification and star rating prediction are described. Similarly, (Hu & Downie, 2006) extend these experiments with a novel approach for predicting usages of music via agglomerative clustering, and conclude that bigram features are more informative than unigram features. Moreover, part-of-speech (POS) tags along pattern mining techniques are applied in (Downie & Hu, 2006) to extract descriptive patterns for distinguishing negative from positive reviews. Additional textual evidence is leveraged in (Choi et al., 2014), who consider lyrics as well as texts referring to the meaning of the song, and used for training a kNN classifier for predicting song subjects (e.g. war, sex or drugs).

By contrast, there are a limited number of papers dealing with image-based genre classification (Libeks & Turnbull, 2011). Regariding multimodal approaches found in the literature, most of them combine audio and song lyrics as text (Laurier et al., 2008; Neumayer & Rauber, 2007). Moreover, other modalities such as audio and video have been explored (Schindler & Rauber, 2015).

Almost all related work about Music Genre Classification is concentrated in multi-class classification of music items into broad genres (e.g., Pop, Rock), assigning a single label per item. This is problematic since there may be hundreds of more specific music genres (Pachet & Cazaly, 2000), and these may not be necessarily mutually exclusive (i.e., a song could be Pop, and at the same time have elements from Deep House and a Reggae grove). Multi-label classification is a widely studied problem (Tsoumakas & Katakis, 2006; Jain et al., 2016). Although there are not many approaches for multi-label classification of music genres (Sanden & Zhang, 2011; Wang et al., 2009), there is a long tradition in MIR for tag classification, which is a highly related multi-label problem (Choi et al., 2016a; Wang et al., 2009).

2.5 Artist Similarity

Music artist similarity has been studied from the score level, the acoustic level, and the cultural level (Ellis et al., 2002). In this disertation, we focus on the latter approach, and more specifically in text-based approaches. Literature on document similarity, and more specifically on the application of text-based approaches for artist similarity is discussed next.

The task of identifying similar text instances, either at sentence or document level, has applications in many areas of Artificial Intelligence and Natural Language Processing (Liu & Wang, 2014). In general, document similarity can be computed according to the following approaches: surface-level representation like keywords or n-grams (Chim & Deng, 2008); corpus representation using counts (Rorvig, 1999), e.g. word-level correlation, jaccard or cosine models; Latent factor models, such as Latent Semantic Analysis (Deerwester et al., 1990); or methods exploiting external knowledge bases like ontologies or encyclopedias (Hu et al., 2009).

The use of text-based approaches for artist and music similarity was first applied in (Cohen & Fan, 2000), by computing co-occurrences of artist names in web page texts and building term vector representations. By contrast, in (Schedl et al., 2005) term weights are extracted from search engine's result counts. In (Whitman & Lawrence, 2002) n-grams, part-of-speech tagging and noun phrases are used to build a term profile for artists, weighted by employing tf-idf. Term profiles are then compared and the sum of common terms weights gives the similarity measure. More approaches using term weight vectors have been developed over different text sources, such as music reviews (Hu et al., 2005), blog posts (Celma et al., 2006), or microblogs (Schedl et al., 2013). In (Logan & Ellis, 2003) Latent Semantic Analysis is used to measure artist similarity from song lyrics. Domain specific ontologies have also been applied to the problem of music recommendation and similarity, such as in (Celma & Serra, 2008). In (Leal et al., 2012), paths on an ontological graph extracted from DBpedia are exploited for recommending music web pages. However, to the best of our knowledge, there are scant approaches in the music domain that exploit implicit semantics and enhance term profiles with external knowledge bases.

2.6 Recommender Systems

Recommender systems can be broadly classified into collaborative filtering (CF), content-based, and hybrid methods. Collaborative filtering methods (Koren et al., 2009) use the item-user feedback matrix and predictions are based on the similarity of user or items profiles. Matrix factorization techniques are currently CF state-of-the-art (Koren et al., 2009). CF methods suffer from the cold-start problem, as new items do not have feedback information (Saveski & Mantrach, 2014). Content-based methods (Mooney & Roy, 1999) rely only on item features, and recommendations are based on similarity between such features. Finally, hybrid methods (Burke, 2002) try to combine both item content and item-user feedback.

2.6.1 Semantic-based Approaches

Ontology-based and semantics-aware recommendation systems have been proposed in many works in the past. In (Middleton et al., 2009) an ontological recommender system is presented that makes use of semantic user profiles to compute collaborative recommendations with the effect of mitigating cold-start and improving overall recommendation accuracy. In (Mobasher et al., 2004) the authors present a *semantically enhanced collaborative filtering* approach, where structured semantic knowledge about items is used in conjunction with user-item ratings to create a combined similarity measure for item comparisons. In (Ziegler et al., 2004) taxonomic information is used to represents the user's interest in categories of products. Consequently, user similarity is determined by common interests in categories and not by common interests in items. In (Anand et al., 2007) the authors present an approach that infers user preferences from rating data using an item ontology. The system collaboratively generates recommendations using the ontology and infers preferences during similarity computation. Another hybrid ontological recommendation system is proposed in (Cantador et al., 2008) where user preferences and item features are described by semantic concepts to obtain users' clusters corresponding to implicit *Communities of Interest*. In all of these works, the experiments prove an accuracy improvement over traditional memory-based collaborative approaches especially in presence of sparse datasets. In the last few years with the availability of Linked Open Data (LOD) datasets, a new class of recommender systems has emerged which can be named as LOD-based recommender systems. One of the first approaches that exploits Linked Open Data for building recommender systems is (Heitmann & Hayes, 2010). In (Fernández-Tob\'\ias et al., 2011) the authors present a knowledge-based framework leveraging DBpedia for computing cross-domain recommendations. In (Di Noia et al., 2012a,b) a model-based approach and a memory-based one to compute content-based recommendations are presented leveraging LOD datasets. Another LOD content-based method is presented in (Ostuni et al., 2014) which defines a neighborhood-based graph kernel for matching graph-based item representations. Two hybrid approaches have been presented lately. In (Ostuni et al., 2013) the authors show how to compute top-N recommendations from implicit feedback using linked data sources and in (Khrouf & Troncy, 2013) the authors propose an event recommendation system based on linked data and user diversity. In (Rowe, 2014) the authors propose a semantic-aware extension of the SVD++ model, named SemanticSVD++, which incorporates semantic categories of items into the model. The model is able also to consider the evolution over time of user's preferences. Finally, another interesting direction about the usage of LOD for content-based RSs is explored in (Musto et al., 2014) where the authors present Contextual eVSM, a content-based context-aware recommendation framework that adopts a semantic representation based on distributional models and entity linking techniques. In particular entity linking is used to detect entities in free text and map them to LOD.

2.6.2 Music Recommendation

An overview about techiques for music recommendation and similarity based on music contextual data is given in (Knees & Schedl, 2013). In (Kaminskas & Ricci, 2012) the authors provide a description of various tools and techniques that can be used for addressing the research challenges posed by context-aware music retrieval and recommendation. A survey about techniques for the generation of music playlists is given in (Bonnin & Jannach, 2014). In particular, the authors provide a review of the literature on automated playlist generation and a categorization of the existing approaches. A context-aware music recommender system which infers contextual information based on the most recent sequence of songs liked by the user is presented in (Hariri et al., 2012). More recently, a playlist generation algorithm with the goal of maximizing coherence and personalization of the playlist has been presented in (Jannach et al., 2015). Finally, in (Aghdam et al., 2015) a technique for adapting recommendations to contextual changes based on hierarchical hidden Markov models is presented.

Social tags have been extensively used as a source of artist content features to recommend music (Knees & Schedl, 2013). However, these tags are usually collectively annotated, which often introduce an artist popularity bias (Turnbull et al., 2008). Artist biographies and press releases, on the other hand, do not necessarily require a collaborative effort, as they may be produced by artists themselves. However, they have seldom been exploited for music recommendation (Oramas et al., 2015b). Part of the work on this disertation focuses on the exploitation of these biographies.

Furthermore, we also make use of audio signals, since these are generally always available and have shown to be helpful when recommending music in the long tail (van den Oord et al., 2013).

In both cases, a hybrid recommendation approach is used based on learning attribute-to-feature mappings (Gantner et al., 2010). This method addresses the lack of feedback for uncommon items in two steps: (1) factorizing the collaborative matrix, and (2) learning a mapping between item content features and item latent factors (van den Oord et al., 2013; Bansal et al., 2016).

Part I

Knowledge-based Approaches

CHAPTER 3

Linking Texts to Music Knowledge Bases

3.1 Introduction

When we refer to the Music Domain in a Natural Language Processing (NLP) context we refer to Music product reviews such as Albums or Songs, Music-related biographies or even song lyrics. While these are valuable resources in NLP for tasks like Sentiment Analysis, Music Information Retrieval (MIR), however, has barely exploited the information and knowledge that can be extracted from textual data. This opens up a vibrant area of research where MIR tasks may benefit dramatically from mining textual data Sordo et al. (2015).

Named Entity Recognition (NER) is the task to identify mentions to entities belonging to a set of predefined categories Zhou & Su (2002). Traditionally, the most widely covered types of entities are Person, Location and Organization, as well as numeric expressions or time-spans. While NER is a widely studied topic, and has been at the core of well-known shared tasks and conferences Nadeau & Sekine (2007) such as MUC, ACE or CoNLL, the advent of large knowledge repositories and collaborative resources has contributed to the emergence of another discipline: Entity Linking (EL), i.e. to discover mentions of entities in text and link them to a suitable knowledge repository Moro et al. (2014c).

to obtain an annotation for Music entity mentions in text, either simply as Music types (e.g. tagging 'Yellow Submarine' as `Song`) or performing Entity Linking (EL), e.g. tagging 'Yellow Submarine' as `dbpedia.org/page/Yellow_Submarine_(song)`. However, this is not a trivial task as mentions to Music entities show language and register idiosyncrasies Tata & Di Eugenio (2010); Gruhl et al. (2009), and therefore a certain degree of tailoring is required in order to account for them. Let us consider multiword Music entities, which usually are those who pose greatest challenges for EL. As Tata & Di Eugenio (2010) point out, they are difficult to discover because they may not be restricted to a single Noun Phrase or may be abbreviated (by means of acronyms, dropping entire words or even full rephrasing). Additionally, a specific trait of Music texts is the fact that one song may have many covers by many different artists and, according to our evaluation, it may be difficult even for a human to identify what *version* of the song the writer is referring to. Furthermore, availability of EL testbeds in general Usbeck et al. (2015), and in the Music domain in particular Gruhl et al. (2009), is scarce, making it very difficult to evaluate novel systems and approaches. Hence, it is difficult to know how well a certain method, which may work well for generic texts, will perform on Music data.

Despite the current context of scarcity of both EL systems and evaluation benchmarks in the Music domain, there are some exceptional cases in which these issues were addressed, such as: (1) Detecting Music entities (e.g. songs or bands) on informal text Gruhl et al. (2009); (2) Applying Hidden Markov Models for discovering Music entity mentions in Chinese corpora Zhang et al. (2009); or (3) Recognizing musical entities in the context of a relation extraction pipeline Oramas et al. (2015b).

We argue that the problem of precision in detecting musical entities may be tackled by leveraging a combination of several generic EL off-the-shelf systems. Simply put, we hypothesize that if two or more generic systems annotate with the same URI an entity mention, the probability of this annotation to be correct increases. To the best of our knowledge, very little effort has been put in exploiting this *agreement* feature. One of the reasons may be that, as of now, most EL systems *speak their own language*, partially due to the fact that each of them points back to different KBs, and hence their output is heterogeneous and cannot be directly compared, let alone combine. This has motivated research towards unification frameworks for evaluation of EL. For instance, Cornolti et al. put forward a benchmarking framework for comparing EL systems. Moreover, Rizzo et al. (2014) describe a system aimed at combining the output of the different NER systems. Finally Usbeck et al. (2015) present GERBIL, an evaluation framework for semantic EL based on Cornolti et al..

In this chapter we aim to provide a twofold answer to the challenges described above, and bridge the gap between the Music domain and EL. Specifically, we present ELMD, an automatically constructed corpus where named entities are classified as any of four predefined *musical categories*, namely SONG, ALBUM, ARTIST, and RECORD LABEL, by leveraging the hyperlinks present in a set of artist biographies. Then, we further enrich ELMD by performing EL and automatically annotating a large portion of the entities with their DBPEDIA URI. The source data used in this paper comes from the music website and social network Last.fm[5]. To the best of our knowledge, this is the first attempt to provide an annotated large-scale corpus of linked entities in the music domain. Finally, a subset of 200 documents from ELMD is manually annotated to

[5]http://www.last.fm

provide a comprenhensive dataset of annotated documents. This latter dataset have been uses in the context of the 3rd Open Knowledge Extraction Challenge ?.

The final resource amounts to 47,254 sentences, in which 92,930 entities are categorized into the aforementioned *musical categories*, and 64% of them are disambiguated and linked to DBpedia. We achieve a Precision score of 97% in the most restrictive setting, in which our approach manages to annotate more than 31,000 entities.

In the remainder of this paper, we first introduce ELVIS (Entity Linking Voting and Integration System), our EL integration and agreement approach. Then, we describe the text corpus we compiled from the Last.fm website and how it is combined with ELVIS. In the next step, the obtained dataset is evaluated. Finally, we describe the resulting output of our system: The ELMD dataset.

3.2 Music Knowledge Bases

MusicBrainz and Discogs are two paramount examples of manually curated MKBs. They are open music encyclopedias of music metadata built collaboratively and openly available. MusicBrainz, in addition, is regularly published as Linked Data by the LinkedBrainz project[6].

As for generic KBs based on Wikipedia, such as the ones described earlier, these include a remarkable amount of music data, such as artist, album and song biographies, definitions of musical concepts and genres, or articles about music institutions and venues. However, their coverage is biased towards the best known artists, and towards products from Western culture. Finally, let us refer to the notable case of Grove Music Online[7], a music encyclopedia containing over 60k articles written by music scholars. However, it has the drawback of not being freely open, as it runs by subscription.

Other than the aforementioned curated repositories, to the best of our knowledge, there is not a single automatically learned open MKB. A first step in this direction was taken in Sordo et al. (2015); Oramas et al. (2014), applying RE techniques to big datasets of music related texts extracted from the web. Moreover, in Oramas et al. (2015a), a Flamenco MKB is created by combining data from curated KBs and information extracted from blogs and websites.

Despite their scarcity, MKBs are becoming increasingly popular in MIR applications, such as artist similarity and music recommendation Celma & Serra (2008); Oramas et al. (2015c); Leal et al. (2012); Ostuni et al. (2015). MKBs have also been exploited as sources of explanations in music recommender systems. According to Celma & Herrera (2008), giving explanations of the

[6] http://linkedbrainz.org/

[7] http://www.oxfordmusiconline.com

recommendations provides transparency to the recommendation process and increases the confidence of the user in the system. In Passant (2010), explanations of recommendations are created by exploiting DBpedia's structured information, whilst in Sordo et al. (2015), explanations are based on an automatically learned MKB.

3.3 ELVIS

In this section we describe ELVIS, the generic integration framework for Entity Linking, which is leveraged for the construction of ELMD. First, we describe our Entity Linking research problem and provide an intuition on how this may be surmounted via an agreement scheme. Then, we provide details on the main modules integrating ELVIS, highlighting the possible cases of agreement and disagreement over the EL systems that are integrated in our framework.

3.3.1 Argumentum ad Populum in EL

Our method relies on the *argumentum ad populum* intuition, i.e. if two or more different EL systems perform the same prediction in linking a named entity mention to its entry in a reference KB, the more likely this prediction is to be correct. We put this intuition into practice by combining the output of three well-known systems, namely DBpedia Spotlight Mendes et al. (2011) , Tagme Ferragina & Scaiella (2012) and Babelfy Moro et al. (2014a), whose agreement (or disagreement) when disambiguating an in-text entity mention is taken as an agreement-driven *confidence score*. These specific tools were chosen for being considered state-of-the-art EL systems and for being well known in the NLP community. However, ELVIS can easily incorporate any additional system. We also selected these tools because entities identified by all of them can be easily referenced to DBpedia URIs. Although there are other knowledge bases (e.g. MusicBrainz) with substantially more musical entities than DBpedia, to the best of our knowledge, there is no EL tool that works with these domain specific knowledge bases.

Let us briefly describe each of the selected EL systems:

- **DBpedia Spotlight** Mendes et al. (2011) is a system for automatically annotating text documents with DBpedia URIs, finding and disambiguating natural language mentions of DBpedia resources. DBpedia Spotlight is shared as open source and deployed as a Web service freely available for public use[8]. DBpedia Spotlight gives as a result the DBpedia URI, start and end char positions, the value of the rdf:type property, and a confidence score for for each prediction.

[8]https://github.com/dbpedia-spotlight

- **TagMe** Ferragina & Scaiella (2012) is an EL system that matches terms with Wikipedia link texts and disambiguates them using the in-link graph and the Wikipedia page dataset. Then, it performs a pruning process by looking at the entity context. TagMe is available as a web service[9], and provides the Wikipedia page id, Wikipedia categories, and a confidence score.

- **Babelfy** Moro et al. (2014a) is an EL and Word Sense Disambiguation based on non-strict identification of candidate meanings (i.e. not necessarily exact string matching), together with a graph based algorithm that traverses the BabelNet graph and selects the most appropriate semantic interpretation for each candidate. Babelfy is available as a web service[10]. Its output is based on the corresponding BabelNet synset of the disambiguated mention. If the synset references to a Wikipedia page, it returns the Wikipedia URL, the DBpedia URI, as well as Wikipedia categories.

While these tools have proven highly competitive on their own, in this paper we explore the gain in performance obtained by combining them together, and apply global agreement-driven decisions on the Last.fm corpus.

3.3.2 'Translating' EL Formats

In order to have each EL system *speak the same language* for measuring agreement in their predictions, output homogenization is required. This is not a trivial task, as each EL approach may be based on a different reference KB, the offsets may be computed differently, and so on. For instance, DBpedia Spotlight links entity mentions via DBpedia URIs, whereas Tagme provides Wikipedia page IDs, and Babelfy disambiguates against BabelNet Navigli & Ponzetto (2012) and its corresponding BabelNet synsets. We attempt to surmount this heterogeneity as follows: First, we retrieve DBpedia URIs of every named entity. There are some considerations to be taken into account, however: (1) Character encoding differs from system to system, which we address by converting the character encoding of the retrieved URI to UTF-8; (2) Several URIs may refer to the same DBpedia resource. We solve this specific issue thanks to the transitive redirections provided by DBpedia. If a URI has a transitive redirection, it is replaced by the redirected URI. (3) Note that, in the case of Tagme, only Wikipedia page IDs are provided, which we can straightforwardly exploit to map entity mentions to their DBpedia equivalent. Finally, and after surmounting compatibility issues among systems, we retrieve DBpedia types (`rdf:type property`) for all entities. This *type* information is further used in the creation of ELMD.

[9] http://tagme.di.unipi.it/
[10] http://babelfy.org/guide

After successfully providing a process which harmonizes the output of EL systems, it is possible to compute the degree of agreement among them, which will become our system's confidence score. We define the following set of *agreement heuristics* to set such score for each linking prediction (an overview of the workflow of ELVIS is provided in Figure 3.1).

- **Full Agreement** (++) When all systems detect an entity with the same URI and offset.
- **Partial Agreement** (+) When more than one but less than all systems detect an entity with the same URI and offset. Outliers (i.e. systems performing a different prediction) may detect a different entity or may not detect anything.
- **Singleton Decision** (−) When only one system detects an entity for a given text offset.
- **Disagreement** (−−) When more than one system performs a linking over the same text offset, but all of their predictions are different.

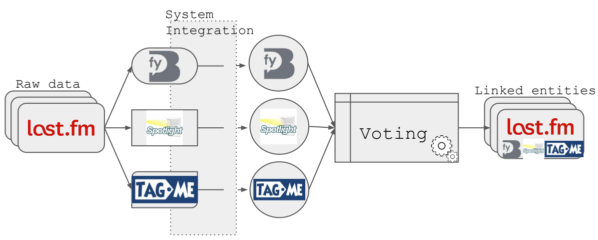


Figure 3.1: ELVIS Workflow

3.4 From Last.fm to ELMD

In what follows, we describe the original data gathered from Last.fm, and the process to apply the integration framework described in Section 4.2, in order to construct a highly precise benchmarking dataset for EL in the Music domain.

In Last.fm, users may add relevant biographical details to any artist's main page in the form of a *wiki*. These edits are regularly moderated. Furthermore, artist biographies are often enriched with hyperlinks to other Last.fm Artist, Album, Song and Record Label pages, similarly as with Wikipedia hyperlinks. Our purpose is to leverage this meta-information to automatically construct a dataset of Music-specific annotated named entities.

We crawled artist biographies from Last.fm in March 2015, and gathered 13,000 artist biographies, which comprise 47,254 sentences with at least one hyperlink, amounting to a total of 92,930 links. These may be broken down as follows: (1) 64,873 hyperlinks referencing Artist pages; (2) 16,302 to Albums; (3) 8,275 to Song pages; and finally (4) 3,480 hyperlinks referencing Record Labels. This *type* information is extracted thanks to the structure of each link's URL, as it includes in its path the category of the annotated entity. Consider, for example, the following sentence:

> After their debut The Intelligence got signed to *In the Red Records*.

Here, we may infer that the entity *In the Red Records* is a Record Label, thanks to its LAST.FM URL: `http://www.last.fm/label/In+the+Red+Records`. This information is extracted from the whole LAST.FM corpus for those entities falling in one of the four *musical categories* previously defined.

3.4.1 Data Enrichment

For the creation of the ELMD dataset, the crowdsourced annotations extracted from LAST.FM biographies are combined with decisions made by ELVIS and its voting framework.

Every entity mention annotated in the LAST.FM corpus is a candidate to be included in ELMD. The challenge is to assign to each entity its correct DBpedia URI. We approach this problem by leveraging (1) The DBpedia URI assigned by ELVIS, (2) The *agreement score* for that prediction, as well as (3) The *type* information derived from the entity's LAST.FM URL. Our intuition is that the higher the *agreement score*, the more likely the prediction is to be correct. Likewise, we also hypothesize that if a linking decision made by ELVIS coincides in *type* with the original LAST.FM annotation, it is more likely to be correct. Since there is no direct mapping between LAST.FM and DBpedia types, we manually set the type equivalences shown in Table 3.1.

Regarding the *agreement score*, it corresponds to the number of systems that agreed in a decision (see **Score** column in Table 3.2). Note that an *agreement score* of 1 may be caused either by cases in which only one system detected an entity mention, or when there is disagreement among systems, but one and only one of them coincides in *type* with the original LAST.FM annotation (last row in Table 3.2).

As for *type value*, this is a binary value (*type-equivalent* or *type-discrepant*) based on coinciding types between LAST.FM URLs and ELVIS decisions.

3.5 Evaluation

Considering the different possibilities of agreement across the systems integrating ELVIS, there are in total 7 possible configurations: 1 with **full agreement**

Last.fm type	DBpedia type
Song	DBpedia:Song, DBpedia:Single, Yago:Song
Album	DBpedia:Album, Yago:Album, Schema:MusicAlbum
Artist	DBpedia:MusicalArtist, DBpedia:Band, Schema:MusicGroup, Yago:Musician, Yago:Creator, DBpedia:Artist
Record Label	DBpedia:RecordLabel

Table 3.1: Type equivalence

Context	Last.fm type	Tagme	Babelfy	Spotlight	Score	Type Eq.
and the academic minimalism of **Steve Reich**	Artist	Steve_Reich (type:artist)	Steve_Reich (type:artist)	Steve_Reich (type:artist)	3	type-equivalent
The new album **Hypocrisy** followed shortly thereafter	Album	—	Hypocrisy (type:band)	Hypocrisy (type:band)	2	type-discrepant
The third album **Lucifer Songs**, opened new and unexpected doors	Album	—	Lucifer_Songs (type:album)	—	1	type-equivalent
The band's debut album, **Cookies**, was released on 14 May 2007	Album	HTTP_cookie (type:unknown)	Cookies (type:album)	—	1	type-equivalent (only Babelfy)

Table 3.2: Agreement examples

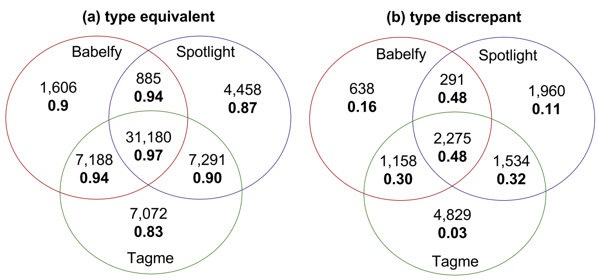


Figure 3.2: Number of entities and precision of the manual evaluation. Note the major differences in Precision between *type-equivalent* and *type-discrepant* systems.

	Agreement	Precision	No. Entities
type-equivalent	$= 3$	0.97	31,180
	≥ 2	0.96	46,544
	≥ 1	0.94	59,680
all	$= 3$	0.94	33,455
	≥ 2	0.90	51,802
	≥ 1	0.81	72.365

Table 3.3: Precision and number of entities with this value of precision. *Type-equivalent* implies entities from the type-equivalent configuration only, whilst *All* implies all entities regardless their type information.

(score= 3); 3 with **partial agreement** (score = 2); and 3 **singleton** configurations (score= 1). Moreover, considering also the two possible values of *type agreement*, namely `equivalent` and `discrepant`, we have a total number of 14 configurations. Figure 3.2 provides a visual overview of these configurations, where we show both Precision scores for each configuration (in bold) in addition to the number of entities disambiguated with ELVIS in each case.

Musical Category	Annotations	Distinct Entities	Avg. words	Most frequent entity
Song	3,302	2,823	2.81	Shine (6)
Album	7,872	6,897	2.69	Like Drawing Blood (6)
Artist	46,337	17,535	1.88	The Beatles (160)
Record Label	2,169	815	1.94	Sub Pop (33)

Table 3.4: Statistics of the linked entities in ELMD. We report, for each *musical category*, the total number of annotations linked to DBpedia, number of unique entities, average number of words per entity mention, and most frequently annotated entity (along with its frequency).

We evaluated 100 randomly selected entity samples (25 for each of the four Music categories we consider) from each one of the 14 possible configurations, and asked an evaluator with computational linguistics background to manually assess the correctness of the 1,400 predictions. From scores obtained from manual evaluation, we estimated Precision for the whole ELMD dataset with different ranges of *agreement score* as well as two options *type*-wise (see Table 3.3). The precision value for all the entities is computed proportionally according to the number of entities and the precision obtained in the manual evaluation for the *type-equivalent* and *type-discrepant* settings, hence these can be seen as Micro Average Precision numbers.

We observe that the *type-equivalent* configuration yields much better Precision with only a slight tradeoff in terms of Recall. Therefore, we decided to select for the final ELMD dataset only those URIs stemming from a *type-equivalent* setting where *agreement score* is equal or greater to 1. This ensures a Precision of at least 0,94 in terms of Entity Linking. Moreover, a manual survey of false positives in the highest scoring setting (*agreement score*= 3 and *type-equivalent*) showed that these are cases in which even a human annotator may not find it trivial to correctly find the correct entity to those entity mentions. One of these cases are those in which ELVIS is presented with and entity mention that on surface may refer to either an Artist or an Album named after the artist or band itself. An actual case of false positive in our evaluation dataset is the following sentence:

> Her debut album , *Kim Wilde*, (released on RAK records) came out in July 1981 and stayed in the U.K. album charts for 14 weeks, peaking at number 3 and getting much acclaim.

Here, the entity *Kim Wilde* should be disambiguated as the Album with the same name as the artist, but ELVIS incorrectly assigned the Artist's DPpedia URI: `dbpedia.org/resource/Kim_Wilde`. In ELMD there are 50 cases where the same surface text is correctly linked to an Artist entity in some sentences, and to a Song entity in others. Similar ambiguous cases involving Artist and Album (148) and Song and Album (95) are correctly resolved by our system. These particularly challenging cases may be interesting for training Music specific EL algorithms.

Another interesting source of false positives comes between musical entities and equally named entities (not necessarily related to Music). In cases in which the latter are more popular in a reference KB, e.g. their associated node in the graph may have higher connectivity, may become prioritized by disambiguation EL algorithms that consider graph connectivity as a feature. Consider the following sentence:

> He is becoming more and more in demand for his remixing skills; working for the likes of Justin Timberlake and Armand van Helden, and labels including *Ministry Of Sound*, Defected and Intec, to name a just a few.

Here, the entity *Ministry of Sound* refers to a Record Label, a spin-off of the well-known club, which is the entity that was incorrectly assigned: `dbpedia.org/resource/Ministry_of_Sound` Cases like this would require, first, to ensure that the different entities derived from *Ministry of Sound* (such as the Record Label or a clothing brand of the same name) exist in a reference KB, and second, to exploit contextual information so that a correct decision is made. A similar situation happens when song or album names may be confused with very common words or expressions (e.g. 'Easy', 'Stupid', 'Sad song', 'If', 'Be there'). ELMD is rich in challenging cases like these.

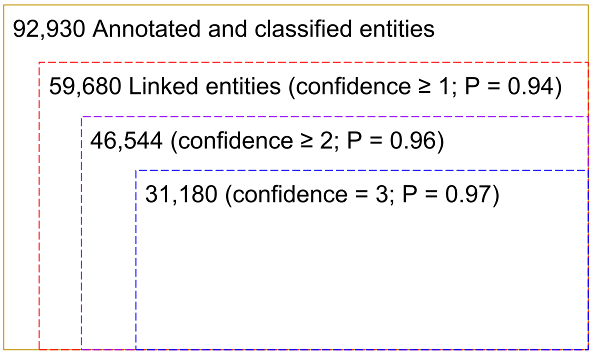


Figure 3.3: ELMD Overview. Number of entities, confidence score and precision values in different subsets of the dataset.

3.6 ELMD2

Following the same methodology, a second set of artist biographies was gathered from Last.fm.

From this set of documents, 200 were manually annotated by one music expert annotator.

3.7 Conclusion and Discussion

In this chapter we have described two main contributions. First, for the task of Entity Linking, we have presented an integration framework called ELVIS which, based on a voting procedure which leverages decisions made by an arbitrary number of off-the-shelf EL systems, provides high confident entity disambiguations. Currently, ELVIS incorporates three state-of-the-art systems, namely DBpedia Spotlight, Tagme and Babelfy, and can be easily extended with additional systems. The *ELVIS* code is available at https://github.com/sergiooramas/elvis. Second, we have leveraged the potential of ELVIS for the creation of a novel benchmarking dataset for EL in the Music domain, called ELMD. This corpus comes from a collection of Last.fm artist biographies, and contains 47,254 sentences with 92,930 annotated and classified entity mentions (64,873 Artists, 16,302 Albums, 8,275 Songs and 3,480 Record Labels). From this set of entity mentions, 59,680 are linked to DBpedia (see Table 3.4), with a precision of at least 0,94. In addition, by setting up a higher confidence threshold it is possible to obtain a subset of ELMD that prioritize higher Precision by sacrificing Recall (see Figure 3.3). The ELMD dataset together with the evaluation data can be downloaded from http://mtg.upf.edu/download/datasets/elmd.

CHAPTER 4

Automatic Construction of Music Knowledge Bases

4.1 Introduction

Knowledge Representation and Reasoning is a key enabler of Intelligent Systems Suchanek et al. (2007), and plays an important role in Natural language Understanding (NLU) Baral & De Giacomo (2015). In this paper, we focus on an important aspect of NLU, which is *how to make sense* of the data that is generated and published online on a daily basis. This data is mostly produced in human-readable format, which makes it unsuitable for automatic processing. Considering that deep understanding of natural language by machines seems to be very far off Cambria & White (2014), there is great interest in formalizing unstructured data, and Knowledge Bases (KBs) are a paradigmatic example of large-scale content processed to make it machine readable.

We may define a KB as a repository of knowledge organized in a predefined taxonomic or ontologic structure, potentially compatible with other KBs, thus contributing to the Linked Open Data initiative[11]. These KBs may be designed to represent unconstrained knowledge, or a single domain of interest. This representation is formalized either manually, automatically, or with a combination of both.

In the music field, previous attempts to formalize knowledge have led to manually curated KBs, such as MusicBrainz[12] and Discogs[13]. Moreover, generic resources like Wikipedia include a sizable amount of music data. By extension, KBs based on Wikipedia, such as DBpedia[14] or Freebase[15], also

[11] http://linkeddata.org/
[12] http://musicbrainz.org/
[13] http://www.discogs.com
[14] http://dbpedia.org
[15] https://www.freebase.com/

include this information, but converted to machine readable formats. However, their coverage is limited as they are not specifically targeting the music domain, and they may miss novel or underground artists, albums or songs, and also musical entities that are only locally relevant. On the other hand of the spectrum, we find that the Web is an incredible rich source of music information about any artist, genre or region. However, leveraging web-based knowledge is a research problem in itself, as it is mostly encoded as unstructured text Oramas (2014). The challenge now lays on how to interpret all the information that is published online in a comprehensive way.

We propose to address this research problem by putting forward an NLP pipeline designed to construct a Music KB (MKB) entirely from scratch in an automatic and unsupervised manner. Our method is able to generate a fully disambiguated MKB with entity mappings against DBpedia and MusicBrainz. All relations have a relation pattern derived from a Relation Extraction (RE) procedure backed up by an algorithm that performs the following steps: (1) Morpho-syntactic rule-based *filtering*; (2) Syntactic dependency-based *clustering*; and (3) Relation *weighting* based on statistical evidence.

We validated our methodology on a large collection of documents in the music domain, obtained from *songfacts.com*, a website that collects "tidbits" (short stories) about songs. We accompany our data release with an intrinsic evaluation carried out on each component of the algorithm, as well as an extrinsic evaluation which consists of a experiment on interpretation of music recommendations, where our automatically learned MKB is used to provide explanations to song recommendations in *natural language*.

Our experimental results indicate that our system is able to extract *high quality* relations (Precision >= 0.8) as well as *novel knowledge*. We unveil thousands of relations absent in both large-scale generic KBs, as well as in music specific resources. Moreover, the recommendation explanation experiment shows that explanations based on the newly learned KB have a positive impact in user experience.

We release the learned MKB at several confidence levels, together with the evaluation data used in the experiments described in this paper. The code for the complete RE pipeline is also released as open source software under MIT license.

The main contributions of this paper are summarized as follows:

- We address for the fist time the problem of automatic construction of MKBs from plain text.
- We put forward a novel approach for clustering relations, based on patterns derived from syntactic dependencies.
- We present a new confidence measure over all extracted relations and

demonstrate its discriminative power.

- We showcase the utility of our method by creating a high quality MKB with *novel knowledge*.

- We demonstrate the usefulness of our automatically constructed MKB for providing explanations in natural language in the context of music recommendation.

This is a joint effort by the author of this thesis, and X, both researchers in the UPF Department of Information Technologies. This collaboration is framed within the Maria de Maeztu strategic program, specifically in the Music Meets NLP project.

4.2 Method

We propose a comprehensive pipeline that learns a full-fledged MKB taking as input raw text collections. The experiments we report in this paper are the result of applying our method to a dataset of plain text extracted from the Songfacts[16] website (see Section 4.3.1). This is a well suited resource both for KB learning and as a testebed for RE due to its specificity. Essentially, Songfacts documents, while not being as rigid as encyclopedic text or newswire text, remain well-formed, sentences make sense, and there is no need for *ad-hoc* preprocessing (as it is required in social networks, e.g. Twitter). Our method, however, can be ported with little effort to music-related corpora of different registers.

4.2.1 Notation

Our method focuses on the extraction of semantic relations between pairs of linked entities (e.g. *Born in the USA*$_{dbr}$, *Bruce Springsteen*$_{dbr}$[17]), which are in turn associated to specific entity types (e.g. *Album*, *MusicalArtist*). In our KB, a relation r is defined by the tuple $\langle \boldsymbol{e_d}, \boldsymbol{e_r}, \boldsymbol{\upsilon_d}, \boldsymbol{\upsilon_r}, \boldsymbol{p}, \boldsymbol{c} \rangle$, where $\boldsymbol{d}$ and $\boldsymbol{r}$ refer to domain and range positions, $\boldsymbol{e_d}$ and $\boldsymbol{e_r}$ to the entities involved in the relation, $\boldsymbol{\upsilon_d}$ and $\boldsymbol{\upsilon_r}$ to their associated entity types, $\boldsymbol{p}$ to a relation pattern, and $\boldsymbol{c}$ to a cluster pattern. A relation pattern is a relation label that may be used in one or several relations (e.g *was recorded by frontman*, *was recorded by singer/songwriter*). Relation patterns with similar semantic and syntactic characteristics may be grouped into cluster patterns (e.g. *was recorded by*). $\mathcal{R}$ denotes the set of all extracted relations included in the KB. For each $r \in \mathcal{R}$, triples of different nature can be constructed by arbitrarily combining elements in r.

[16] `http://www.songfacts.com`

[17] We use the *dbr* subscript to refer to disambiguated entities linked to DBpedia resources.

- $t_{\boldsymbol{p}} : \langle \boldsymbol{e_d}, \boldsymbol{p}, \boldsymbol{e_r} \rangle$, e.g. {*Born in the USA*$_{dbr}$ - *was recorded by frontman* - *Bruce Springsteen*$_{dbr}$}.

- $t_{\boldsymbol{c}} : \langle \boldsymbol{e_d}, \boldsymbol{c}, \boldsymbol{e_r} \rangle$, e.g. {*Born in the USA*$_{dbr}$ - *was recorded by* - *Bruce Springsteen*$_{dbr}$}.

- $\tau_{\boldsymbol{p}} : \langle \boldsymbol{\upsilon_d}, \boldsymbol{p}, \boldsymbol{\upsilon_r} \rangle$, e.g. {*Album* - *was recorded by frontman* - *MusicalArtist*}.

- $\tau_{\boldsymbol{c}} : \langle \boldsymbol{\upsilon_d}, \boldsymbol{c}, \boldsymbol{\upsilon_r} \rangle$, e.g. {*Album* - *was recorded by* - *MusicalArtist*}.

Finally, different subsets of $\mathcal{R}$ may be constructed by selectively filtering all $r \in \mathcal{R}$.

- $\mathcal{R}_p = \{r_1^p, ...r_n^p\}$ All relations with a specific relation pattern p.

- $\mathcal{R}_c = \{r_1^c, ...r_n^c\}$ All relations with a specific cluster pattern c.

- $\mathcal{R}_{\boldsymbol{\tau_p}} = \{r_1^{\tau_p}, ...r_n^{\tau_p}\}$ All relations with a specific relation pattern, and domain and range entity types.

- $\mathcal{R}_{\boldsymbol{\tau_c}} = \{r_1^{\tau_c}, ...r_n^{\tau_c}\}$ All relations with a specific cluster pattern, and domain and range entity types.

In what follows, we describe a method for acquiring new entities, types and relations, and combining them in a meaningful way for KB construction.

4.2.2 Morphosyntactic Preprocessing

Our morphosyntactic preprocessing module takes as input a collection of text documents in the music domain. First, sentence splitting and tokenization is carried out thanks to the *Stanford NLP tokenizer*[18]. Next, a dependency parse tree is obtained via the MATE Parser, described in Bohnet (2010). We justify the use of the latter because of the richness of its tagset, as well as performance in terms of accuracy and speed, which were appropriate for the task at hand.

In a dependency tree, each node includes information, at least and depending of the model and the language, about surface and lemmatized forms, along with its part-of-speech. Each edge in the tree is labeled with a dependency relation such as *subject* or *noun modifier* (an example is shown in Figure 4.1).

[18] `http://nlp.stanford.edu/software/tokenizer.shtml`

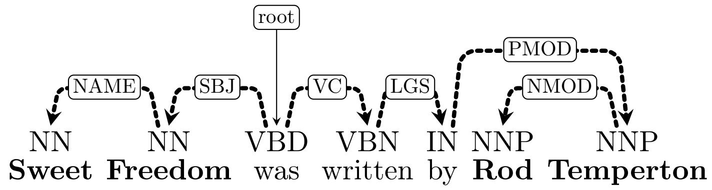


Figure 4.1: Example sentence with dependency parsing tree

4.2.3 Semantic Processing: Entity Linking

Entity Linking (EL) acts as a semantic bridge between plain text and a reference knowledge inventory. While there are a number of popular EL systems which are not bound to any domain or discipline, there is no benchmark of such systems in the music domain Oramas et al. (2016). Therefore, we do not know *a priori* how well each of them works in music corpora. Musical entities may raise a plethora of challenges, derived mostly from ambiguity and polysemy. For example, an album may have the same name as the band who recorded it (e.g. *Weezer* the band and their first album). Moreover, an artist, a song or an album may have words or expressions much more common in another domain or area of knowledge (e.g. *Berlin, The Who*). Thus, the choice of the best EL algorithm or off-the-shelf tool(s) is crucial, as potential errors may propagate throughout the different modules and hinder considerably the quality of the resulting KB.

Among the available EL systems we considered, namely Tagme Ferragina & Scaiella (2010), Babelfy Moro et al. (2014c) and DBpedia Spotlight Mendes et al. (2011), we opted for the latter, as it has shown to be the least prone to errors in musical texts (further details are provided in Section 4.4.1).

Adding Co-references

In the music domain, prototypical factoid documents such as artist biographies, album reviews, or song tidbits, normally refer to one specific entity. Based on this observation, we may exploit co-referential pronouns and *resource-specific co-references*, replacing them by the name of the reported entity. A similar approach is used in Voskarides & Meij (2015), where the frequency of pronouns "he" and "she" is computed in every document (Wikipedia articles in this specific case) to determine the entity's gender, and then, these pronouns are replaced by the entity title. Similarly, in Oramas et al. (2014), a gender identifier web service is used to determine the gender of subjects in artist biographies as part of a RE pipeline.

We have observed an exploitable *resource-specific co-reference* in music reviews, where terms like "this album" or "the song" can be replaced by the document's title. In the dataset used for the experiments (see Section 4.3.1), the expressions "this song" and "the song" are replaced with the name of the song as it appears in the document, and disambiguated with the URI of the entity they unequivocally refer to.

Co-reference resolution is a difficult and crucial task in NLP, affecting tasks such as Information Extraction Soon et al. (2001) or document summarization Saggion & Gaizauskas (2004). It is also sensitive to the domain in which it appears (see, for instance, the case of the patents domain Bouayad-Agha et al. (2014)). We acknolwedge the difficulty of this task. However, while addressing this problem in its entirety is out of the scope of this paper, these strategies allow us to increase coverage of entity mentions while maintaining a high precision.

Type Filtering

In DBpedia, most resources are associated with one or more types via the `rdf:type` property. In addition, among the different types present in DBpedia (coming from the DBpedia ontology, Yago types, or `schema.org`), the DBpedia ontology provides a relatively small and tidy taxonomy of 685 classes based on Wikipedia infoboxes. Other KBs such as Yago or Freebase have their own ontological structure, which is in general broader and noisier. MusicBrainz, in contrast, has a very narrow set of entity types.

This type information can be exploited in order to narrow down the set of allowed types for a given candidate and its potential annotations. In this way, we ensure that all entities will be, at least, related to the music domain. Restricting the search space to types such as Artist or Song reduces considerably the number of errors derived from cross-domain ambiguity. For instance, the EL system detects a substantial amount of entities whose DBpedia type is *FictionalCharacter*, which are in most of the cases misleading song titles or band names with fictional characters of the same name. This situation is observed also with other types of entities such as *Athlete*, *Species* or *Disease*.

Depending on the envisioned application of the KB resulting from our pipeline, the predefined set of entity types may vary. In our case we restricted them to Musical Artists, Other Artists, Songs, Albums, Genres, Films and Record Labels. In Table 4.1 we present the mapping between the DBpedia ontology, MusicBrainz entity types and our selected set of types.

4.2.4 Syntactic Semantic Integration

The information obtained from the syntactic and semantic processes is combined into a graph representation of the sentence. For each music entity identified during the semantic enrichment step (Section 4.2.3), all nodes in the dependency tree with a correspondence with an entity mention are collapsed into one single node: *Sweet* and *Freedom* into *Seet Freedom (Album)*, and *Rod* and *Temperton* into *Rod Temperton (Artist)*. Figure 4.2 shows the resulting syntactic-semantic representation of a sentence.

Our MKB	**DBpedia ontology**	**MusicBrainz**
MusicalArtist	Person/Artist/MusicalArtist Organization/Band Writer/MusicComposer Writer/SongWriter	Artist
OtherArtist	Person/Artist (¬ MusicalArtist) Person/Writer(¬ MusicComposer & ¬ SongWriter)	—
Album	Work/MusicalWork/Album	Release
Song	Work/MusicalWork/Song Work/MusicalWork/Single	Recording Work
Genre	TopicalConcept/Genre	—
Film	Work/Film	—
RecordLabel	Agent/Organization/Company/RecordLabel	Label

Table 4.1: Type mapping.

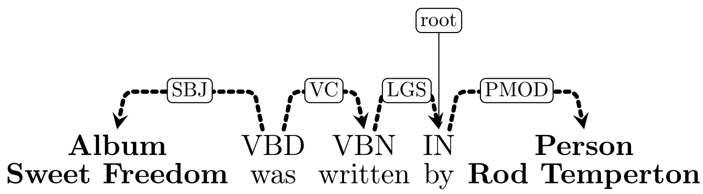


Figure 4.2: Semantic integration on syntactic dependencies.

4.2.5 Relation Extraction and Filtering

Our approach to RE is lightweight, unsupervised and rule-based. Having syntactic and semantic information available, potential relations between entities may be discovered by traversing the dependency tree. Two entities in such tree are considered to be related if there is a path between them that does not contain any other entity in between, and does not contain parentheses. If there is more than one path, we consider only the shortest path as the most representative path of the relation.

Our method encodes a relation pattern between two entities as all words in the shortest path between them. In the example provided in Figure 4.2, the shortest path between *Sweet Freedom* and *Rod Temperton* contains the words *was*, *written* and *by*.

While RE via shortest path in syntactic trees is common practice in the literature Bovi et al. (2015b); Moro & Navigli (2012); Nakashole et al. (2012), not all shortest paths are valid, and incorrect relations may be extracted from overly long and syntactically complex sentences. We aim at surmounting these problems by defining three filtering heuristics over surface forms (*lemma-paths*), part-of-speach patterns (*pos-paths*), and labels of syntactic dependencies (*dependency-paths*).

First, we filter out all relations with reporting verbs (e.g. “say”, “tell” or “express”) in the lemma-path. The intuition being that sentences with these verbs are by definition syntactically complex, and semantic relations in them may not be encoded via shortest paths. We illustrate this with the following sample sentence, where the relation extracted with syntactic tree traversal by means of shortest path would be incorrect:

> **Sentence:** `Nile Rodgers` *told* NME that the first album he bought was Impressions by `John Coltrane`.
>
> **Relation**: `nile_rodgers` told that was impressions by `john_coltrane`

Second, we only selected relations where the syntactic function that connects in the dependency-path the fist entity with the first word of the relation pattern is a subject (which may be preceded by a nominal modifier or an apposition), a direct or indirect object, a predicative complement or a verb chain. When this condition holds, the relation is considered *valid*. If the above condition does not hold, an extra validation step is applied over the pos-path in order to capture relations without verbs, which seem to be idiosyncratic of the music domain, e.g. $\langle e_d, \texttt{frontman of}, e_r\rangle$, $\langle e_d, \texttt{drummer}, e_r\rangle$, or $\langle e_d, \texttt{guitarist and singer}, e_r\rangle$.

4.2.6 Dependency-Based Loose Clustering

In this section we describe a simple but powerful clustering algorithm aimed at reducing the number of relation patterns in the KB.

Let us consider the following three relation patterns: (1) *was written by blunt producer*, (2) *was written by singer/producer*, and (3) *was written by manager and guitarist*. Intuitively, these three relation patterns seem to be semantically similar, and if all of them were expressed as *was written by*, the original meaning would not be lost, and the set of relations would become more compact.

This observation, which we found to occur quite frequently, motivated the inclusion of a *dependency-based loose clustering* module. First, we perform a second run of dependency parsing over all relation patterns extracted by our system, aiming at discovering their root node. We apply this second run because the root of the original sentence does not need to correspond with the relation pattern's root. Then, our algorithm considers all possible paths from the root to every leaf node of the relation pattern dependency tree, and selects the path that complies with a predefined syntactic constraint (e.g. a sequence of verbs plus adverb or preposition, or adverb plus nominal and preposition modifiers) based on regular expressions of syntactic labels. The sequence of tokens that matches this regular expression constitutes the cluster pattern. The complete set of defined regular expressions is included in the released source code.

As an illustrative case, consider the extracted relation pattern *is track was released on label* from the sentence *`Sing Out The Song` is the 7th track on Wishbone Four which was released in the UK May 1973 on the `MCA` label.* After re-parsing the relation pattern, we obtain the parse tree shown in Figure 4.3 and a cluster pattern over those nodes in the dependency tree that satisfy one of the regular expressions crafted in the aforementioned syntactic constraint. Finally, the obtained relation is **Sing_out_the_song** *was released on label* **MCA**.

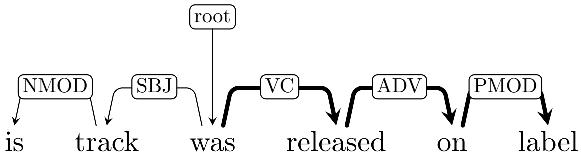


Figure 4.3: Example of a parsed relation pattern $p \in \mathcal{P}$ and a valid cluster pattern (bold).

Filtering out spurious information in OIE following similar approaches has proven effective while not being computationally expensive Fader et al. (2011).

Ours is a *loose clustering* method because it does not enforce a pattern to fully match all rules, but rather allows partial matching. This module provides an enrichment of all $r \in \mathcal{R}$ such that $r = \langle e_d, e_r, \upsilon_d, \upsilon_r, p, c \rangle$, where c is the cluster pattern derived from the relation pattern p. A relation cluster is the set of all relations with the same cluster pattern, and is denoted as $\mathcal{R}_c$.

Cluster pattern c	**Typed cluster pattern τ_c**	**Relation triples t_p**
was written by	S *was written by* MA	s1 *was writen by artist* ma1
		s2 *was written by composer* ma2
		s3 *was written by singer* ma2
		s4 *was writen by* ma1
		s5 *was written by frontman* ma3
	A *was written by* MA	a1 *was written by frontman* ma3
		a2 *was written by guitarist* ma1
		a3 *was written by artist* ma2
		a4 *was written by frontman* ma5

Table 4.2: Example of a relation cluster $\mathcal{R}_c$, where c = *was written by*. *S* refers to Song, *MA* to MusicalArtist and *A* to Album types, whilst *sX* refers to Song, *maX* to MusicalArtist and *aX* to Album entities.

4.2.7 Scoring

So far, our approach has identified entity mentions in text and has linked them in meaningful relations, filtering out those that did not comply with predefined linguistic rules. We incorporate one additional factor $score(r)$ that takes into account statistical evidence computed over $\mathcal{R}$. It has three main components, which we flesh out as follows.

We hypothesize that the relevance of a cluster may be inferred by the number and proportion of triples it encodes, and whether these are evenly distributed. Our metric encompasses a combination of three different components. First, we focus on the *degree of specificity* of the relation cluster, as previous work has demonstrated that this can contribute to Information Extraction pipelines Bovi et al. (2015a). Second, we analyze *intrinsic features* of the relation pattern, such as frequency, length and fluency. Finally, we incorporate a *smoothing factor*, namely the proportion of the related typed cluster pattern in the cluster.

A cluster $\mathcal{R}_c$ may be decomposed into a set of typed cluster patterns τ_c (see Table 4.2). The intuition behind the specificity measure of a cluster is that clusters with one prominent τ_c are more specific, i.e. they are largely used for encoding one specific type of relations. One example of this would be *performed with*, which enforces a relation to include MusicalArtists on both the domain and range sides. Thus, we define $\mathcal{L}_c$ as the list of cardinalities (number of triples) of every typed cluster pattern $\tau_c \in \mathcal{R}_c$, being $\mathcal{L}_c = \{|\mathcal{R}_{\tau_c^1}|, ..., |\mathcal{R}_{\tau_c^n}|\}$. We define the specificity measure as the variance of $\mathcal{L}$, expressed as $s(\mathcal{R}_c) = var(\mathcal{L}_c)$.

Furthermore, we consider a *relation's fluency* metric, which is aimed at capturing its comprehensibility. Simply put, the more the sentence's original word order is preserved in the relation pattern, the more understandable it should be. This metric is introduced due to the fact that word order is lost after modelling text under a dependency grammar framework, and so we design a *penalty measure* over the number of jumps needed to reconstruct the original ordered word sequence. Let k be the number of tokens in the relation pattern, w_i the *ith* word in the pattern, and $h(w_i)$ a function that returns the correspondent word index in the original sentence, we put forward a fluency measure f defined as:

$$f(p) = \frac{\sum_{i=1}^{k} \alpha |h(w_i) - h(w_{i-1})|}{k} \tag{4.1}$$

where $\alpha = 2$ if $h(w_{i-1}) > h(w_i)$ and $\alpha = 1$ otherwise. Note that higher values of f means low fluency. For instance, for the relation pattern *is hit for* the score would be much higher than a mixed-up order relation pattern such as *joined because added were and hit*, which would have a very high f.

Finally, the global confidence measure for each relation $r \in R$ is expressed as follows:

$$score(r) = \left(s(\mathcal{R}_c) + \frac{|\mathcal{R}_p|}{|p| + 2^{f(p)}} \right) \times \frac{|\mathcal{R}_{\tau_c}|}{|\mathcal{R}_c|} \tag{4.2}$$

As an illustrative example of the measure, the score of a relation with the typed cluster pattern ⟨Song, *was released on*, RecordLabel⟩, will have a much higher score than a relation whose typed cluster pattern is ⟨Album, *was released on*, MusicalArtist⟩. This latter pattern is incorrect, probably due to a disambiguation error in the EL step. Relations like this show the type of errors which our proposed confidence score is expected to consider for pruning.

4.3 Experimental Setup

In this section, we describe our experimental setting. We refer first to the source raw corpus, and second to the resulting KBs as output of different branches of our approach.

4.3.1 Source dataset

Songfacts[19] is an online database that collects, stores and provides facts, stories and trivia about songs. These are collaboratively written by registered users, and reviewed by the website staff. It contains information about more than 30,000 songs from nearly 6,000 artists. This information may refer to what the song is about, who wrote it, who produced it, who collaborated with whom or who directed the video. These texts are rich sources of information not only for well-known music facts, but also for music-specific trivia, as in the following sample sentence (about David Bowie's *Space Oddity*): "Bowie wrote this song after seeing the 1968 Stanley Kubrick movie 2001: A Space Odyssey".

We crawled the Songfacts website in mid-January 2014. Then, for each song article, we performed a mapping between the song and its MusicBrainz song ID, using the MusicBrainz Search API. We successfully mapped 27,655 songs.

The RE pipeline was run over the 27,655 document Songfacts corpus, which amounts to 306,398 sentences. After the Semantic Processing step, we obtained 202,767 entity mentions (8,880 for *Albums*, 3,136 *Record Labels*, 74,908 *Songs*, 107,253 *Musical Artists*, 1,760 *Genre* labels, 3,467 for *Other Artist*, and 3,363 for *Film*). There were 48,122 sentences with at least two entities, and it is on this subset where we apply our RE pipeline.

4.3.2 Learned Knowledge Bases

Our aim is to assess to what extent each of the modules integrating our approach contributes to the quality of the resulting KB. After executing the whole pipeline, we generate two *learned* KBs (KBSF-ft and KBSF-th), two *baseline* KBs (KBSF-co and KBSF-raw), and a *competitor* KB (KBSF-rv).

The *learned* KBs are the result of applying the RE method to the Songfacts dataset under different conditions. KBSF-ft is derived from applying the RE pipeline entirely, and KBSF-th comes from a selection of all triples in KBSF-ft with a confidence score above a certain threshold. To determine the best threshold to prune KBSF-ft, we aimed at maximizing the number of triples and at the same time minimizing the number of relation patterns. Our intuition is that less patterns means a tidier KB. Therefore, we computed the percentage of triples and relation patterns from KBSF-ft that remain in a pruned KB, whose triples have a score greater than a certain threshold θ. We computed

[19] `http://www.songfacts.com`

these percentages for every θ value ranging from 0 to 1 in bins of 0.01 (see Figure 4.4). Our goal was to discover the θ value which maximizes the distance between the amount of triples and the amount of relation patterns in a pruned KB. After confirming a maximized difference with $\theta = 0.05$, we created KBSF-th, whose triples have a score greater than or equal to 0.05. In this pruned KB, we have 36.56% of KBSF-ft triples, with only 12.52% of its relation patterns.

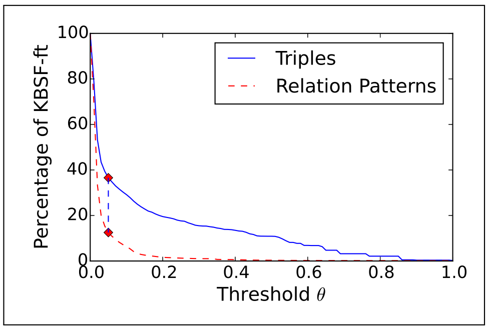


Figure 4.4: Percentage of triples and relation patterns from KBSF-ft that remain after pruning at different values of θ. Maximum distance at $\theta = 0.05$.

In addition, we created two baseline KBs for evaluation purposes. KBSF-co is a baseline which consists of simple entity co-occurrence. More specifically, if two entities are mentioned in the same sentence, an unlabelled triple that anchors them is added to the KB. In addition, KBSF-raw was created following the RE pipeline, but without applying the filtering process described in Section 4.2.5. Finally, KBSF-rv constitutes the competitor KB, and is built as follows: After running ReVerb over the Songfacts dataset, we search coinciding relations, at both domain and range positions, that include entity mentions identified in our disambiguation step. These relations are included in KBSF-rv. Statistics about the five KBs are reported in Table 4.3.

KB	Entities	Triples	Relation Patterns	Cluster Patterns
KBSF-ft	20,744	32,055	20,438	14,481
KBSF-th	10,977	11,720	2,484	828
KBSF-co	30,671	113,561	—	—
KBSF-raw	29,280	71,517	47,089	32,712
KBSF-rv	9,255	7,532	2,830	—

Table 4.3: Statistics of all the learned KBs

4.4 Experiments

4.4.1 Quality of Entity Linking

We mentioned in Section 4.2.3 the lacking of both music-specific EL tools as well as benchmarking datasets. For this reason, we performed a set of experiments to select the best-suited Entity Linking tool for the music domain, among some of the best known and reputed. Specifically, we perform evaluation experiments on DBpedia Spotlight, TagMe and Babelfy. Let us briefly describe each of them:

- **Babelfy** Moro et al. (2014c) A graph-based system for EL and Word Sense Disambiguation. It operates on the back of BabelNet, which serves as a reference sense inventory.
- **Tagme** Ferragina & Scaiella (2010) An Entity Linking automatic annotator which matches terms with Wikipedia hyperlink texts and disambiguates them using both the in-link graph and the page datasets. It incorporates a context-aware pruning step.
- **DBpedia Spotlight** Mendes et al. (2011) Automatically identifies entity mentions in free text, linking each match with its corresponding DBpedia URI.

As of now, most EL systems *speak their own language*, partially due to the fact that they perform entity disambiguation with different KBs as reference. Since their output is heterogeneous in format, performing a comparison between them is not straightforward. In order to evaluate the aforementioned EL systems, we used ELVIS[20] Oramas et al. (2016), an EL integration tool which provides a common output for different EL system.

In addition, we created a dataset of annotated musical entities and applied both quantitative and qualitative evaluations in order to verify which system performs better with musical entities, and is more suitable for our task.

Evaluation Data

We created an *ad-hoc* gold standard dataset to evaluate the different EL systems, with the Songfacts dataset (Section 4.3.1) as our testbed. In this corpus, each document univocously refers to one single song. In addition, we have information about artist and song names at our disposal. We used this information to obtain the MusicBrainz ID for songs and artists. In MusicBrainz, artist and song items sometimes have information about their equivalent Wikipedia page. We leveraged this information, when available, to obtain their corresponding DBpedia URIs. Finally, we obtained a mapping with DBpedia of 7,691 songs and 3,670 artists. From the DBpedia resources of each song, we gathered their corresponding album name and URI, if available, obtaining information of about 2,092 albums. Then, for every document, we looked for exact string matches of the reported song, and its related album and artist names. Every detected entity is thus annotated with its DBpedia

[20] https://github.com/sergiooramas/elvis

URI. At the end of this process, the newly created gold standard dataset contains 6,052 documents where 17,583 sentences are annotated with the following entities: 5,981 Song, 12,137 Artist and 1,722 Album entities. As mentioned in Section 4.2.3, there are typical cases of ambiguity in musical entities where songs, artists and albums can potentially share the same name. Therefore, we manually corrected the entities detected in 212 documents where this kind of ambiguity was present.

Entity Linking Evaluation

	Album		Artist		Song		Macro Average		
	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	F-measure
Babelfy	0.93	0.28	0.98	0.55	0.96	0.31	0.96	0.38	0.54
Tagme	0.75	0.69	0.97	0.77	0.65	0.71	0.79	0.72	**0.76**
Spotlight	0.80	0.52	0.94	0.83	0.59	0.42	0.78	0.59	0.67

Table 4.4: Precision and recall of the Entity Linking Systems considered

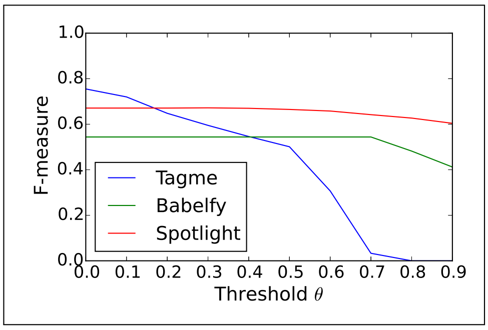


Figure 4.5: F-measure of the Entity Linking systems at different confidence thresholds

The three EL systems under review provide their own confidence measure. Hence, we evaluated their output filtering out the entities with a confidence measure below to a certain threshold θ. We run the evaluation for different values of θ, ranging from 0 to 0.9 in bins of 0.1. After evaluating on the gold dataset, the best results in terms of F-measure were obtained by all the systems at $\theta = 0$ (see Figure 4.5), which means that there is no need to apply any filtering process based on the EL system own confidence score. Detailed results on the run of every system at $\theta = 0$ are shown in Table 4.4. We used macro-average Precision and Recall measures, i.e. we averaged their values from the three sets of entities.

We may conclude from these results that Babelfy is the system with highest Precision on musical entities. However, its recall is lower than the other systems under consideration, and specifically with respect to Tagme, which in turn, shows much lower precision. DBpedia Spotlight, on the other hand, achieves a similar precision score as Tagme, but with a slightly lower recall.

This evaluation experiment is only focused on measuring the precision in the annotation of entities present in the gold standard. However, since all possible entities in a document may be not annotated, we also report on specific types of false positives which emerged during a qualitative inspection of classification results. For example, a frequent error that is not being evaluated concerns cases in which a text span not annotated in the ground truth is identified incorrectly as an entity by any system. Therefore, to complement the evaluation, we listed the most frequently identified entities by each system (see Table 4.5). As we can see, Babelfy and Tagme are misidentifying common words as entities very frequently, whereas DBpedia Spotlight is not doing so. These errors may propagate to the rest of the RE pipeline, penalizing the accuracy of the final KB. Although a filtering process could be applied to filter out misidentified entities by computing their tf-idf score in each document, we opted for using DBpedia Spotlight, as it has shown pretty good performance, its output does not require any further processing, and it is released as open source, which means that there are no limitations on the number of queries.

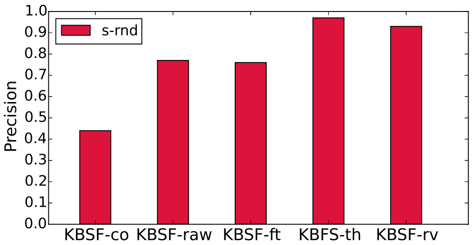
System	Song	Album	Artist
Babelfy	**Carey**	**Debut**	John_Lennon
	Stephen	**Song_For**	Eminem
	Rap_Song	**Sort_Of**	Paul_McCartney
	Singing_This_Song	**First_Song**	Bob_Dylan
	A_Day_in_the_Life	**Debut_Album**	Drake
Tagme	**The_Word**	**Up!**	John_Lennon
	The_End	**When_We_On**	**The_Notorious_B.I.G.**
	If	Up	**Do**
	Once	**Together**	Paul_McCartney
	For_You	**By_the_Way**	**Neil_Young**
Spotlight	Sexy_Sadie	The_Wall	Madonna
	Helter_Skelter	Let_It_Be	Eminem
	Cleveland_Rocks	Born_This_Way	Rihanna
	Stairway_to_Heaven	Thriller	John_Lennon
	Minnie_the_Moocher	Robyn	Britney_Spears

Table 4.5: Top-5 most frequent entities by type and tool. Disambiguation errors appear in bold.

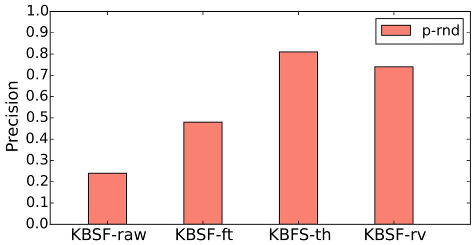
4.4.2 Quality of Relations

RE evaluation is not trivial, as semantic relations between entities may vary in terms of correctness over time. Also, correct relations may be linguistically flawed, i.e. not fluent. Previous approaches assessed automatically extracted relations in terms of correctness according to human judgement Fader et al. (2011); Mausam et al. (2012). Additionally, a finer grained analysis is carried out in Banko et al. (2007a), adding a prior step in which relations are judged as being *concrete* or *abstract*.

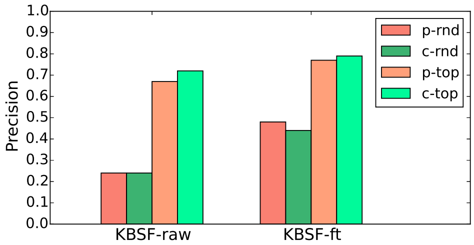
In this paper, we made use of extensive human input and asked two experts in Computational Linguistics to evaluate the *top 100* scoring relations as yielded by our weighting policy (Section 4.2.7), as well as a random sample of 100 relations. This was done for all the KBs produced by our pipeline and



(a) In sentence



(b) Relation patterns



(c) Patterns and clusters

Figure 4.6: Precision of relations at sentence (s), relation pattern (p) and cluster pattern (c) levels in top (*top*) and random (*rnd*) samples of relations

for KBSF-rv. Cohen's kappa coefficient ranged from 0.60 to 0.81, which is generally considered as *substantial* agreement.

In Figures 4.6a and 4.6b, where we compare random samples from each KB, we observe a gradual improvement of the quality of relations as the different modules of our implementation are incorporated. The difference between these figures is that in the former, a relation is deemed correct if it has extracted a relation *expressed in the original sentence*, whereas the latter figure reports numbers on whether the extracted relation pattern was correct, i.e. if it *meant* the same as it was intended in the source sentence. We may infer from these results that co-occurrence between entities does not guarantee an explicit relation, whereas the presence of a path between two entities over a sentence dependency tree, without any other entity mention in between, generally suggests a monsemous and unambiguous relation.

It is remarkable how well ReVerb performs (Figure 4.6b), only being surpassed by the KB resulting from the complete implementation described in this paper. We note that the good results of the ReVerb extractor are also due to the semantic processing of our system, which is forcing ReVerb to select good candidates as relation arguments. Recall that the difference between KBSF-ft and KBSC-th is the inclusion of the *scoring* module, and the increase in Precision confirms that incorporating *statistical evidence contributes to better relations.*

This is further confirmed in the results showcased in Figure 4.6c, where we provide a comparison between top 100 relations according to our ranking policy against a random sample. Note that *in all* KBS*, highly scoring relations are more often marked as correct*, which constitutes additional support for the contribution of the scoring module. Together with the quality of the relation pattern, this figure shows the quality of the cluster pattern associated with the evaluated relations. We observe that cluster patterns inferred in our clustering module have similar quality than relation patterns in the random sample, and slightly better in the top 100 sample. This result implies that the scoring module is rewarding good clusters.

4.4.3 Coverage of the Extracted Knowledge Base

With this experiment, we aim to compare the coverage of music relations in our KBs with respect to other resources with human intervention, such as DBpedia, MusicBrainz, and with fully automatic resources. For the latter, we considered DefIE as our closest competitor due to several methodological similarities (dependency parsing, EL and RE over shortest paths).

We selected all triples in KBSF-th whose domain and range entities could be mapped to both DBpedia and MusicBrainz. As our extracted KB has only MusicBrainz ID of entities of types MusicalArtist and Song, the set of triples to evaluate is restricted to relations between them. Since entities in DefIE are disambiguated against BabelNet ids, we mapped all DBpedia uris to their corresponding BabelNet id, which yielded a subset of 3,633 triples. From here, we selected all possible domain-range entity pairs, and retrieved from the other KBs all triples with the same pairs, and counted them. The procedure to do so on DBpedia was via SPARQL queries. We discarded triples with predicate *wikiPageWikiLink*, as this predicate means an unlabeled relation. However, the mapping with MusicBrainz was not trivial. MusicBrainz is not a KB of triples, but a relational database. Entities are stored in tables, and relations between entities are represented in a set of tables of relations, having one table for each possible relation. The entities in the studied set of triples were only of type MusicalArtist and Song. However, an entity of type Song in KBSF-th can be related to either a Recording or a Work entity in MusicBrainz (see Section 4.2.3). Therefore, for the analysis of relations involving a Song entity, we obtained the equivalent Recording and Work MusicBrainz entities, and looked up relations where any of them where present.

Mapping results are shown in Table 4.6. Let us highlight the fact that most semantic relations encoded in KBSF-th are novel, as they were not found in any of the other resources we compared against. In the overlapping cases, most of the times the relation labels were semantically equivalent, and often the relation label of KBSF-th triples was more specific than the ones retrieved from other KBs (e.g. *frontman* and *member of*)

	KBSF-th	MusicBrainz	DBpedia	DefIE
Relation instances	3,633	1,535	1,240	456

Table 4.6: Number of triples with labeled relations in the different KBs for the same set of domain-range entity pairs

4.4.4 Interpretation of Music Recommendations

The main aim of this experiment is to evaluate the suitability of KBSF-th to explain relations between songs, and study their impact on user's experience in music recommendation. Since our aim is not to measure the performance of a recommender system, we implemented a baseline recommender approach. Recommendations are based on the concept of song similarity, which exploits the graph-based structure of our KB, following Oramas et al. (2015c).

We designed the experiment as an online survey, where the participant is first asked to select 5 songs from different artists of his/her choice. From each selected song, the system randomly selects 3 recommendations among the list of its top-10 most similar songs. One of them is shown together with an explanation in natural language (the source text), another with an explanation based on relation patterns, and finally the third one appears without explanation. Participants can listen to all songs with an embedded player. After listening to the recommendation and reading the explanation attached to it, participants were asked to rate each recommendation from 1 to 5 (1 being worst), and to mention whether they were familiar or not with the recommended songs (see Figure 4.7).

The experiment involved 35 participants, 28 males and 7 females, ranging from 26 to 38 years old and with different musical background and listening habits. Most of the participants said that they had previous experience with recommendation systems. A total of 525 answers (corresponding to individual song recommendations) were collected. In 38% of the cases, the user was familiar with the recommended songs.

The average rating of recommendations with natural language explanations is slightly higher (3.20±1.29) than recommendations without explanations (3.08±1.35), or with explanations based on relation labels (3.04±1.34). In addition, for musically educated individuals, recommendations of unfamiliar songs, whether accompanied with or without explanations, have similar average rating (2.87 and 2.95 respectively). However, for untrained users, recommendations with explanations have a remarkable higher average rating (2.93) than without them (2.36). Thus, we can infer that the introduction of explanations in recommender systems improves the user experience of musically untrained subjects when discovering songs.

We also asked the subjects to select among a set of adjectives those that better described the recommendation experience. The general trend was to rate positively the experiment. Most users rated the experience as *enjoyable* (40%), followed by *useful* (31%) and enriching (29%). Negativity was much lower in general, with *confusing* being the most voted (17%), followed by *complicated* and *too geeky* (8% in both cases). This suggests that the introduction of explanations generated from our MKB in the recommendations was in general a satisfactory experience to users.

Figure 4.7: User interface for the music recommendation experiment.

4.5 Conclusions and Future Work

We have presented an NLP pipeline that learns a Knowledge Base in the music domain taking raw text collections as input. It combines methods easily applicable to a general purpose application with domain-specific heuristics which are designed to exploit particularities of the domain.

The result of applying our approach over a dataset of stories about songs is a new Music Knowledge Base, which encodes semantic relations among musical entities. Our method relies on the syntactic structure (defined via dependency parsing) of sentences and the use and adaptation of music-specific heuristics for both EL and RE. In addition, we include modules for semantic clustering and pattern scoring, aimed at the efficient removal of noisy relations. Our modular evaluation shows that our RE module is able to capture a highly precise and compact set of weighted triples, and demonstrates the positive impact of the novel scoring metric we introduced. Moreover, we have shown that a high percentage of the knowledge encoded in our MKB is not present in other KBs, both general and domain-specific. Finally, regarding extrinsic evaluation, the experiment on recommendation interpretation confirms that explanations based on the learned KB are positively regarded by the users.

We have identified several promising avenues for future work. For instance, we would like to extend our experiments to other music datasets of varied registers (e.g. social networks, magazines, encyclopedias), in order to fully understand the core differences between this domain and standard language. This should give an approximate idea of whether we need specific tools in certain specific NLP tasks. For instance, it seems reasonable to envision a music EL tool that is able to cope better with certain particularities of the domain. In addition, the development of new methodologies in Music Information Retrieval that exploit MKBs is still an open area of research.

CHAPTER 5

Artist Similarity

5.1 Introduction

Artist biographies are a big source of musical context information and have been previously used for computing artist similarity. However, only shallow approaches have been applied by computing word co–occurrences and thus the semantics implicit in text have been barely exploited. To do so, semantic technologies, and more specifically Entity Linking tools may play a key role to annotate unstructured texts. These are able to identify named entities in text and disambiguate them with their corresponding entry in a knowledge base (e.g. Wikipedia, DBpedia or BabelNet).

This paper describes a method for computing semantic similarity at document-level, and presents evaluation results in the task of artist similarity. The cornerstone of this work is the intuition that semantifying and formalizing relations between entity mentions in documents (both at in-document and cross-document levels) can represent the relatedness of two documents. Specifically, in the task of artist similarity, this derives in a measure to quantify the degree of relatedness between two artists by looking at their biographies.

Our experiments start with a preprocessing step which involve Entity Linking over artist biographical texts. Then, a knowledge representation is derived from the detected entities in the form of a semantic graph or a mapping to a vector-space model. Finally, different similarity measures are applied to a benchmarking dataset. The evaluation results indicate that some approaches presented in this paper clearly outperform a baseline based on shallow word co-occurrence metrics. Source code and datasets are available online[21].

The remainder of this article is structured as follows: Section **??** reviews prominent work in the fields and topic relevant to this paper; Section 5.2 details the different modules that integrate our approach; Section 5.3 describes the settings in which experiments were carried out together with the evaluation

[21] http://mtg.upf.edu/downloads/datasets/semantic-similarity

metrics used; Section 5.4 presents the evaluation results and discusses the performance of our method; and finally Section 12 summarizes the main topics covered in this article and suggests potential avenues for future work.

5.2 Methodology

The method proposed in this paper can be divided in three main steps, as depicted in Fig 5.1. The first step performs entity linking, that is the detection of mentions to named entities in the text and their linking to an external knowledge base. The second step derives a semantically motivated knowledge representation from the named entity mentions. This can be achieved by exploiting natural language text as anchor between entities, or by incorporating semantic information from an external knowledge base. In the latter case, a document is represented either as a semantic graph or as a set of vectors projected on a vector space, which allows the use of well known vector similarity metrics. Finally, the third step computes semantic similarity between documents (artist biographies in our case). This step can take into consideration semantic similarity among entity mentions in document pairs, or only the structure and content of the semantic graph.

The following sections provide a more detailed description of each one of these steps, along with all the approaches we have considered in each step.

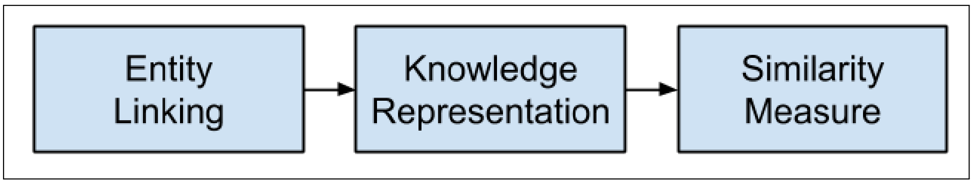


Figure 5.1: Workflow of the proposed method.

5.2.1 Entity Linking

Entity linking is the task to associate, for a given candidate textual fragment, the most suitable entry in a reference Knowledge Base (KB) Moro et al. (2014a). It encompasses similar subtasks such as Named Entity Disambiguation Bunescu & Pasca (2006), which is precisely linking mentions to entities to a KB, or Wikification Mihalcea & Csomai (2007), specifically using Wikipedia as KB.

We considered several state-of-the-art entity linking tools, including Babelfy Moro et al. (2014a), TagMe Ferragina & Scaiella (2010), Agdistis Usbeck et al. (2014a) and DBPedia Spotlight Mendes et al. (2011). However we opted to use the first one for consistency purposes, as in a later step we exploit *SensEmbed* Iacobacci et al. (2015), a vector space representation of concepts based on BabelNet Navigli & Ponzetto (2010). Moreover, the use of a single tool across approaches guarantees that the evaluation will only reflect the appropriateness of each one of them, and in case of error propagation all the approaches will be affected the same.

Babelfy Moro et al. (2014a) is a state-of-the-art system for entity linking and word sense disambiguation based on non-strict identification of candidate meanings (i.e. not necessarily exact string matching), together with a graph based algorithm that traverses the BabelNet graph and selects the most appropriate semantic interpretation for each candidate.

5.2.2 Knowledge representation

Relation graph

Relation extraction has been defined as the process of identifying and annotating relevant semantic relations between entities in text Jiang & Zhai (2007). In order to exploit the semantic relations between entities present in artist biographies, we applied the method defined in Oramas et al. (2015b) for relation extraction in the music domain. The method basically consists of three steps. First, entities are identified in the text by applying entity linking. Second, relations between pairs of entities occurring in the same sentence are identified and filtered by analyzing the structure of the sentence, which is obtained by running a syntactic parser based on the formalism of dependency grammar Bohnet (2010). Finally, the identified entities and relations are modeled as a knowledge graph. This kind of extracted knowledge graphs may be useful for music recommendation Sordo et al. (2015), as recommendations can be conveyed to users by means of natural language. We apply this methodology to the problem of artist similarity, by creating a graph that connects the entities detected in every artist biography. We call this approach RG (relation graph). Figure 5.2 shows the output of this process for a single sentence.

Semantically enriched graph

A second approach is proposed using the same set of linked entities. However, instead of exploiting natural language text, we use semantic information from the referenced knowledge base to enrich the semantics of the linked entities. We follow a semantic enrichment process similar to the one described in Ostuni et al. (2015). We use semantic information coming from DBpedia[22]. DBpedia resources are generally classified using the DBpedia Ontology, which is a shallow, cross-domain ontology based on the most common infoboxes of Wikipedia. DBpedia resources are categorized using this ontology among others (e.g. Yago, schema.org) through the `rdfs:type` property. In addition, each Wikipedia page may be associated with a set of Wikipedia categories,

[22] `http://dbpedia.org`

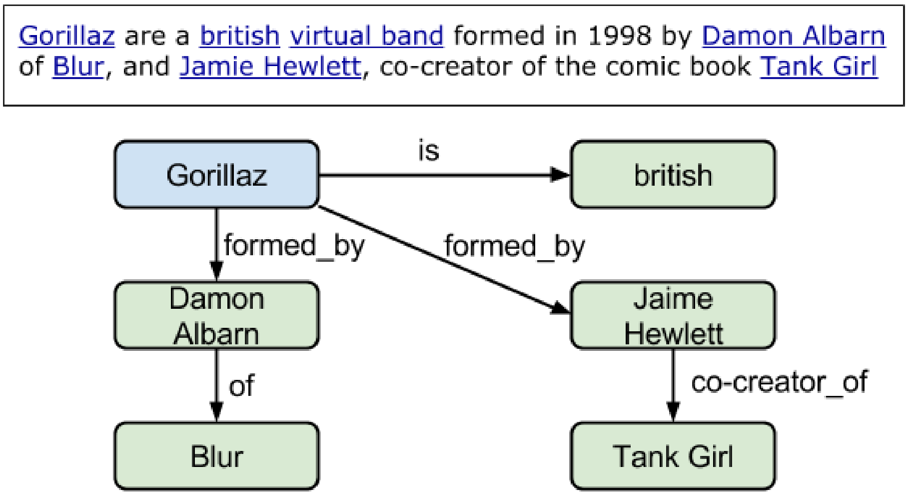


Figure 5.2: Relation graph of a single sentence

which link articles under a common topic. DBpedia resources are related to Wikipedia categories through the property `dcterms:subject`.

We take advantage of these two properties to build our semantically enriched graph. We consider three types of nodes for this graph: 1) artist entities obtained by matching the artist names to their corresponding DBPedia entry; 2) named entities detected by the entity linking step; and 3) Wikipedia categories associated to all the previous entities. Edges are then added between artist entities and the named entities detected in their biographies, and between entities and their corresponding Wikipedia categories. For the construction of the graph, we can select all the detected named entities, or we can filter them out according to the information related to their `rdfs:type` property. A set of six types was selected, including *artist*, *band*, *work*, *album*, *musicgenre*, and *person*, which we consider more appropriate to semantically define a musical artist.

From the previous description, we define five variants of this approach. The first variant, which we call AEC (Artists-Entities-Categories), considers all 3 types of nodes along with their relations (as depicted in Figure 5.3). The second variant, named AE (Artists-Entities) ignores the categories of the entities. The third and fourth variant, named AEC-FT and AE-FT, are similar to the first and second variant, respectively, except that the named entities are filtered using the above mentioned list of 6 entity types. Finally, the fifth variant, EC, ignores the artist entities of node type 1.

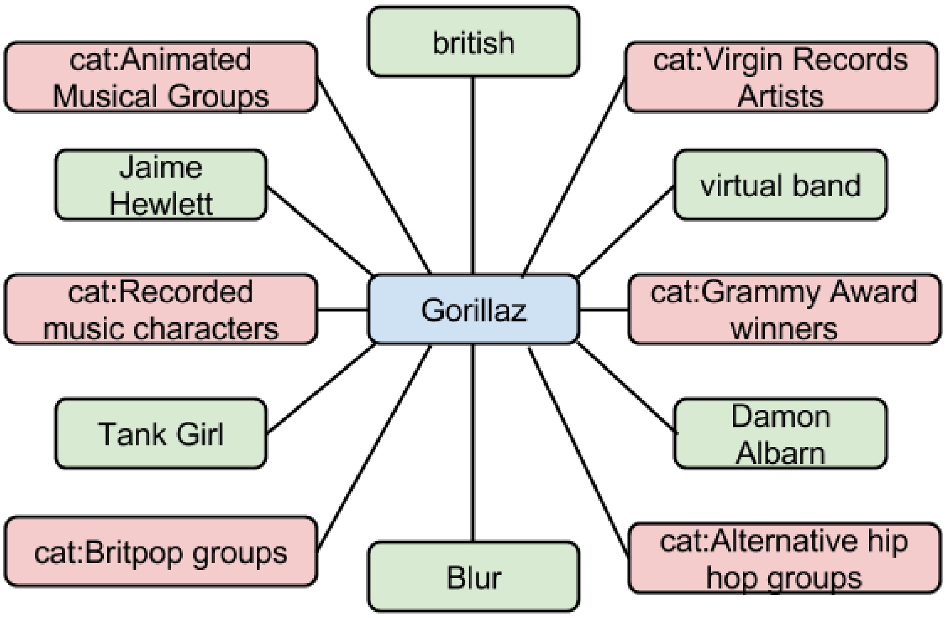


Figure 5.3: Semantically enriched subgraph of the same sentence from Figure 5.2, variant AEC with h=1

Sense embeddings

The semantic representation used in this approach is based on SensEmbed Iacobacci et al. (2015). SensEmbed is a vector space semantic representation of words similar to word2vec Mikolov et al. (2013), where each vector represents a BabelNet synset and its lexicalization. Let A be the set of artist biographies in our dataset. Each artist biography $a \in A$ is converted to a set of disambiguated concepts Bfy_a after running Babelfy over it.

5.2.3 Similarity approaches

SimRank

SimRank is a similarity measure based on an simple graph-theoretic model Jeh & Widom (2002). The intuition is that two nodes are similar if they are referenced by similar nodes. In particular we use the definition of bipartite SimRank Jeh & Widom (2002). We build a bipartite graph with named entities and their corresponding Wikipedia categories (the EC variant from Section 5.2.2). The similarity between two named entities (say p and q) is computed with the following recursive equation:

$$s(p,q) = \frac{C}{|O(p)||O(q)|} \sum_{i=1}^{|O(p)|} \sum_{j=1}^{|O(q)|} s(O_i(p), O_j(q)) \tag{5.1}$$

where O denotes the out-neighboring nodes of a given node and C is a constant between 0 and 1. For $p = q$, $s(p, q)$ is automatically set up to 1. Once the similarity between all pairs of entities is obtained, we proceed to calculate the similarity between pairs of artists (say a and b) by aggregating the similarities between the named entities identified in their biographies, as shown in the following formula:

$$sim(a,b) = Q(a,b)\frac{1}{N}\sum_{e_a \in a}\sum_{e_b \in b} s(e_a, e_b) \quad \text{if } s(e_a, e_b) \geq 0.1 \tag{5.2}$$

where s denotes the SimRank of entities e_a and e_b and N is the number of (e_a, e_b) pairs with $s(e_a, e_b) \geq 0.1$. This is done to filter out less similar pairs. Finally, $Q(a, b)$ is a normalizing factor that accounts for the pairs of artists with more similar entity pairs than others.

Maximal common subgraph

Maximal common subgraph (MCS) is a common distance measure on graphs. It is based on the maximal common subgraph of two graphs. MCS is a symmetric distance metric, thus $d(A, B) = d(B, A)$. It takes structure as well as content into account. According to Bunke & Shearer (1998), the distance between two non empty graphs G_1 and G_2 is defined as

$$d(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{max(|G_1|, |G_2|)} \tag{5.3}$$

It can also be seen as a similarity measure s, assuming that $s = 1-d$, as applied in Lux & Granitzer (2005). To compute this similarity measure we need to have a graph for each artist. This can be achieved by finding subgraphs in the graph approaches defined in Section 5.2.2. A subgraph will include an artist entity node and its neighboring nodes. Furthermore, we apply the notion of h-hop item neighborhood graph defined in Ostuni et al. (2014) to a semantic graph. Let $G = (E, P)$ be an undirected graph where E represent the nodes (entities), and P the set of edges with $P \subseteq E \times E$. For an artist item a in G, its h-hop neighborhood subgraph $G^h(a) = (E^h(a), P^h(a))$ is the subgraph of G formed by the set of entities that are reachable from a in at most h hops, according to the shortest path. Following this approach, we obtain an h-hop item neighborhood graph for each artist node of the semantic graph. Then, maximal common subgraph is computed between each pair of h-hop item neighborhood graphs. For each artist, the list of all similar artists ordered from the most similar to the less one is finally obtained.

Cumulative cosine similarity

For each pair of concepts $c \in \mathrm{Bfy}_a$ and $c' \in \mathrm{Bfy}'_a$ (as defined in Section 5.2.2), we are interested in obtaining the similarity of their closest senses. This is achieved by first deriving the set of associated vectors V_c and $V'_{c'}$ for each pair of concepts c, c', and then optimizing

$$\max_{v_c \in V_c, v'_{c'} \in V'_{c'}} \left(\frac{v_c \times v'_{c'}}{||v_c|| \, ||v'_{c'}||} \right) \tag{5.4}$$

i.e. computing cosine similarity between all possible senses (each sense represented as a vector) in an all-against-all fashion and keeping the highest scoring similarity score for each pair. Finally, the semantic similarity between two artist biographies is simply the average among all the cosine similarities between each concept pair.

5.3 Experimental Setup

To evaluate the accuracy of the proposed approaches we designed an experimental evaluation over two datasets. The first dataset contains 2,336 artists and it is evaluated using the list of similar artists provided by the Last.fm API as a ground truth. The second dataset contains 188 artists, and it is evaluated against user similarity judgements from the MIREX Audio Music Similarity and Retrieval task. Apart from the defined approaches, a pure text-based approach for document similarity is added to act as a reference for the obtained results.

5.3.1 Datasets

Last.fm dataset

A dataset of 2,336 artist biographies was gathered from Last.fm. The artists in this dataset share a set of restrictions. Their biography has at least 500 characters and is written in English. All of the artists have a correspondent Wikipedia page, and we have been able to mapped it automatically, obtaining the DBpedia URI of every artist. For every artist, we queried the getSimilar method of the Last.fm API and obtained an ordered list of similar artists. Every artist in the dataset fulfills the requirement of having at least 10 similar artists within the dataset. We used these lists of similar artists as the ground truth for our evaluation.

MIREX dataset

To build this dataset, the gathered artists from Last.fm were mapped to the MIREX Audio Music Similarity task dataset. The AMS dataset (7,000 songs from 602 unique artists) contains human judgments of song similarity. According to Schedl et al. (2013), the similarity between two artists can be roughly estimated as the average similarity between their songs. We used the same approach in Schedl et al. (2013), that is, two artists were considered similar if the average similarity score between their songs was at least 25 (on a fine scale between 0 and 100).

After the mapping, we obtained an overlap of 268 artists. As we want to evaluate Top-10 similarity, every artist in the ground truth dataset should have information of at least 10 similar artists. However, not every artist in the MIREX evaluation dataset fulfills this requirement. Therefore, after removing the artists with less than 10 similars, we obtained a final dataset of 188 artists, and used it for the evaluation.

5.3.2 Baseline

In order to assess the goodness of our approaches, we need to define a baseline approach with which to compare to. The baseline used in this paper is a classic vector-based model approach used in many Information Retrieval systems. A text document is represented as a vector of word frequencies (after removing English stopwords and words with less than 2 characters), and a matrix is formed by aggregating all the vectors. The word frequencies in the matrix are then re-weighted using TF-IDF, and finally latent semantic analysis (LSA) Deerwester et al. (1990) is used to produce a vector of concepts for each document. The similarity between two documents can be obtained by using a cosine similarity over their corresponding vectors.

	Precision@N		nDCG@N	
Approach variants	N=5	N=10	N=5	N=10
LSA	0.100	0.169	0.496	0.526
RG MCS 1-hop	0.059	0.087	0.465	0.476
RG MCS 2-hop	0.056	0.101	0.433	0.468
AE MCS	0.106	0.178	0.503	0.517
AE-FT MCS	0.123	0.183	0.552	0.562
AEC MCS 1-hop	0.120	0.209	0.573	0.562
AEC MCS 2-hop	0.086	0.160	0.550	0.539
AEC-FT MCS 1-hop	**0.140**	**0.218**	**0.588**	**0.578**
AEC-FT MCS 2-hop	0.100	0.160	0.527	0.534
EC SimRank	0.097	0.171	0.509	0.534
SE Cosine	0.095	0.163	0.454	0.484

Table 5.1: Precision and normalized discounted cumulative gain for Top-N artist similarity using the MIREX dataset (N={5, 10})

	Precision@N		nDCG@N	
Approach variants	N=5	N=10	N=5	N=10
LSA	0.090	0.088	0.233	0.269
RG MCS 1-hop	0.055	0.083	0.126	0.149
AE MCS	0.124	0.200	0.184	0.216
AE-FT MCS	0.136	0.201	0.224	0.260
AEC MCS 1-hop	0.152	0.224	0.277	0.297
AEC-FT MCS 1-hop	**0.160**	**0.242**	**0.288**	**0.317**

Table 5.2: Precision and normalized discounted cumulative gain for Top-N artist similarity using the Last.fm dataset (N={5, 10})

5.3.3 Evaluated approaches

From all possible combinations of knowledge representations, similarity measures and parameters, we selected a set of 10 different approach variants. The prefixes AEC, RG and AE refer to the graph representations (see Sections 5.2.2 and 5.2.2). SE refers to the sense embeddings approach, and LSA to the latent semantic analysis baseline approach. When these prefixes are followed by FT, it means that the entities in the graph have been filtered by type. The second term in the name refers to the similarity measure. MCS refers to maximal common subgraph, and SimRank and Cosine to SimRank and cumulative cosine similarity measures. MCS approaches are further followed by a number indicating the number of h-hops of the neighborhood subgraph.

5.3.4 Evaluation measures

To measure the accuracy of the artist similarity we adopt two standard performance metrics such as Precision@N, and nDCG@N (normalized discounted cumulative gain). Precision@N is computed as the number of relevant items (i.e., true positives) among the top-N items divided by N, when compared to a ground truth. Precision considers only the relevance of items, whilst nDCG takes into account both relevance and rank position. Denoting with s_{ak} the relevance of the item in position k in the Top-N list for the artist a, then nDCG@N for a can be defined as:

$$\text{nDCG@N} = \frac{1}{\text{IDCG@N}} \sum_{k=1}^{N} \frac{2^{s_{ak}} - 1}{\log_2(1+k)} \tag{5.5}$$

where IDCG@N indicates the score obtained by an ideal or perfect Top-N ranking and acts as a normalization factor. We run our experiments for $N = 5$ and $N = 10$.

	Genres							
Approach variants	Blues	Country	Edance	Jazz	Metal	Rap	Rocknroll	Overall
Ground Truth	5.78	5.46	6.88	7.04	7.10	8.68	5.17	6.53
LSA	4.43	4.12	3.80	4.64	5.79	5.08	4.74	4.69
RG MCS 1-hop	2.63	3.50	1.50	2.95	4.00	2.54	1.70	2.68
RG MCS 2-hop	4.14	4.92	1.69	2.80	3.78	3.06	2.77	3.27
AE MCS	5.52	5.15	4.36	7.00	4.34	5.36	4.46	5.11
AE-FT MCS	5.43	6.12	4.16	6.20	6.32	5.36	3.77	5.26
AEC MCS 1-hop	**7.22**	5.92	5.24	7.12	5.48	6.92	4.86	6.02
AEC MCS 2-hop	4.22	3.69	4.56	6.20	4.55	4.64	4.09	4.54
AEC-FT MCS 1-hop	6.91	**6.80**	**6.04**	**7.60**	**6.79**	**7.12**	**5.37**	**6.59**
AEC-FT MCS 2-hop	4.09	4.36	5.56	6.72	4.39	4.16	3.77	4.67
EC SimRank	6.74	5.38	3.16	6.40	4.59	4.44	3.80	4.85
SE Cosine	3.39	5.50	5.32	5.16	4.31	5.36	4.31	4.75

Table 5.3: Average genre distribution of the top-10 similar artists using the MIREX dataset. In other words, on average, how many of the top-10 similar artists are from the same genre as the query artist. LSA stands for Latent Semantic Analysis, RG for Relation Graph, SE for Sense Embeddings, and AE, AEC and EC represent the semantically enriched graphs with Artists-Entities, Artist-Entities-Categories, and Entities-Categories nodes, respectively. As for the similarity approaches, MCS stands for Maximum Common Subgraph.

5.4 Results and discussion

We evaluated all the approach variants described in Section 5.3.3 on the MIREX dataset, but only a subset of them on the Last.fm dataset, due to the high computational cost of some of the approaches.

Table 5.1 shows the Precision@N and nDCG@N results of the evaluated approaches using the MIREX dataset, while Table 5.2 shows the same results for the Last.fm dataset. We obtained very similar results in both datasets. The approach that gets best performance for every metric, dataset and value of N is the combination of the Artists-Entities-Categories graph filtered by types, with the maximal common subgraph similarity measure using a value of $h = 1$ for obtaining the h-hop item neighborhood graphs.

Furthermore, given that the MIREX AMS dataset also provides genre data, we analyzed the distribution of genres in the top-10 similar artists for each artist, and averaged them by genres. The idea is that an artist's most similar artists should be from the same genre as the seed artist. Table 5.3 presents the results. Again, the best results are obtained with the approach that combines the Artists-Entities-Categories graph filtered by types, with the maximal common subgraph similarity measure using a value of $h = 1$ for the h-hop item neighborhood graphs.

We extract some insights from these results. First, semantic approaches are able to improve pure text-based approaches. Second, using knowledge from an external knowledge base provides better results than exploiting the relations inside the text. Third, using a similarity measure that exploits the structure and content of a graph, such as maximal common subgraph, overcomes other similarity measures based on semantic similarity among entity mentions in document pairs.

5.5 Conclusion

In this paper we presented a methodology that exploits semantic technologies for computing artist similarity, which can be divided in three main steps: First, named entity mentions are identified in the text and linked to a knowledge base. Then, these entity mentions are used to construct a semantically motivated knowledge representation. Finally a similarity function is defined on top of the knowledge representation to compute the similarity between artists. For each one of these steps we explored several approaches, and evaluated them against a small dataset of 188 artist biographies, and a larger dataset of 2,336 artists, both obtained from Last.fm.

Results showed that a combination of the Artists-Entity-Categories graph filtered by types, and a maximal common subgraph similarity measure using a value of $h = 1$ for obtaining the h-hop item neighborhood graphs, clearly outperforms a baseline approach that exploits word co–occurrences and latent factors. In the light of these results, the following conclusions can be drawn: First, semantic approaches may outperform pure text-based approaches. Second, we observe that knowledge leveraged from external ontologies may improve the accuracy of the similarity measure. Third, reducing noise by filtering linked entities by type is a rewarding step that contributes to an improved performance. Finally, we show that similarity measures that take into consideration the structure and content of a graph representation may achieve much higher performance.

There are still many avenues for future work. We would like to compare our semantic-based approach with acoustic and collaborative filtering approaches. In addition, the use of text sources different from artist biographies can be studied. Finally, in order to improve the results obtained by our semantic approach, different state-of-the-art entity linking tools can be applied, or a specific entity linking tool for the music domain could be created for this purpose.

CHAPTER 6

Music Recommendation with Knowledge Graphs

6.1 Introduction

6.2 Knowledge enrichment via entity linking

In order to add more semantics to the description of musical items, we exploit contextual information, i.e., tags and text descriptions, and then use this information to create a knowledge graph. Several approaches have been developed to enrich tags with semantics Garcia-Silva et al. (2012). We follow an ontology-based approach, enriching both tags and keywords extracted from textual descriptions by associating them with relevant entities defined in online semantic datasets. The first step in this direction is to link and disambiguate tags and keywords to Linked Data resources. For this purpose we adopted Babelfy, a state of the art tool for Entity Linking and Word Sense Disambiguation Moro et al. (2014b). Babelfy maps words from a given text to entities in the BabelNet[23] knowledge base. BabelNet is a multilingual encyclopedic dictionary that mixes knowledge from WordNet and Wikipedia. Thus, for every mapped and disambiguated text fragment, Babelfy returns the related WordNet *synset*, and/or the related Wikipedia page (and its equivalent DBpedia resource).

To build our semantically enriched graph, the entity linking tool is firstly run on both tags and keywords of every item. Identified named entities are linked to DBpedia resources, whilst disambiguated words are linked to WordNet *synsets*. Every musical item is added to the graph, and connected with the words taken from its context that are identified as entities by Babelfy. Words are in turn connected with their corresponding URIs, whether they are a DBpedia resource or a WordNet *synset*. Subsequently, we use both WordNet and

[23] `http://babelnet.org`

DBpedia to semantically expand the entities added to the graph after the entity linking phase. Each synset obained from the linking is further expanded considering other concepts in the WordNet hierarchy of sysnsets by following the hypernymy[24] relations. From the WordNet hierarchy we extract up to 2-hop hypernyms starting from the mapped synset. We empirically selected the maximum distance of two hops because we wanted to avoid too broad generalization of the original concept. For the same reason we discard those hypernyms farther less than six hops away from the root of the WordNet hierarchy. Regarding DBpedia, Babelfy returns directly the URI of the linked entity and a set of related Wikipedia categories. In Wikipedia, categories are used to organize the resources, and they help users to group articles of the same subject. This reflects in DBpedia as resources are related to categories through the property `dcterms:subject`[25]. Those categories are in turn organized in a taxonomy. In particular, more specific categories are related to more generic ones by means of the `skos:broader`[26] property. Thus, for each category found by Babelfy, all the direct broader categories were gathered and added to our knowledge graph. Similarly to what we did with WordNet, only one level of broader categories were considered to avoid too broad or unrelated categories.

To show an example of entity linking performed by Babelfy we use the sound `prac-snare2.wav`[27] from Freesound. The description associated to this sound is *"standard snare sample. lower/mid tuning on the head"* and tags are *drums*, *percussion*, *snare*. Babelfy was able to detect and link most of the entities. Just to describe a few of them, the word *sample* from the description was linked to the DBpedia entity `Sampling_(music)`, the tag *percussion* was mapped to the DBpedia entity `Rythm_section`, the tag *snare* was linked to the WordNet concept `snare_drum.n.01` and DBpedia entity `Snare_drum`. As shown in Figure 6.1, DBpedia entities and WordNet synsets are then further enriched with their related categories and hypernyms. Following the Linked Data principles[28], we reused classes and properties from external vocabularies. The final knowledge graph after the entity linking and expansion process contains four main classes: `wordnet:Synset`, `Entity`, `Tag` and `skos:Concept` and seven relations: `hasTag`, `hasKeyword`, `wordnet:synset_member`, `dcterms:relation`, `dcterms:subject`, `skos:broader` and `wordnet:hypernym`[29]. In particular, for the sounds recommendation dataset based on Freesound we further enriched the ontology originally developed in Font & Oramas (2014) as also shown in the left hand side of Figure 6.1.

[24] Hypernymy models generalization relations between synsets.
[25] `http://dublincore.org/documents/dcmi-terms/#elements-subject`
[26] `http://www.w3.org/2004/02/skos/core#broader`
[27] `http://www.freesound.org/people/TicTacShutUp/sounds/439/`
[28] `http://www.w3.org/DesignIssues/LinkedData.html`
[29] All the prefix we use here are the ones available via the `http://prefix.cc` service

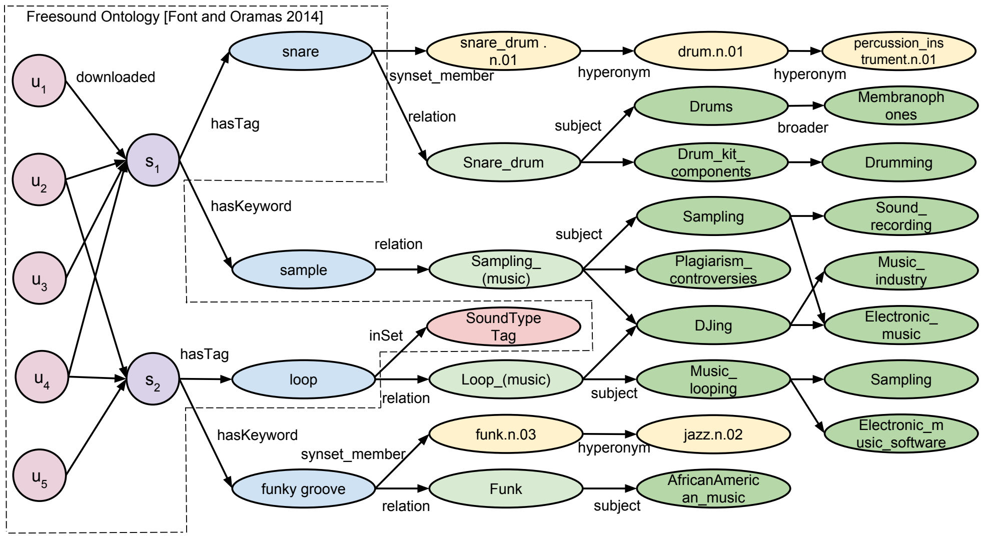


Figure 6.1: Portion of the final knowledge graph enriched with WordNet and DBpedia

6.3 Recommendation approach

As aforementioned, we adopted a hybrid recommendation approach to leverage both collaborative information coming from the user's community and content information coming from the knowledge graph. Following the taxonomy of hybrid recommender systems presented in Burke (2002) we developed a hybrid feature combination recommender system. The particularity of such schema is that hybridization is not based on the combination of different recommendation components but instead on the combination of different data sources. Specifically, collaborative information is treated as additional features of the content feature space and a content-based technique is used over this augmented space. Therefore, we build feature item representations by considering the item graph-based descriptions represented in the knowledge graph and enrich such feature vectors with collaborative features. Subsequently, we use such data to feed a content-based recommendation engine.

A common way of computing content-based recommendation is learning a function that, for each item in the system, predicts the relevance of such item for the user. The application of Machine Learning techniques is a typical way to accomplish such task. A *top-N* item recommendation problem in a standard content-based setting is mainly split into two different tasks: (i) given a collection of items for which past user's preferences are available, learn a regression or classification model to predict the relevance associated to unknown items; (ii) eventually, according to such scores, recommend the most relevant items to the user. Past user's preferences can be obtained from either explicit or implicit feedback. As for Freesound, we considered as an implicit positive feedback the "download data". The rationale behind our choice is that if a user downloads a sound it is reasonable to assume that she likes it even without an explicit rating, as the system lets users listen to sounds before downloading. Also the Last.fm dataset used in the experimental evaluation contains user song listening actions, which is another form of implicit feedback. Thus, in the following we will refer to the problem of computing recommendations from implicit feedback data. Following the notation introduced by Rendle et al. (2009) for implicit feedback scenarios, let S be the matrix of implicit feedback, where $s_{ui} = 1$ if item i was downloaded from user u, 0 otherwise. Starting from S we define $I_u^+ = \{i \in I | s_{ui} = 1\}$ as the set of relevant items for u. The main problem with implicit feedback is that they reflect only positive user preferences. On the contrary, the system cannot infer anything about what the user dislikes. The unobserved data are a mixture of actually negative and missing values Rendle et al. (2009), but the system does not have any information for discriminating between them. Then, learning a predictive model from such unary data becomes infeasible because there are no negative examples. To overcome this issue for each user we select a portion of unobserved items $I_u^- \subset (I \setminus I_u^+)$ to be used as negative data points in the training of the model. In Ostuni et al. (2013), the authors show that choosing $|I_u^-| = 2 \cdot |I_u^+|$ does not affect accuracy results. The unobserved items are exactly the items that have to be ranked. The ultimate goal of the system is to rank in the *top-N* positions items likely to be relevant for the user.

Given the generic user u, let T_u be the training set for u defined as:

$$T_u = \{\langle x_i, s_{ui} \rangle | i \in (I_u^+ \cup I_u^-)\}$$

where $x_i \in \mathbb{R}^D$ is the feature vector associated to the item i and let TS_u be the test set defined as:

$$TS_u = \{\langle x_i, s_{ui}^* \rangle | i \in (I \setminus I_u^+)\}$$

The two tasks for the *top-N* recommendation problem, in our setting, consist then of: (i) learning a function $f_u : \mathbb{R}^D \to \mathbb{R}$ from the training data T_u which assigns a relevance score to the items in I; (ii) using such function to predict the unknown score s_{ui}^* in the test set TS_u, to rank them and recommend the *top-N*.

Given that items are represented as entities in a knowledge graph we are particularly interested in those machine learning methods that are appropriate for dealing with objects structured as graphs. There are two main ways of learning with structured objects. The first is to use *Kernel Methods* Shawe-Taylor & Cristianini (2004). Given two input objects i and j, defined in an input domain space D, the basic idea behind Kernel Methods is to construct a kernel function $k : D \times D \to \mathbb{R}$, that can be informally seen as a similarity measure between i and j. This function must satisfy $k(i, j) = \langle \phi(i), \phi(j) \rangle$ for all $i, j \in D$, where $\phi : D \to F$ is a mapping function to a inner product feature space F. Then, the classification or regression task involves linear convex methods based exclusively on inner products computed using the kernel in the embedding feature space. The alternative way is to explicitly compute the *explicit feature mapping* $\phi(i)$ and to directly use linear methods in the related space. By transforming the graph domain into a vector domain any traditional learning algorithm working on feature vectors can be applied.

While kernel methods have been widely applied to solve different tasks, their usage becomes prohibitive when dealing with large datasets. In addition, when the input data lie in a high-dimensional space, linear kernels have performances comparable to more complex non linear ones. Due to the high volume of users we deal with in our Freesound dataset (see Section **??**), we focused on learning methods which are computationally efficient. For this reason we adopted the approach of computing the explicit feature mapping of the item graphs and use linear methods to learn the user model. Specifically, we use the Linear Support Vector Regression Ho & Lin (2012) algorithm. Regarding the explicit feature mapping computation we define two sparse high-dimensional feature maps: the one based on entities, the other on paths that we call *entity-based item neighborhood mapping* and *path-based item neighborhood mapping*, respectively. In the following we formalize the computation of such graph embeddings.

6.3.1 Explicit feature mappings for graph-based Item Representations

Let us formally define the knowledge graph as a multi-relational graph $G = \{t \mid t \in E \times R \times E\}$, where E denotes the set of entities and R indicates the set of properties or relations, namely the edge labels. Moreover, we have $I \subseteq E$ since we consider items as a particular type of entities.
With E_i^h we denote the set of entities reachable in *at most* h hops from i according to the shortest path in G. For a generic item i we then define its h-hop neighborhood graph $G_i^h = \{t = (e_i, r_j, e_k) \mid t \in E_i^h \times R \times E_i^h\}$ that is the subgraph of G induced by the set of triples involving entities in E_i^h.

Figure 6.2 shows an example of 3-hop item neighborhood graph for item i, namely G_i^3. We see that, if we consider the shortest path, all the entities are no more than 3 hops distant from i. To clarify the definition and computation of G_i^h and E_i^h for item i, we show their computation with reference to the example shown in Figure 6.2:
$G_i^1 = \{(i, p_1, e_1), (i, p_1, e_2), (e_3, p_2, i)\}$
$G_i^2 = G_i^1 \bigcup \{(e_1, p_3, e_4), (e_1, p_3, e_5), (e_2, p_4, e_5), (e_2, p_1, e_6), (e_2, p_1, e_7), (e_3, p_2, e_7), (e_3, p_1, e_8)\}$
$G_i^3 = G_i^2 \bigcup \{(e_4, p_3, e_9), (e_7, p_4, e_9), (e_7, p_3, e_{10}), (e_8, p_3, e_{10}), (e_8, p_3, e_{11})\}$
$E_i^1 = \{e_1, e_2, e_3\}$
$E_i^2 = E_i^1 \bigcup \{e_4, e_5, e_6, e_7, e_8\}$
$E_i^3 = E_i^2 \bigcup \{e_9, e_{10}, e_{11}\}$

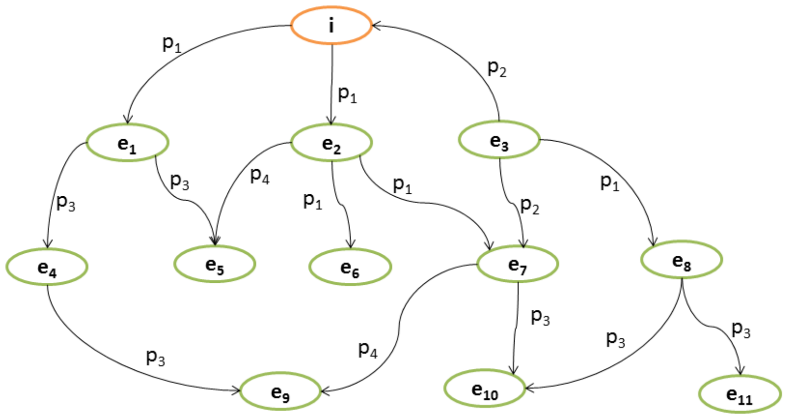


Figure 6.2: An example of 3-hop item neighborhood graph for the item i.

Starting from those item graph-based representations we define the two different feature mappings which are described in what follows.

Entity-based item neighborhood mapping In this mapping each feature refers to an entity in E and the corresponding score represents the weight associated to that entity in G_i^h. The resulting feature vector $\phi_E(G_i^h)$ is:

$$\phi_E(G_i^h) = (w_{i,e_1}, w_{i,e_2}, ...w_{i,e_m}, ..., w_{i,e_t})$$

where the weight associated to the generic entity e_m is computed as follows:

$$w_{i,e_m} = \sum_{l=1}^{h} \alpha_l \cdot c_{l,e_m}$$

with

$$\alpha_l = \frac{1}{1 + \log(l)}$$

and

$$c_{l,e_m} = |\{(e_n, p, e_m) \mid e_n \in \widehat{E}_i^{l-1} \wedge e_m \in \widehat{E}_i^l\} \bigcup \{(e_m, p, e_n) \mid e_m \in \widehat{E}_i^l \wedge e_n \in \widehat{E}_i^{l-1}\}|$$

where $\widehat{E}_i^l = E_i^l \setminus E_i^{l-1}$ is the set of entities *exactly* l hops far from i.
In particular, c_{l,e_m} corresponds to the number of triples connecting e_m to entities in the previous hop $(l-1)$, whether e_m appears either as subject or object of the triple. In other words, c_{l,e_m} can be seen as the *occurrence* of the entity e_m in the item neighborhood at distance l. The more the entity e_m is connected to neighboring entities of i, the more it is descriptive of i. α_l can be seen as a decay factor depending on the distance l from the item i, whose aim is to incrementally penalize farther entities from the item. It allows us

to take into account the *locality* of those entities in the graph neighborhood. The closer an entity e_m to the item i, the stronger its relatedness to it. We use a logarithmic decay. Indeed, the discount factor can also be parametrized defining a specific weight for each hop. In such case, an optimal combination of weights can be found.

With reference to example showed in Figure 6.2, the c_{l,e_m} values are computed as follows: $c_{1,e_1} = 1$, $c_{1,e_2} = 1$, $c_{1,e_3} = 1$, $c_{2,e_4} = 1$, $c_{2,e_5} = 2$, $c_{2,e_6} = 1$, $c_{2,e_7} = 2$, $c_{2,e_8} = 1$, $c_{3,e_9} = 2$, $c_{3,e_{10}} = 2$, $c_{3,e_{11}} = 1$. All the others are zero. The presented graph embedding is an adaptation of the one presented in Ostuni et al. (2014), in this work we use a logarithmic discount factor instead of a parametric one.

Path-based item neighborhood mapping Differently from the previous case, in this mapping we represent a feature as a sequence of nodes in G. Given two entities e_1 and e_n, we consider the sequence of nodes $e_1 \cdot e_2 \cdot \ldots \cdot e_{n-1} \cdot e_n$ met while traversing the graph to go from e_1 to e_n and we refer to such sequence as *path*. In this mapping, a feature is then represented by a path. In particular, in this mapping each feature refers to several variants of paths rooted in the item node. We first collect all the paths rooted in i which can be indicated as sequence of entities $i \cdot e_1 \cdot e_2 \cdot \ldots \cdot e_{n-1} \cdot e_n$. Then, starting from those paths we define various features considering sub-paths of the original paths. Specifically we form sub-paths composed by only those entities progressively farther from the item. Considering the path given above we build the following features: $e_1 \cdot e_2 \cdot \ldots \cdot e_{n-1} \cdot e_n$, $e_2 \cdot \ldots \cdot e_{n-1} \cdot e_n$, ..., $e_{n-1} \cdot e_n$, e_n. The rationale behind this choice is that it allows to explicitly represent substructures shared between items with no overlapping in their immediate neighborhoods but somehow connected at further distance. Items connected to the same entities have same common structures because both closer and further entities are shared. Items connected to different entities which are however linked directly or at a farther distance to same entities share less or none sub-paths depending on how much far the common entities are, if any.

More formally, let P_i be the set of paths rooted in i and P_i^* be the list of all possible sub-paths extracted from them. We use $p_m(i)$ and $p_m^*(i)$ to refer to the $m-th$ elements in P_i and P_i^*, respectively. Then, the feature mapping for item i is:

$$\phi_P(G_i^h) = (w_{i,p_1^*}, w_{i,p_2^*}, ...w_{i,p_m^*}, ..., w_{i,p_t^*})$$

where each w_{i,p_m^*} is computed as:

$$w_{i,p_m^*} = \frac{\#p_m^*(i)}{|p_m| - |p_m^*|}$$

where $|p_m|$ indicates the length of path p_m and $\#p_m^*(i)$ the occurrence of $p_m^*(i)$ in P_i^*. The denominator is a discounting factor which takes into account the difference between the original path p_m and its sub-path p_m^*. The shorter the

sub-path the more the discount because it contains entities farther from the item.
With respect to item i we have:
$P_i = \{i \cdot e_1 \cdot e_4 \cdot e_9,\ i \cdot e_1 \cdot e_5,\ i \cdot e_2 \cdot e_6,\ i \cdot e_2 \cdot e_7 \cdot e_9,\ i \cdot e_2 \cdot e_7 \cdot e_{10},\ i \cdot e_3 \cdot e_7 \cdot e_{10},\ i \cdot e_3 \cdot e_8 \cdot e_{10},\ i \cdot e_3 \cdot e_8 \cdot e_{11}\}$
$P_i^* = [e_1 \cdot e_4 \cdot e_9,\ e_4 \cdot e_9,\ e_9,\ e_1 \cdot e_5,\ e_5,\ e_2 \cdot e_6,\ e_6,\ e_2 \cdot e_7 \cdot e_{10},\ e_7 \cdot e_{10},\ e_{10},\ e_2 \cdot e_7 \cdot e_9,\ e_7 \cdot e_9,\ e_9,\ e_3 \cdot e_7 \cdot e_{10},\ e_7 \cdot e_{10},\ e_{10},\ e_3 \cdot e_8 \cdot e_{10},\ e_8 \cdot e_{10},\ e_{10},\ e_3 \cdot e_7 \cdot e_{11},\ e_7 \cdot e_{11},\ e_{11}]$

6.3.2 Feature Combination

Each final feature vector x_i is obtained by concatenating a vector of collaborative features $\phi_{col}(i)$ to the item neighborhood mapping vector $\phi(G_i^h)$. Collaborative features are simply added by encoding in the feature vector those users who downloaded that item. The collaborative feature vector regarding the generic item is then:

$$\phi_{col}(i) = (w_{i,u_1}, w_{i,u_2}, ..., w_{i,u_1})$$

where $w_{i,u_1} = 1$ if user u_1 downloaded item i.

Although more sophisticated and advanced methods can be used for feature combination Beliakov et al. (2015), our experimental evaluation (see Section **??**) shows the effectiveness of our choice.

6.4 Experimental Evaluation

For the evaluation of our approach we adopted the `All Unrated Items` methodology presented in Steck (2013). It consists in creating a *top-N* recommendation list for each user by predicting a score for every item not rated by that particular user, whether the item appears in the user test set or not. Then, performance metrics are computed comparing recommendation lists with test data. The evaluation has been carried out using the holdout method consisting in splitting the data in two disjoint sets: the one for training and the other for testing. We used 80% of user downloads for building the training set T and remaining 20% as test data for measuring recommendation accuracy. We repeated the procedure three times by randomly drawing new training/test sets in each round and averaged the results.

For measuring recommendation accuracy we adopted the following standard performance metrics: Precision and Recall. Precision@N (P@N) is computed as the fraction of *top-N* recommended items appearing in the test set, while Recall@N (R@N) is computed as the ratio of *top-N* recommended items appearing in the test set to the number of items in the test set. Note that in such implicit feedback setting all items in the test set are relevant. In addition to the standard precision and recall metrics we also measure the Mean Reciprocal

Rank (MRR) which measure the quality of the highest ranked recommendations. For each user recommendation list the Reciprocal Rank (RR) measures how early in the list is positioned the first relevant recommendation.

As pointed out by McNee et al. (2006), the most accurate recommendations according to the standard metrics are sometimes not the recommendations that are most useful to users. In order to assess the utility of a recommender system, it is extremely important to evaluate also its capacity to suggest items that users would not readily discover for themselves, that is its ability to generate novel and unexpected results. The *Entropy-Based Novelty* (*EBN*) Bellog\'\in et al. (2010) expresses the ability of a recommender system to suggest less popular items, i.e. items not known by a wide number of users. In particular, for each user's recommendation list L_u, the novelty is computed as:

$$EBN_u@N = -\sum_{i \in L_u} p_i \cdot \log_2 p_i$$

where:

$$p_i = \frac{|\{s_{ui} = 1 | u \in U\}|}{|U|}$$

Particularly, p_i is the ratio of users who downloaded item i. The lower $EBN_u@N$, the better the novelty.

Another important quality of the system is aggregate diversity. In our work we adopt the *diversity-in-top-N* metric presented in Adomavicius & Kwon (2012) that measures the distinct items recommended across all users. In particular we compute its normalized version with respect to the size of the item catalog. For brevity we refer to it as $ADiv@N$ and we compute it as follows:

$$ADiv@N = \frac{|\bigcup_u L_u|}{|I|}$$

This metric is an indicator of the level of personalization provided by a recommender system. Low values of aggregated diversity indicate that all users are being recommended almost the same few items. This corresponds to a low level of personalization of the system. Instead, high values mean that users receive very different recommendations which can be indirectly seen as a high level of personalization of the system.

All the reported metrics, besides aggregated diversity, are computed for each single user and eventually averaged.

6.4.1 Datasets Description

Freesound Dataset We evaluated our approach on historical data about sound downloads collected from February 2005 to October 2013. The initial dump consisted in 3,275,092 users, 183,246 sounds and 48,636,182 downloads. However, for the purpose of our experimentation, we selected a subset of sounds

dataset	items	avg. tags	avg. keywords	resources	synsets	categories
Freesound	21,552	6.44	11.36	16,407	20,034	54,419
Last.fm	8,640	42.09	77.33	46,109	27,708	96,942

Table 6.1: Number of tags and keywords identified by Babelfy averaged by item, plus total number of distinct DBpedia resources, WordNet synsets and Wikipedia categories.

that fulfilled some criteria. We selected those sound with at least two tags classified in the Freesound Ontology Font & Oramas (2014). After that we filtered out all sounds with less than 10 downloads to reduce the sparsity of the implicit feedback matrix and have a fairer comparison with pure collaborative filtering methods. After some further data cleansing, the final dataset consisted in 20,000 users, 21,552 items and 2,117,698 downloads[30]. The sparsity of the implicit feedback matrix was 99.51%. Statistics on the enriched knowledge graph of the final dataset are shown in Table **??**.

Last.fm Dataset To recreate most of the conditions of the Freesound dataset in a typical music recommendation scenario, a new dataset is created combining user's implicit feedback, tags and textual descriptions of songs. This dataset combines a corpus of user's listening habits and song-related tags coming from Last.fm[31] Vigliensoni & Fujinaga (2014), with a corpus of textual descriptions about songs obtained from Songfacts.com[32] Sordo et al. (2015). The former is an implicit feedback dataset consisting of user-song listening data, indicating the frequency a user listened to a song. For every user in the corpus we chose the users' average listening count as a threshold to identify the relevant songs for each user. From Last.fm, we only selected for our dataset user-song relations with a number of listens above each user's threshold. Moreover, only those songs that were relevant to at least 10 users, and users with at least 50 relevant songs were added to the dataset. The final dataset consisted in 5,199 users, 8,640 songs and 751,531 relations between users and songs. The sparsity of the implicit feedback matrix was 98.33%. This collaborative information was complemented with the list of top tags of every song provided by the Last.fm API, and a textual description of each song coming from Songfacts.com. Information about the enriched knowledge graph is shown in Table **??**.

[30] A dump of the datasets is available at `http://mtg.upf.edu/download/datasets/knowledge-graph-rec`

[31] `http://last.fm`

[32] `http://songfacts.com`

6.4.2 Experiment settings

As mentioned in Section **??**, each user model is learnt using the Linear Support Vector Regression method. In particular we adopted the efficient *LIBLINEAR*[33] library and chose the *L2-regularized Support Vector Regression* Ho & Lin (2012). The tuning of the model hyper-parameters of the learning algorithm was performed through cross-validation on validation data obtained by selecting the 15% of feedback for each user from the training data. We set the parameters C and e by using a grid-search varying C from 0.1 to 1000 with step 10 and $e = \{0.1, 0.01\}$ (tolerance of termination criterion). Before the training we performed some pre-processing on the feature vectors. We removed those features appearing in less then 5 sounds and scaled all features to the range $[0, \ldots, 1]$ using min-max normalization. Finally each feature vector was normalized to unit length using the L2 norm.

Regarding the run time performances of the entire recommender for the Freesound experiment, the highest computation time (corresponding to the path-based feature mapping with 3-hops) lasted about 28 minutes, from feature extraction to recommendation generation, on a dedicated server machine with 4 Xeon quad-core 2.93GHz processors and 32GB RAM. Since each user model is learnt independently, the learning process is highly parallelizable. Moreover, being a model-based recommender, each user model learning can be performed offline periodically once a certain number of new feedbacks are accumulated for that specific user. The implementation of the recommendation algorithm presented in this work is available on GitHub [34].

In the following we describe the experiments we carried out to evaluate our approach. In particular we are interested in evaluating the impact of semantic enrichment of the original data on the recommendation quality and the differences among the two feature mapping methods we implemented. Furthermore, we compare our approach with state of the art algorithms for implicit feedback scenarios.

6.4.3 Sound Recommendation Experiment

Evaluation of the semantic item description enhancement To evaluate the impact of the various features and information sources we built several variants of item feature vectors by varying: the information sources considered, the size of the item neighborhood graphs (number of hops) and the feature mapping method. In addition, we built a content-based approach purely based on 352 low-level audio features[35] extracted from the sound signal by using Essentia Bogdanov et al. (2013). In this approach, predictions are computed by aggregating the Euclidean distances between the sounds downloaded by the

[33] http://www.csie.ntu.edu.tw/~cjlin/liblinear/
[34] https://github.com/sisinflab/lodreclib
[35] https://www.freesound.org/docs/api/analysis_example.html#all-descriptors

Approach	**Enrichment**	**h-hops**	**MRR**	**P@10**	**R@10**	**EBN@10**	**ADiv**
Ent	fso	h=3	0.303	0.113	0.065	2.791	0.25
Ent	fso+wn+db/tags	h=3	0.303	0.115	0.066	2.617	0.33
Ent	fso+wn+db/tags	h=4	0.302	0.114	0.065	2.507	0.36
Ent	fso+wn+db/keyw+tags	h=3	**0.306**	**0.118**	**0.067**	2.426	0.36
Ent	fso+wn+db/keyw+tags	h=4	**0.306**	0.117	0.066	2.303	0.39
Path	fso	h=3	0.301	0.113	0.065	2.750	0.28
Path	fso+wn+db/tags	h=3	0.301	0.114	0.064	2.279	0.46
Path	fso+wn+db/tags	h=4	0.292	0.106	0.059	1.863	**0.55**
Path	fso+wn+db/keyw+tags	h=3	0.304	0.116	0.065	2.019	0.46
Path	fso+wn+db/keyw+tags	h=4	0.296	0.111	0.061	**1.618***	0.53
Collab			0.293	0.110	0.062	2.890	0.18
Ent - noCollab	fso+wn+db/keyw+tags	h=3	0.154	0.058	0.034	0.384	0.59
Path - noCollab	fso+wn+db/keyw+tags	h=3	0.151	0.049	0.028	**0.369**	**0.67**
VSM	keyw+tags	h=1	0.301	0.116	0.066	2.621	0.30
VSM - noCollab	keyw+tags	h=1	0.151	0.055	0.032	0.389	**0.67**
Audio Sim			0.022	0.004	0.002	0.382	0.04

Table 6.2: Accuracy, Novelty and Aggregate Diversity results for different versions of the Freesound dataset. Best values in each column are in bold. The * symbol indicates best values for hybrid and collaborative configurations. `Ent` and `Path` refers to graph embedding options; `fso`, `wn` and `db` to the initial Freesound Ontology, WordNet and DBpedia respectively; `tags` to item tags, and `keyw` to text description keywords; `h` indicates the length of the h-hop neighborhood graph; `Collab` means that only collaborative features are considered; `noCollab` that no collaborative features are considered; `VSM` refers to Vector Space Model embedding; `Audio Sim` to the audio-based approach.

user and the target sound to recommend. All the results are reported in Table **??**.

Looking at the accuracy results we see that there are no marked differences among all the feature vector variants. Noteworthy is that without considering the collaborative information (`noCollab`) the accuracy drops significantly. In addition, when considering only collaborative features accuracy performances are comparable with respect to hybrid feature combination variants. The best hybrid semantic version `Ent(fso+wn+db/keyw+tags/h=3)` is slightly better than pure collaborative (+0.8% in terms of P@10). Regarding the comparison of the two mapping methods, the Entity-based item neighborhood mapping has generally slightly higher accuracy than the Path-based one. We can also note that considering too far entities does not improve accuracy. In fact, in both the two feature mapping when four hops are considered the results drop slightly with respect to three hops. Finally, we see that the semantic expansion of tags and terms do not improve consistently accuracy with respect to the usage of pure keywords and tags combined with collaborative information. The semantic configuration with highest accuracy (`Ent(fso+wn+db/keyw+tags/h=3)`) is only 0.2% better in terms of P@10 with respect to `VSM keyw+tags`. We can also observe that the pure audio based approach (`Audio Sim`) has by far lower performances than all the others. All the differences between the hybrid graph embeddings and the other baselines are statistically significant ($p < 0.01$) according to the paired t-test.

Novelty and aggregate diversity results instead show more interesting insights. We observe that the semantic expansion, with both feature mappings, results in an improving of both novelty and aggregated diversity. In fact, the semantic enriched variant `(fso+wn+db+keyw+tags/h=4)` has much better novelty and diversity than considering only the original tagging ontology `(fso)`. Furthermore, with respect to the variants without semantic expansion, that is the variants based only on keywords and tags, the usage of semantic expansion improves considerably novelty and diversity. Hence, thanks to this exploitation of the knowledge graph we are able to recommend good items which are also not so popular. We also see that the Path-based embedding has better performances than the Entity-based one. Such approaches allow to explore better the long tail distribution of items and to increase the personalization of the system.
The variants without collaborative information are the ones with better novelty and diversity. The reason behind this behavior is that pure content-based approaches are not influenced by popularity biases. However, when using only content data the system recommends unpopular but very inaccurate items. Good novelty without accuracy does not imply good recommendation quality. Finally, the usage of only collaborative information has much lower catalog coverage (aggregate diversity) than feature vectors containing also semantic features. For example `Path(fso+wn+db+keyw+tags/h=4)` has comparable performances in terms of accuracy with respect to `Collab` but considerably better catalog coverage and novelty (lower EBN).

To conclude, we can state that the semantic expansion, especially when combined with the Path-based mapping, improves recommendation quality in terms of novelty and aggregated diversity. The intuition behind these results is that the semantic expansion allows the system to find items semantically related to the ones in the user profile. Conversely, when using only keyword or tag-based representations the system is able to retrieve only those few items with an exact keyword/tag match with those liked by the user. Thus, the system is unable to widely explore the item space to find those items which are semantically related to the ones liked by the user.

Comparison with other methods We compared our approach with several state of the art recommendation algorithms. `MostPop` is a popularity-based baseline which provides the same recommendation to all users based

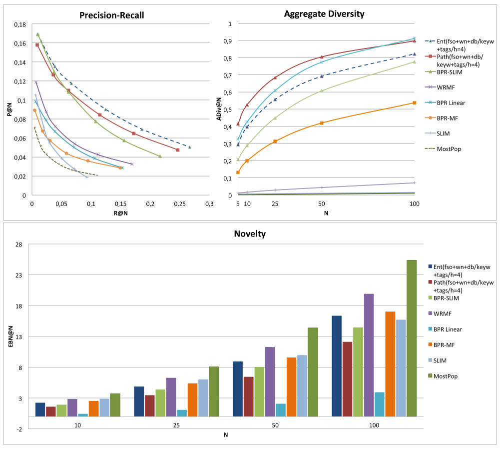


Figure 6.3: Precision-Recall, Novelty and Aggregate Diversity plots in Freesound dataset

on the global popularity of items. `BPR-MF` Rendle et al. (2009) is a matrix factorization-based method optimized with Bayesian Personalized Ranking optimization criterion. `WRMF` is a weighted matrix factorization method Hu et al. (2008). `SLIM` Ning & Karypis (2012) uses a Sparse Linear method for learning a sparse aggregation coefficient matrix. `BPR-SLIM` is similar to `SLIM` but it uses the BPR optimization criterion. `BPR Linear` is a hybrid matrix factorization method able to work with sparse datasets Gantner et al. (2010). We used keywords and tags as item attribute data. The computation of the recommendations for all these comparative algorithms has been done with the publicly available software library *MyMediaLite*[36].

Figure 6.3 shows precision-recall, novelty and aggregated diversity plots. In those plots we report the competitive algorithms used for comparison and the `Ent(fso+wn+db/keyw+tags/h=4)` and `Path(fso+wn+db+keyw+tags/h=4)` configurations which we chose as representative for our approach due to its performances in terms of novelty and aggregate diversity.
With reference to the accuracy results we notice that our two approaches largely outperforms the others. The only method which is close to the approaches we propose is `BPR-SLIM` which slightly outperforms `Path(fso+wn+db+keyw+tags/h=4)` for low values of recommendation list length ($N = 5, 10$). All differences between our approach and the other methods are statistically significant ($p < 0.01$) according to the paired t-test. With respect to the Novelty plot, our approach has much better novelty than all the other collaborative filtering algorithms but `BPR Linear` which however have much lower accuracy. Our approach outperforms most of the collaborative filtering algorithms in terms of aggregated diversity. It is able to achieve a coverage of almost 80% and 90% for $N = 50$ and $N = 100$, respectively. The approach closer to ours is `BPR`

[36] `http://www.mymedialite.net/`.

Approach	Enrichment	h-hops	MRR	P@10	R@10	EBN@10	ADiv@
Ent	wn+db/tags	h=2	**0.612**	0.321	**0.122**	2.414	0.357
Ent	wn+db/tags	h=3	**0.612**	0.319	0.121	2.383	0.374
Ent	wn+db/tags	h=4	0.599	0.314	0.119	2.356	0.389
Ent	wn+db/keyw+tags	h=3	0.604	0.315	0.114	2.448	0.316
Ent	wn+db/keyw+tags	h=4	0.601	0.312	0.113	2.424	0.331
Path	wn+db/tags	h=3	0.570	0.287	0.108	2.112	0.479
Path	wn+db/tags	h=4	0.537	0.260	0.097	**1.911***	**0.544***
Path	wn+db/keyw+tags	h=3	0.570	0.289	0.104	2.173	0.411
Path	wn+db/keyw+tags	h=4	0.537	0.259	0.093	1.942	0.484
Collab			0.597	0.313	0.113	2.664	0.240
Ent - noCollab	wn+db/tags	h=3	0.292	0.114	0.043	0.983	0.703
Path - noCollab	wn+db/tags	h=3	0.285	0.113	0.043	**0.981**	**0.736**
VSM	tags	h=1	0.610	**0.322**	**0.122**	2.454	0.346
VSM	keyw	h=1	0.599	0.309	0.112	2.642	0.249

Table 6.3: Accuracy, Novelty and Aggregate Diversity results for different versions of the Last.fm dataset. Best values in each column are in bold. The * symbol indicates best values for hybrid and collaborative configurations.

`Linear` that for $N = 100$ reaches same performances. Also, `BPR-SLIM` and `BPR-MF` have acceptable diversity results. Instead, all the others have very low diversity results meaning that they focus mostly on a few specific items and recommend them to all users indiscriminately.
Summing up, the experimental results show that our approach is able to give more accurate and at the same time less popular recommendations, than collaborative filtering methods. It is able to better find good recommendations in the long tail. Effective recommendation systems should promote novel and relevant items taken primarily from the tail of the distribution. In addition, our approach shows much higher aggregated diversity which can be seen as a higher personalization of the system.

6.4.4 Music Recommendation Experiment

The recommendation algorithms we propose have been further validated on the Last.fm dataset. We performed the same experiments on this dataset to assess the applicability of the approach to other musical contexts.

Evaluation of the semantic item description enhancement As we may notice from the results shown in Table **??**, Entity-based embedding, `Collab`, and `VSM tags` approaches have very similar performance in terms of precision and recall. The first two Entity-based embedding variants have slightly higher MRR than `VSM tags`, meaning that they better locate relevant items in the

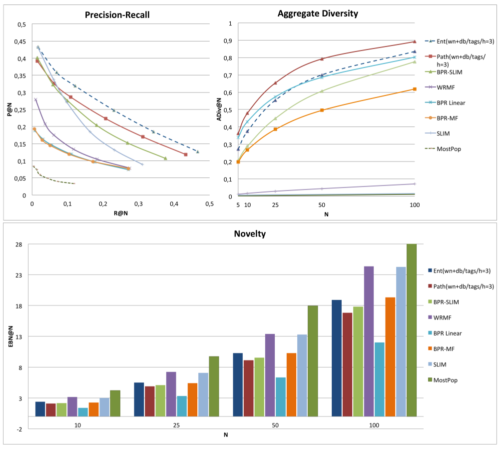


Figure 6.4: Precision-Recall, Novelty and Aggregate Diversity plots in Last.fm dataset

top positions. Analogously to the previous sounds recommendation task, the approaches exploiting semantic expansion outperform the others in terms of novelty and aggregated diversity. The same tendency of the previous experiment is observed with the Entity-based and Path-based item neighborhood mappings. The Path-based approaches have lower precision, but much better novelty and aggregated diversity. Moreover, it is very interesting to observe that for both embedding options if we expand the graph by means of farther entities (h=4) precision decreases whilst novelty and diversity improve. It is noteworthy thet differently from the results of the Freesound experiment, here we obtain higher accuracy with the approach that uses only tags and not keywords. Our interpretation of this trend is that, as shown in Table **??**, the number of tags in the Freesound dataset is somehow scarce, and the addition of keywords taken from the textual descriptions improves the annotation of the items. On the other side, in the Last.fm dataset, the set of tags is already very rich, then the addition of keywords introduces noise within the items description thus deteriorating the accuracy of recommendations. Also in this experiment we can observe that when no collaborative feature is used, accuracy is significantly worse even if novelty and diversity seem to be better. We may confirm from results in both experiments that collaborative features are a very strong signal for the accuracy of the recommendations. Nonetheless, the inclusion of semantic features allows the system to further improve accuracy and provide novel and diverse recommendations, thus better leveraging the long tail. All the differences between the hybrid graph embeddings and the other baselines are statistically significant ($p < 0.01$) according to the paired t-test.

Comparison with other methods We compared our approach with the same set of state of the art algorithms presented in the sound recommendation experiment. Based on the observations made in the previous paragraph, we used for this experiment only tags as item attribute data for `BPR Linear`. Figure 6.4 shows precision-recall, novelty and aggregated diversity plots of the comparison with the other methods. We compare the competitive algorithms with the `Ent(wn+db/tags/h=3)` and `Path(wn+db/tags/h=3)` configurations which in this scenario results to be the most representative for our approach. Results are pretty similar to the ones observed in the sound recommendation experiment. Our two approaches largely outperforms the others in terms of accuracy. `BPR-SLIM` and `SLIM` have performance similar to our Entity-based mapping approach for low values of recommendation list length (N = 5, 10), and slightly higher that the Path-based one. All differences between our approaches and the other methods are statistically significant ($p < 0.01$) according to the paired t-test. Our approaches have much better novelty results than all other collaborative filtering algorithms but `BPR Linear`, which again has much lower accuracy. In terms of aggregated diversity, our approach outperforms most of the collaborative filtering algorithms. `BPR Linear` achieves similar diversity, but much lower accuracy. Summing up, our approach is able to recommend less popular items with higher accuracy than other collaborative filtering algorithms also in this recommendation scenario. Therefore, our approach is able to improve the level of personalization of the recommended items, and better explore the long tail also for songs recommendation.

6.5 Conclusion

We have presented a hybrid approach to recommend musical items, i.e. sounds and songs, by exploiting the information encoded within a knowledge graph. We conducted various experiments on two different datasets, the one of sounds coming from Freesound.org, the other one of songs gathered from Last.fm and Songfacts.com. They may be considered as representative of the two classes of users we find the music domain: producers looking for sounds to create new music and consumers looking for new songs to listen to.

Information coming from item descriptions and tags have been enriched with data coming from two external knowledge graphs: DBpedia and WordNet. Entity Linking tools have been adopted to extract relevant entities from textual sources associated to musical items, namely tags and text descriptions, thus creating a new graph encoding the knowledge associated to users, items and their mutual interactions. We then developed a recommendation engine that combines different features, that is semantic content-based ones extracted from the resulting knowledge graph and collaborative information from implicit user feedback. An evaluation with two explicit feature mappings, *entity-based item neighborhood* and *path-based item neighborhood*, has been conducted on both datasets in order to asses the performance of the system in terms of accuracy, diversity and novelty.

Experimental results in sounds and songs recommendation show that the proposed approach is able to improve the quality of the recommended list with respect to state of the art collaborative filtering algorithms and with respect to other content-based baselines. Our results also show that the data related to the music knowledge domain encoded in freely available datasets such as DBpedia or WordNet have reached a quality level that makes possible its usage in the creation of recommendation engines whose target are either music producers or music consumers. The semantic enrichment of the initial knowledge graph performed by means of entity linking techniques is a good choice to boost the performances of the system in terms of novelty and aggregate diversity. A knowledge-based approach can improve the degree of personalization in the recommendations of musical items from various points of view such as prediction accuracy, catalog coverage and promote long tail recommendations. We have presented a methodology that achieves these objectives by combining semantic knowledge with collaborative information.

Summing up, knowledge graphs can be a useful tool when properly leveraged within recommender systems for musical items. Indeed, the graph-based nature of the information they contain, on the one hand, makes possible a linkage to other graphs thus resulting in an easy plugging of new content-based data. On the other hand, by exploring the graph new connections and commonalities between items and users can be discovered and exploited while computing the recommendation list.

CHAPTER 7

Music Genre Classification Using Linear Models

7.1 Introduction

With the democratisation of Internet access, vast amounts of information are generated and stored in online sources, and thus there is great interest in developing techniques for processing this information effectively Ruiz-Casado et al. (2008). The Music Information Retrieval (MIR) community is sensible to this reality, as music consumption has undergone significant changes recently, especially since users are today just one click away from millions of songs Celma & Herrera (2008). This context results in the existence of large repositories of unstructured knowledge, which have great potential for musicological studies or tasks within MIR such as music recommendation.

In this paper, we put forward an integration procedure for enriching with music-related information a large dataset of Amazon customer reviews McAuley et al. (2015a,b), with semantic and acoustic metadata obtained from MusicBrainz[37] and AcousticBrainz[38], respectively. MusicBrainz (MB) is a large open music encyclopedia of music metadata, whist AcousticBrainz (AB) is a database of music and audio descriptors, computed from audio recordings via state-of-the-art Music Information Retrieval algorithms Porter et al. (2015). In addition, we further extend the *semantics* of the textual content from two standpoints. First, we apply an aspect-based sentiment analysis framework Dong et al. (2013) which provides specific sentiment scores for different aspects present in the text, e.g. album cover, guitar, voice or lyrics. Second, we perform Entity Linking (EL), so that mentions to named entities such as Artist Names or Record Labels are linked to their corresponding Wikipedia entry Oramas et al. (2016).

[37] `http://musicbrainz.org/`
[38] `http://acousticbrainz.org`

This enriched dataset, henceforth referred to as Multimodal Album Reviews Dataset (MARD), includes affective, semantic, acoustic and metadata features. We benefit from this multidimensional information to carry out two experiments. First, we explore the contribution of such features to the Music Genre classification task, consisting in, given a song or album review, predict the genre it belongs to. Second, we use the substantial amount of information at our disposal for performing a diachronic analysis of music criticism. Specifically, we combine the metadata retrieved for each review with their associated sentiment information, and generate visualizations to help us investigate any potential trends in diachronic music appreciation and criticism. Based on this evidence, and since music evokes emotions through mechanisms that are not unique to music Juslin & Västfjäll (2008), we may go as far as using musical information as means for a better understanding of global affairs. Previous studies argue that national confidence may be expressed in any form of art, including music Moïsi (2010), and in fact, there is strong evidence suggesting that our emotional reactions to music have important and far-reaching implications for our beliefs, goals and actions, as members of social and cultural groups Alcorta et al. (2008). Our analysis hints at a potential correlation between the language used in music reviews and major geopolitical events or economic fluctuations. Finally, we argue that applying sentiment analysis to music corpora may be useful for diachronic musicological studies.

7.2 Multimodal Album Reviews Dataset

MARD contains texts and accompanying metadata originally obtained from a much larger dataset of Amazon customer reviews McAuley et al. (2015a,b). The original dataset provides millions of review texts together with additional information such as overall rating (between 0 to 5), date of publication, or creator id. Each review is associated to a product and, for each product, additional metadata is also provided, namely Amazon product id, list of similar products, price, sell rank and genre categories. From this initial dataset, we selected the subset of products categorized as *CDs & Vinyls*, which also fulfill the following criteria. First, considering that the Amazon taxonomy of music genres contains 27 labels in the first hierarchy level, and about 500 in total, we obtain a music-relevant subset and select 16 of the 27 which really define a music style and discard for instance region categories (e.g. World Music) and other categories non specifically related to a music style (e.g. Soundtrack, Miscellaneous, Special Interest), function-oriented categories (Karaoke, Holiday & Wedding) or categories whose albums might also be found under other categories (e.g. Opera & Classical Vocal, Broadway & Vocalists). We compiled albums belonging only to one of the 16 selected categories, i.e. no multiclass. Note that the original dataset contains not only reviews about CDs and Vinyls, but also about music DVDs and VHSs. Since these are not strictly speaking music audio products, we filter out those products also classified as "Movies & TV". Finally, since products classified as Classical and Pop are substantially more frequent in the original dataset, we compensate this unbalance by limiting the number of albums of any genre to 10,000. After this preprocessing, MARD amounts to a total of 65,566 albums and 263,525 customer reviews. A breakdown of the number of albums per genre is provided in Table 7.1.

Genre	**Amazon**	**MusicBrainz**	**AcousticBrainz**
Alternative Rock	2,674	1,696	564
Reggae	509	260	79
Classical	10,000	2,197	587
R&B	2,114	2,950	982
Country	2,771	1,032	424
Jazz	6,890	2,990	863
Metal	1,785	1,294	500
Pop	10,000	4,422	1701
New Age	2,656	638	155
Dance & Electronic	5,106	899	367
Rap & Hip-Hop	1,679	768	207
Latin Music	7,924	3,237	425
Rock	7,315	4,100	1482
Gospel	900	274	33
Blues	1,158	448	135
Folk	2,085	848	179
Total	66,566	28,053	8,683

Table 7.1: Number of albums by genre with information from the different sources in MARD

Having performed genre filtering, we enrich MARD by extracting artist names and record labels from the Amazon product page. We pivot over this information to query the MB search API to gather additional metadata such as release id, first release date, song titles and song ids. Mapping with MB is performed using the same methodology described in Oramas et al. (2015a), following a pair-wise entity resolution approach based on string similarity with a threshold value of $\theta = 0.85$. We successfully mapped 28,053 albums to MB. Then, we retrieved songs' audio descriptors from AB. From the 28,053 albums mapped to MB, a total of 8,683 albums are further linked to their corresponding AB entry, which encompasses 65,786 songs. The final dataset is freely available for download[39].

7.3 Text Processing

In this section we describe how we extract linguistic, sentimental and semantic information from textual reviews. This information will serve both as input features for our genre classification experiments, and also constitute the basis for the diachronic study described in Section 8.4.

[39] http://mtg.upf.edu/download/datasets/mard

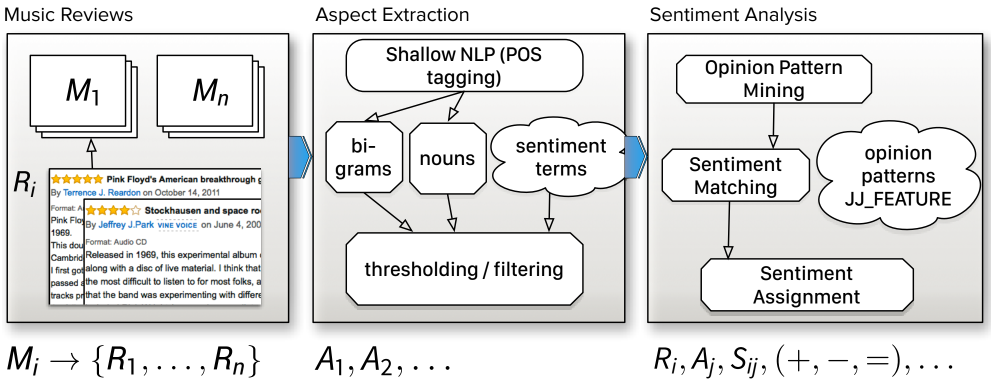


Figure 7.1: Overview of the opinion mining and sentiment analysis framework.

7.3.1 Sentiment Analysis

Following the work of Dong et al. (2013, 2014) we use a combination of shallow NLP, opinion mining, and sentiment analysis to extract opinionated features from reviews. For reviews R_i of each album, we mine bi-grams and single-noun aspects (or review features), see Hu & Liu (2004); e.g. bi-grams which conform to a noun followed by a noun (e.g. *chorus arrangement*) or an adjective followed by a noun (e.g. *original sound*) are considered, excluding bi-grams whose adjective is a sentiment word (e.g. *excellent*, *terrible*). Separately, single-noun aspects are validated by eliminating nouns that are rarely associated with sentiment words in reviews, since such nouns are unlikely to refer to item aspects. We refer to each of these extracted aspects A_j as review aspects.

For a review aspect A_j we determine if there are any sentiment words in the sentence containing A_j. If not, A_j is marked neutral, otherwise we identify the sentiment word w_{min} with the minimum word-distance to A_j. Next we determine the POS tags for w_{min}, A_i and any words that occur between w_{min} and A_i. We assign a sentiment score between -1 and 1 to A_j based on the sentiment of w_{min}, subject to whether the corresponding sentence contains any negation terms within 4 words of w_{min}. If there are no negation terms, then the sentiment assigned to A_j is that of the sentiment word in the sentiment lexicon; otherwise this sentiment is reversed. Our sentiment lexicon is derived from SentiWordNet Esuli & Sebastiani (2006) and is not specifically tuned for music reviews. An overview of the process is shown in Figure 7.1. The end result of sentiment analysis is that we determine a sentiment label S_{ij} for each aspect A_j in review R_i. A sample annotated review is shown in Figure 7.2

7.3.2 Entity Linking

Entity Linking (EL) is the task to provide, given a mention to a named entity (e.g. person, location or organization), its most suitable entry in a reference Knowledge Base (KB) Moro et al. (2014a). In our case, EL was performed

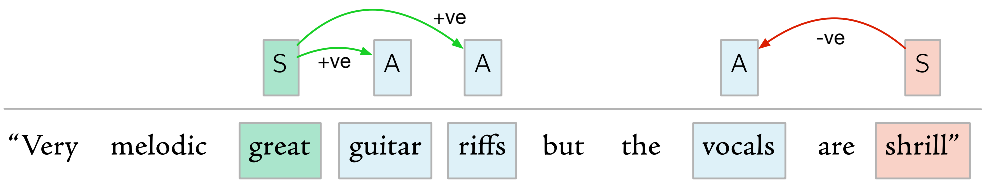


Figure 7.2: A sentence from a sample review annotated with opinion and aspect pairs.

	Alt. Rock	Classical	Country	Electronic	Folk	Jazz	Latin	Metal	New Age	Pop
Alt. Rock	28 / 42	1 / 3	3 / 1	10 / 10	7 / 1	1 / 2	2 / 0	18 / 12	10 / 2	4 / 10
Classical	0 / 0	87 / 95	1 / 0	0 / 0	1 / 1	1 / 1	2 / 2	1 / 0	5 / 1	1 / 0
Country	2 / 1	0 / 0	51 / 84	3 / 0	9 / 1	9 / 0	3 / 0	0 / 1	3 / 0	8 / 8
Electronic	7 / 3	3 / 1	1 / 2	40 / 61	4 / 1	1 / 2	2 / 2	6 / 0	7 / 5	6 / 5
Folk	4 / 6	11 / 0	13 / 10	7 / 0	27 / 55	6 / 1	7 / 3	4 / 2	6 / 9	5 / 9
Jazz	7 / 0	10 / 1	6 / 2	2 / 2	5 / 0	45 / 82	6 / 3	3 / 0	8 / 2	3 / 5
Latin	4 / 3	6 / 4	9 / 2	1 / 2	5 / 1	10 / 2	28 / 78	3 / 0	6 / 2	11 / 4
Metal	13 / 5	1 / 0	1 / 1	2 / 2	1 / 0	0 / 1	1 / 0	63 / 87	1 / 0	1 / 0
New Age	9 / 2	7 / 6	9 / 0	7 / 4	10 / 10	9 / 2	7 / 6	3 / 3	15 / 53	10 / 7
Pop	6 / 2	9 / 1	10 / 2	9 / 2	5 / 3	9 / 2	5 / 2	2 / 0	7 / 1	19 / 73
R&B	8 / 2	0 / 1	16 / 3	8 / 4	2 / 0	5 / 3	5 / 0	1 / 0	3 / 0	7 / 10
Hip-Hop	8 / 2	0 / 0	2 / 1	8 / 2	0 / 1	0 / 1	1 / 0	4 / 3	2 / 0	4 / 1
Rock	17 / 15	1 / 2	6 / 8	4 / 7	10 / 5	2 / 4	7 / 1	12 / 13	4 / 1	9 / 7

Table 7.2: Confusion matrix showing results derived from AB acoustic-based classifier/BoW+SEM text-based approach.

taking advantage of Tagme[40] Ferragina & Scaiella (2012), an EL system that matches entity candidates with Wikipedia links, and then performs disambiguation exploiting both the in-link graph and the Wikipedia page dataset. TagMe provides for each detected entity, its Wikipedia page id and Wikipedia categories.

7.4 Music Genre Classification

7.4.1 Dataset Description

Starting from MARD, our purpose is to create a subset suitable for genre classification, including 100 albums per genre class. We enforce these albums to be authored by different artists, and that review texts and audio descriptors of their songs are available in MARD. Then, for every album, we selected audio descriptors of the first song of each album as representative sample of the album. From the original 16 genres, 3 of them did not have enough instances complying with these prerequisites (Reggae, Blues and Gospel). This results in a classification dataset composed of 1,300 albums, divided in 13 different genres, with around 1,000 characters of review per album.

[40]http://tagme.di.unipi.it/

7.4.2 Features

Textual Surface Features

We used a standard Vector Space Model representation of documents, where documents are represented as bag-of-words (BoW) after tokenizing and stop-word removal. All words and bigrams (sequences of two words) are weighted according to *tf-idf* measure.

Semantic Features

We enriched the initial BoW vectors with semantic information thanks to the EL step. Specifically, for each named entity disambiguated with Tagme, its Wikipedia ID and its associated categories are added to the feature vector, also with *tf-idf* weighting. Wikipedia categories are organized in a taxonomy, so we enriched the vectors by adding one level more of broader categories to the ones provided by Tagme. Broader categories were obtained by querying DBpedia[41].

Sentiment Features

Based on those aspects and associated polarity extracted with the opinion mining framework, with an average number of aspects per review around 37, we follow Montero et al. (2014) and implement a set of sentiment features, namely:

- Positive to All Emotion Ratio: fraction of all sentimental features which are identified as positive (sentiment score greater than 0).
- Document Emotion Ratio: fraction of total words with sentiments attached. This feature captures the degree of affectivity of a document regardless of its polarity.
- Emotion Strength: This document-level feature is computed by averaging sentiment scores over all aspects in the document.
- F-Score[42]: This feature has proven useful for describing the contextuality/formality of language. It takes into consideration the presence of *a priori* "descriptive" POS tags (nouns and adjectives), as opposed to "action" ones such as verbs or adverbs.

[41] `http://dbpedia.org`

[42] Not to be confused with the evaluation metric.

	BoW	BoW+SEM	BoW+SENT
Linear SVM	**0.629**	**0.691**	**0.634**
Ridge Classifier	0.627	0.689	0.61
Random Forest	0.537	0.6	0.521

Table 7.3: Accuracy of the different classifiers

Acoustic Features

Acoustic features are obtained from AB. They are computed using Essentia[43]. These encompass loudness, dynamics, spectral shape of the signal, as well as additional descriptors such as time-domain, rhythm, and tone Porter et al. (2015).

7.4.3 Baseline approaches

Two baseline systems are implemented. First, we implement the text-based approach described in Hu et al. (2005) for music review genre classification. In this work, a Naïve Bayes classifier is trained on a collection of 1,000 review texts, and after preprocessing (tokenisation and stemming), BoW features based on document frequencies are generated. The second baseline is computed using the AB framework for song classification Porter et al. (2015). Here, genre classification is computed using multi-class support vector machines (SVMs) with a one-vs.-one voting strategy. The classifier is trained with the set of low-level features present in AB.

7.4.4 Experiments

We tested several classifiers typically used for text classification, namely Linear SVM, Ridge Classifier and Nearest Centroid, using the implementations provided by the scikit-learn library[44]. Among them, Linear SVM has shown better performance when combining different feature sets (see Table 7.3). Therefore, we trained a Linear SVM classifier with L2 penalty over different subsets of the features described in Section 7.4.2, which are combined via linear aggregation. Specifically, we combine the different feature sets into five systems, namely **BoW** (BoW), **BoW+Semantic** without broader categories (BoW+SEM), **BoW+Semantic Broader** with broader categories (BoW+SEMb), **BoW+Sentiment** (BoW+SENT) and **BoW+Semantic+Sentiment** (BoW+SEM+SENT). In this way, we aim at understanding the extent to which sentiment and semantic features (and their interaction) may contribute to the review genre classification task. Note that this paper is focused on the influence of textual features in genre classification, and classification based on acoustic features is simply used as a baseline for comparison. A proper combination of acoustic

[43] `http://essentia.upf.edu/`
[44] http://scikit-learn.org/

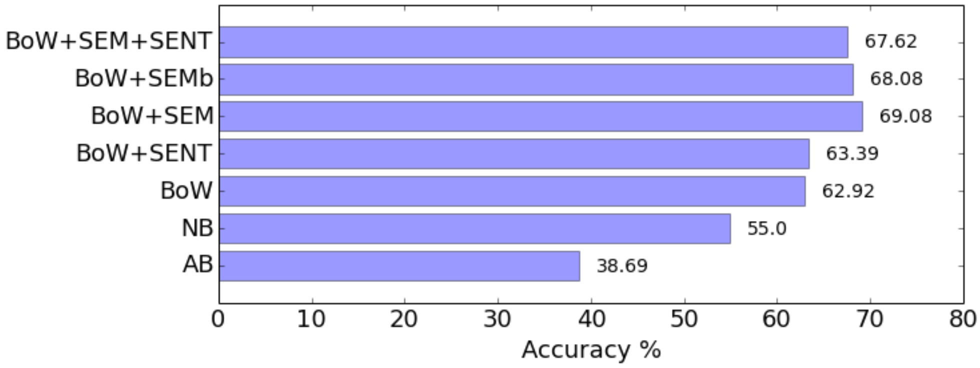


Figure 7.3: Percentage of accuracy of the different approaches. AB refers to the AcousticBrainz framework. NB refers to the method based on Naïve Bayes from Hu et al. (2005).

and textual features in text classification is a challenging problem and would require a deeper study that is out of the scope of this paper. The dataset is split 80-20% for training and testing, and accuracy values are obtained after 5-fold cross validation.

7.4.5 Results and Discussion

Accuracy results of the two baseline approaches introduced in Section 7.4.3 along with our approach variants are shown in Figure 7.3. At first sight, we may conclude that sentiment features contribute to slightly outperforming purely text-based approaches. This result implies that affective language present in a music review is not a salient feature for genre classification (at least with the technology we applied), although it certainly helps. On the contrary, semantic features clearly boost pure text-based features, achieving 69.08% of accuracy. The inclusion of broader categories does not improve the results in the semantic approach. The combination of semantic and sentiment features improves the BoW approach, but the achieved accuracy is slightly lower than using semantic features only.

Let us review the results obtained with baseline systems. The Naïve Bayes approach from Hu et al. (2005) is reported to achieve an accuracy of 78%, while in our results it is below 55%. The difference in accuracy may be due to the substantial difference in length of the review texts. In Hu et al. (2005), review texts were at least 3,000 characters long, much larger that ours. Moreover, the addition of a distinction between Classic Rock and Alternative Rock is penalizing our results. As for the acoustic-based approach, although the obtained accuracy may seem low, it is in fact a good result for purely audio-based genre classification, given the high number of classes and the absence of artist bias in the dataset Bogdanov et al. (2016). Finally, we refer to Table 7.2 to highlight the fact that the text-based approach clearly outperforms the acoustic-based classifier, although in general both show a similar behaviour across genres. Also, note the low accuracy for both Classic Rock and Alternative Rock, which suggests that their difference is subtle enough for making it a hard problem for automatic classification.

7.5 Conclusions and Future Work

In this work we have presented MARD, a multimodal dataset of album customer reviews combining text, metadata and acoustic features gathered from Amazon, MB and AB respectively. Customer review texts are further enriched with named entity disambiguation along with polarity information derived from aspect-based sentiment analysis. Based on this information, a text-based genre classifier is trained using different combinations of features. A comparative evaluation of features suggests that a combination of bag-of-words and semantic information has higher discriminative power, outperforming competing systems in terms of accuracy.

CHAPTER 8

Applications in Musicology

8.1 Introduction

8.2 Building Culture-specific Knowledge Bases: The Flamenco Case

8.2.1 Introduction

Music context information is now playing a key role in MIR research. Multimodal approaches, semantic approaches, and text-IR approaches have shown important achievements in typical MIR problems, such as music recommendation and discovery, genre classification, or music similarity Schedl et al. (2014). Therefore, collecting and storing music context information may be extremely useful for the MIR research community Oramas et al. (2014). There are some broad repositories of music context information such as MusicBrainz[45] or Discogs[46]. Although some of these repositories are very complete and accurate, there is still a vast amount of music information out there, which is generally scattered among different sources on the Web. Hence, harvesting and combining that information is a crucial step in the creation of practical and meaningful music knowledge bases. In addition, the creation of genre-specific knowledge bases may be very valuable for research and dissemination purposes, and particularly to non-western music traditions.

In this paper, we propose a methodology for the creation of a genre-specific knowledge base; in particular, a knowledge base of flamenco music. The proposed methodology combines content curation and knowledge extraction processes. First, an important amount of information is gathered from different data sources, which are subsequently combined by applying pair-wise entity resolution. Next, new knowledge is extracted from unstructured harvested

[45] http://musicbrainz.org
[46] http://www.discogs.com/

texts and employed to populate the knowledge base. For this purpose, an entity linking system has been expressly developed. Finally, the content of the knowledge base is used to compute artist relevance and results are evaluated according to flamenco experts criteria. The content of the knowledge base is freely available and downloadable as data dumps in RDF and JSON formats.

The remainder of the paper is organized as follows. In Section 8.2.2, an introduction to flamenco music is presented. In Section **??** some relevant prior work is briefly surveyed. Section 8.2.3 describes the structure of the knowledge base. Next, in Section 8.2.4 the process of content curation is explained. Section 8.2.5 shows the methodology applied for knowledge extraction. In Section **??** artist relevance is computed and some statistics about the content are laid out. Finally, Section 12 concludes the paper and points out for future lines of work.

8.2.2 Flamenco music

Several musical traditions contributed to the genesis of flamenco music as we know it today. Among them, the influences of the Jews, Arabs, and Spanish folk music are recognizable, but indubitably the imprint of Andalusian Gypsies' culture is deeply ingrained in flamenco music. Flamenco occurs in a wide range of settings, including festive *juergas* (private parties), *tablaos* (flamenco venues), concerts, and big productions in theaters. In all these settings we find the main components of flamenco music: *cante* or singing, *toque* or guitar playing, and *baile* or dance. According to Gamboa Gamboa (2005), flamenco music grew out of the singing tradition, as a melting process of all the traditions mentioned above, and therefore the role of the singer soon became dominant and fundamental. *Toque* is subordinated to *cante*, especially in more traditional settings, whereas *baile* enjoys more independence from voice.

In the flamenco jargon styles are called *palos*. Criteria adopted to define flamenco *palos* are rhythmic patterns, chord progressions, lyrics and its poetic structure, and geographical origin. In flamenco geographical variation is important to classify *cantes* as often they are associated to a particular region where they were originated or where they are performed with gusto. Rhythm or *compás* is a unique feature of flamenco. Rhythmic patterns based on 12-beat cycles are mainly used. Those patterns can be classed as follows: binary patterns, such as *tangos* or *tientos*; ternary patterns, which are the most common ones, such as *fandangos* or *bulerías*; mixed patterns, where ternary and binary patterns alternate, such as *guajira*; free-form, where there is no a clear underlying rhythm, such as *tonás*. For further information on fundamental aspects of flamenco music, see the book of Fernández Fernández (2004). For a comprehensive study of styles, musical forms and history of flamenco the reader is referred to the books of Blas Vega and Ríos Ruiz Blas Vega & Ríos Ruiz (1988), Navarro and Ropero Navarro & Ropero (1995), and Gamboa Gamboa

(2005) and the references therein.

8.2.3 FlaBase

FlaBase (Flamenco Knowledge Base) is the acronym of a new knowledge base of flamenco music. Its ultimate aim is to gather all available online editorial, biographical and musicological information related to flamenco music. A first version is just being released. Its content is the result of the curation and extraction processes explained in Sections 8.2.4 and 8.2.5. FlaBase is stored in RDF and JSON formats, and it is freely available for download[47]. Its RDF version follows the Linked Open Data principles, and it might be queried by setting up a SPARQL endpoint. A JSON version is also available, thus facilitating the use of the content by all the community of researchers and developers. This first release of FlaBase contains information about 1,174 artists, 76 *palos* (flamenco genres), 2,913 albums, 14,078 tracks, and 771 Andalusian locations.

Ontology Definition

The FlaBase data structure is defined in an ontology schema. One of the advantages of using an ontology as a schema is that it can be easily modified. Thus, our design is a first building block that can be enhanced and redefined in the future. The initial ontology is structured around five main classes: MusicArtist, Album, Track, Palo and Place, and three domain specific classes: *cantaor* (flamenco singer), guitarist (flamenco guitar player), and *bailaor* (flamenco dancer). These three classes were defined because they are the most frequent types of artists in the data. Other instrument players may be instantiated directly from the MusicArtist class. We have tried to reuse as much vocabulary as we could. We re-utilized most of the classes and some properties from the Music Ontology[48], a standard model for publishing music-related data. We selected the classes according to the ones used by the LinkedBrainz project[49], which maps concepts from MusicBrainz to Music Ontology.

8.2.4 Content Curation

The first step towards building a domain-specific knowledge base is to gather all possible content from available data sources. This implies at least two problems, namely, the selection of sources, and the matching between entities from different sources. In what follows we enumerate the involved data sources and describe the methodology applied to entity resolution.

[47] `http://mtg.upf.edu/download/datasets/flabase`
[48] `http://musicontology.com`
[49] `https://wiki.musicbrainz.org/LinkedBrainz`

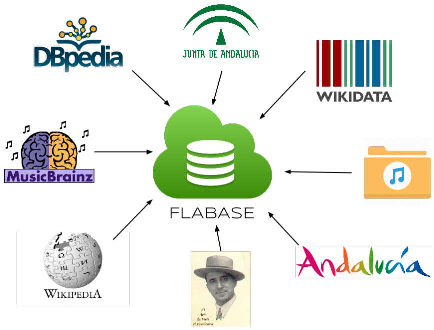


Figure 8.1: Selected data sources

Data Acquisition

Our aim is to gather an important amount of information about musical entities, including textual descriptions and available metadata. A schema of the selected data sources is shown in Figure 8.1. We started by looking at Wikipedia[50], the free and multilingual Internet encyclopedia. It is the Internet's largest and most popular general reference work. Each Wikipedia article may have a set of associated categories. Categories are intended to group together pages on similar subjects and are structured in a taxonomical way. To find Wikipedia articles related to flamenco music, we first looked for flamenco categories. The taxonomy of categories can be explored by querying DBpedia, a knowledge base with structured content extracted from Wikipedia. In particularly, we employed the SPARQL endpoint of the Spanish DBpedia[51]. We queried for categories related to the flamenco category in the taxonomy. At the end, we obtained 17 different categories (e.g., *cantaores de flamenco, guitarristas de flamenco*).

By querying again DBpedia, we gathered all DBpedia resources related to one of these categories. We obtained a total number of 438 resources in Spanish, of which 281 were also in English. Each DBpedia resource is associated with a Wikipedia article. Text and HTML code were then extracted from Wikipedia articles in English and Spanish by using the WikiMedia API. Next, we classified the extracted articles according to the ontology schema defined in our knowledge base (Section 8.2.3). For this purpose, we exploited classification information provided by DBpedia (DBpedia ontology and Wikipedia categories). At the end, from all gathered resources, we only kept those related to artists and *palos*, totalling 291 artists and 56 *palos*.

However, the amount of information present in Wikipedia related to flamenco music is somewhat scarce. Therefore, we decided to expand our knowledge base with information from two different websites. First, *Andalucia.org*, the

[50] `http://www.wikipedia.org`
[51] `http://es.dbpedia.org`

touristic web from the Andalusia Government[52]. It contains 422 artist biographies in English and Spanish, and the description of 76 *palos* also in both languages. Second, a website called *El arte de vivir el flamenco*[53], which includes 749 artist biographies among *cantaores*, *bailaores* and guitarists. Both webs were crawled and their content stored in our knowledge base.

MusicBrainz is one of the biggest and more reliable open music databases, which provides an unambiguous form of music identification. Therefore, we turned to it in order to fill our knowledge base with information about flamenco album releases and recordings. Artists present in FlaBase were intended to be mapped with MusicBrainz artists. For every match, all content related to releases and recordings was gathered. After doing so, we obtained a total number of 814 releases and 9,942 recordings.

The information gathered from MusicBrainz is a little part of the actual flamenco discography. Therefore, to complement it we used a flamenco recordings database gathered by Rafael Infante and available at CICA website[54] (Computing and Scientific Center of Andalusia). This database has information about releases from the early time of recordings until present time, counting 2,099 releases and 4,136 songs. For every song entry, a *cantaor* name is provided, and most of the times also guitarist and *palo*, which is a very valuable information to define flamenco recordings.

Finally, we supplied our knowledge base with information related to Andalusian towns and provinces. We gathered this information from the official database SIMA[55] (Multi-territorial System of Information of Andalusia).

Entity Resolution

Entity Resolution (ER) is the problem of extracting, matching and resolving entity mentions in structured and unstructured data Getoor (2012). There are several approaches to tackle the ER problem. For the scope of this research, we selected a pair-wise classification approach based on string similarity between entity labels.

The first issue after gathering the data is to decide whether two entities from different sources are referring to the same one. Therefore, given two sets of entities A and B, the objective is to define an injective and non-surjective mapping function f between A and B that decides whether an entity $a \in A$ is the same as an entity $b \in B$. To do that, a string similarity metric $sim(a, b)$ based on the Ratcliff-Obershelp algorithm Ratcliff & Metzener (1988) has been defined. It measures the similarity between two entity labels and outputs a value between 0 and 1. We consider that a and b are the same entity if their

[52] http://andalucia.org

[53] http://www.elartedevivirelflamenco.com/

[54] http://flun.cica.es/index.php/grabaciones

[55] http://www.juntadeandalucia.es/institutodeestadisticaycartografia/sima

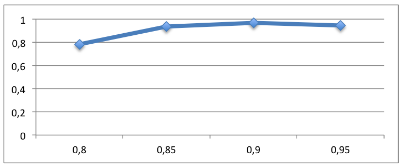


Figure 8.2: F-measure for different values of θ

similarity is bigger than a parameter θ. If there are two entities $b, c \in B$ that satisfy that $sim(a, b) \geq \theta$ and $sim(a, c) \geq \theta$, we consider only the mapping with the highest score. To determine the value of θ, we tested the method with several θ values over an annotated dataset of entity pairs. To create this dataset, the 291 artists gathered from Wikipedia were manually mapped to the 422 artists gathered from Andalucia.org, obtaining a total amount of 120 pair matches. As it is shown in Figure 8.2 the best F-measure (0,97) was obtained with $\theta = 0.9$. Finally, we applied the described method with $\theta = 0.9$ to all gathered entities from the three data sources. Thanks to the entity resolution process, we reduced the initial set of 1,462 artists and 132 *palos* to a set of 1,174 artists and 76 *palos*.

Once we had our artist entities resolved, we began to gather their related discographic information. First, we tried to find out the MusicBrainz ID of the gathered artists. Depending on the information about the entity, two different process were applied. First, every Wikipedia page, and its equivalent DBpedia resource, has a correspondent entity defined in Wikidata. Wikidata is a free linked database which acts as a structured data storage of Wikipedia. There are several properties in Wikidata that may link Wikidata items with MusicBrainz items. Thus, the equivalent Wikidata resource of a Wikipedia artist page may have a link to its corresponding MusicBrainz artist ID. Therefore, we looked for these relations and mapped all possible entities. For those artists without a direct link to MusicBrainz, we queried the MusicBrainz API by using the artist labels, and then applied our entity resolution method to the obtained results.

Finally, to integrate the discography database of CICA into our knowledge base, we applied the entity resolution method to the fields *cantaor*, guitarist and *palo* of each recording entry in the database. From the set of 202 *cantaores* and 157 guitarists names present in the recording entries, a total number of 78 *cantaores* and 44 guitarists were mapped to our knowledge base. The number of mapped artists was low due to differences between the way of labeling an artist. An artist name may be written using one or two surnames, or using a nickname. In the case of *palos*, there were 162 different *palos* in the database, 54 of which were mapped with the 76 of our knowledge base. These 54 *palos* correspond to an 80% of *palo* assignments present in the recording entries.

8.2.5 Knowledge Extraction

Once the process of data acquisition is finished, the knowledge base is ready for use. However, there is an important amount of knowledge present in the data that has not been fully exploited. Texts gathered contain a huge epistemic potential that remains implicit. Consequently, to enhance the amount of structured data in FlaBase, a process of knowledge extraction has been carried out. This implicit knowledge may vary from biographical data, such as place and date of birth, to more complex semantic relations involving different entities. Three tasks play a key role in the process of knowledge extraction from non-structured text: named entity recognition (NER), named entity disambiguation (NED), and relation extraction (RE) Usbeck et al. (2014b). In this research, we focus on the two first tasks. In what follows, a system for entity recognition and disambiguation is described and evaluated. Lastly, an information extraction process is applied to populate the knowledge base.

Named entity recognition and disambiguation

To extract implicit knowledge from a text, the first step is to semantically annotate it by identifying entity mentions. Named entity recognition is a task that seeks and classify words in text into pre-defined categories (e.g., person, organization, or place). Named entity disambiguation, also called entity linking, aims to determine what is actually a named entity present in a text. It generally does so by identifying it in a knowledge base of reference. NED can be addressed directly from the text, or applied to the output of a NER system. We propose a method that employs a combination of both approaches, depending on the category of the entity. For NER, we used the Stanford NER system Finkel et al. (2005), implemented in the library Stanford Core NLP[56] and trained on Spanish texts. For NED we tried two different approaches. First, we looked for exact string matches between FlaBase entity labels and word n-grams extracted from the text. Second, we searched for exact string matches between FlaBase entity labels and the output of the NER system. In fact, we tried several combinations of both approaches until we obtained the most satisfactory one.

For the scope of this research, we focused on Spanish texts, as flamenco texts are mostly written in Spanish. Although there are many entity linking tools available, we decided to develop ours because state-of-the-art systems (e.g., Tag-me or Babelfy) are well-tuned for English texts, but do not perform well on Spanish texts, and even less with music texts of a specific domain. In addition, we wanted to have a system able to map entities to our knowledge base. Therefore, we developed a system able to detect and disambiguate three categories of entities: person, *palo* and location. Three different approaches were defined by combining NER and NED in different ways according to the cat-

[56] `http://nlp.stanford.edu/software/corenlp.shtml`

Approach	Precision	Recall	F-measure
1) NED	0.829	**0.694**	0.756
2) NED + NER to PERS & LOC	0.739	0.347	0.472
3) NED + NER to LOC	**0.892**	0.674	**0.767**

Table 8.1: Precision, Recall and F-measure of NER+NED

egory. First, directly applying NED to text. Second, disambiguating location and person entities from the NER output, and *palo* directly from text. Third, only disambiguating location entities from the NER output, and location and *palo* directly from text.

To determine which approach performs better, three artist biographies coming from three different data sources were manually annotated, having a total number of 49 annotated entities. We followed an evaluation methodology similar to the one used in KBP2014 Entity Linking Task[57]. Results on the different approaches are shown in Table 8.1. We observe that applying NER to entities of the person category before NED worsens performance significantly, as recall suddenly decrease by half. After manually analysing false negatives, we observed that this is caused because many artist names have definite articles between name and surname (e.g., *de, del*), and this is not recognized by the NER system. In addition, many artists have a nickname that is not interpreted as a person entity by the NER system. The best approach is the third (NED + NER to LOC), which is slightly better than the first (only NED) in terms of precision. This is due to the fact that many artists have a town name as a surname or as part of his nickname. Therefore, applying NED directly to text is misclassifying person entities as location entities. Thus, by adding a previous step of NER to location entities we have increased overall performance, as it can be seen on the F-measure values.

Knowledge base population

Biographical texts coming from different data sources have been stored in FlaBase. These texts are full of relevant information about FlaBase entities, but in an unstructured way. Thus, a process of information extraction is necessary to transform the unstructured information into structured knowledge. For the scope of this research, we focused on extracting two specific data: birth year and birth place, as they can be very relevant for anthropologic studies. We observed that this information is often in the very first sentences of the artist biographies, and always near the word *nació* (Spanish translation of "was born"). Therefore, to extract this information, we looked for this word in the first 250 characters of every biographical text. If it is found, we apply our entity linking method to this piece of text. If a location entity is found near the

[57] `http://nlp.cs.rpi.edu/kbp/2014/`

word "nació", we assume that this entity is the place of birth of the biography subject. In addition, by using regular expressions, we look for the presence of a year expression in the neighborhood. If it is found, we assume it as the year of birth. If more than one year is found, we select the one with the smaller value.

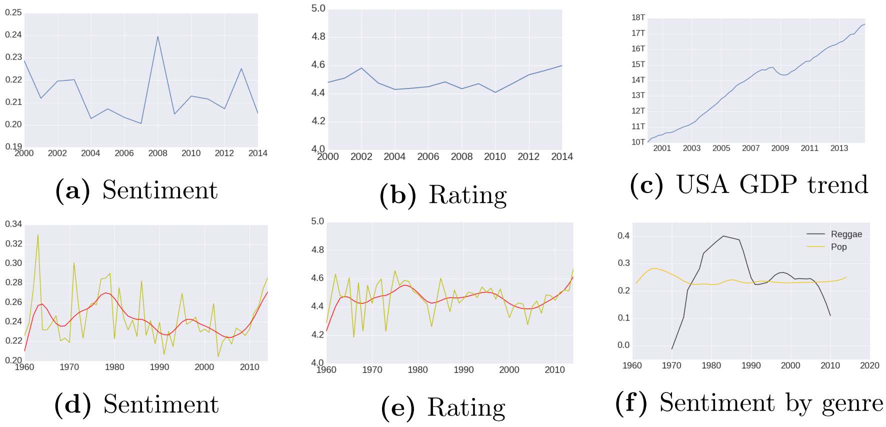
To evaluate our approach, we tested the extraction of birth places in all texts coming from the web Andalucia.org (442 artists). We chose this subset because Andalucia.org also provides specific information about artist origin that had been previously crawled and stored in FlaBase. However, we observed that in many occasions the artist origin provided by the data source was wrong. Therefore, we decided to manually annotate the province of precedence of these 442 artists for building ground truth data. After the application of the extraction process on the annotated test set, we obtained a precision value of 0,922 and a recall of 0,648. Therefore, we can state that our method is extracting biographic information with very high precision and quite reasonable recall. We finally applied the extraction process to all artist entities with biographical texts coming from any of the two flamenco crawled websites. Thus, from a total number of 1,123 artists coming from these data sources (95% of the artists in the knowledge base), 743 birth places and 879 birth years were extracted.

8.3 Exploring Music Digital Libraries

8.4 Diachronic Study of Music Criticism

We carried out a study of the evolution of music criticism from two different temporal standpoints. Specifically, we consider when the review was written and, in addition, when the album was first published. Since we have sentiment information available for each review, we first computed an average sentiment score for each year of review publication (between 2000 and 2014). In this way, we may detect any significant fluctuation in the evolution of affective language during the 21st century. Then, we also calculated the average sentiment for each review by year of album publication. This information is obtained from MB and complemented with the average of the Amazon rating scores.

In what follows, we show visualizations for sentiment scores and correlation with ratings given by Amazon users, according to these two different temporal dimensions. Although arriving to musicological conclusions is out of the scope of this paper, we provide *food for thought* and present the readers with hypotheses that may explain some of the facts revealed by these data-driven trends.



(a) Sentiment **(b)** Rating **(c)** USA GDP trend

(d) Sentiment **(e)** Rating **(f)** Sentiment by genre

Figure 8.3: Sentiment and rating averages by review publication year (a and b); GDP trend in USA from 2000 to 2014 (c), and sentiment and rating averages by album publication year (d, e and f)

8.4.1 Evolution by Review Publication Year

We applied sentiment and rating average calculations to the whole MARD dataset, grouping album reviews by year of publication of the review. Figure 8.3a shows the average of the sentiment scores associated to every aspect identified by the sentiment analysis framework in all the reviews published in a specific year, whilst Figure 8.3b shows average review ratings per year. At first sight, we do not observe any correlation between the trends illustrated in the figures. However, the sentiment curve (Figure 8.3a) shows a remarkable peak in 2008, a slightly lower one in 2013, and a low between 2003 and 2007, and also between 2009 and 2012. It is not trivial to give a proper explanation of this variations on the average sentiment. We speculate that these curve fluctuations may suggest some influence of economical or geopolitical circumstances in the language used in the reviews, such as the 2008 election of Barack Obama as president of the US. As stated by the political scientist Dominique Moïsi in Moïsi (2010):

> In November 2008, at least for a time, hope prevailed over fear. The wall of racial prejudice fell as surely as the wall of oppression had fallen in Berlin twenty years earlier [...] Yet the emotional dimension of this election and the sense of pride it created in many Americans must not be underestimated.

Another factor that might be related to the positiveness in use of language is the economical situation. After several years of continuous economic growth, in 2007 a global economic crisis started[58], whose consequences were visible

[58] `https://research.stlouisfed.org`

in the society after 2008 (see Figure 8.3c). In any case, further study of the different implied variables is necessary to reinforce any of these hypotheses.

.

8.4.2 Evolution by Album Publication Year

In this case, we study the evolution of the polarity of language by grouping reviews according to the album publication date. This date was gathered from MB, meaning that this study is conducted on the 42,1% of the MARD that was successfully mapped. We compared again the evolution of the average sentiment polarity (Figure 8.3d) with the evolution of the average rating (Figure 8.3e). Contrary to the results observed by review publication year, here we observe a strong correlation between ratings and sentiment polarity. To corroborate that, we computed first a smoothed version of the average graphs, by applying 1-D convolution (see line in red in Figures 8.3d and 8.3e). Then we computed Pearson's correlation between smoothed curves, obtaining a correlation $r = 0.75$, and a p-value $p \ll 0.001$. This means that in fact there is a strong correlation between the polarity identified by the sentiment analysis framework in the review texts, and the rating scores provided by the users. This correlation reinforces the conclusions that may be drawn from the sentiment analysis data.

To further dig into the utility of this polarity measure for studying genre evolution, we also computed the smoothed curve of the average sentiment by genre, and illustrate it with two idiosyncratic genres, namely *Pop* and *Reggae* (see Figure 8.3f). We observe in the case of *Reggae* that there is a time period where reviews have a substantial use of a more positive language between the second half of the 70s and the first half of the 80s, an epoch which is often called the golden age of *Reggae* Alleyne & Dunbar (2012). This might be related to the publication of Bob Marley albums, one of the most influential artists in this genre, and the worldwide spread popularity of reggae music. In the case of *Pop*, we observe a more constant sentiment average. However, in the 60s and the beginning of 70s there are higher values, probably consequence by the release of albums by The Beatles. These results show that the use of sentiment analysis on music reviews over certain timelines may be useful to study genre evolution and identify influential events.

8.5 Conclusions

Our diachronic study of the sentiment polarity expressed in customer reviews explores two interesting ideas. First, the analysis of reviews classified by year of review publication suggests that geopolitical events or macro-economical circumstances may influence the way people speak about music. Second, an analysis of the reviews classified by year of album publication is presented. The results show how sentiment analysis can be very useful to study the evolution of music genres. The correlation observed between average rating and sentiment scores suggest the suitability of the proposed sentiment-based approach to predict user satisfaction with musical products. Moreover, according to the observed trend curves, we can state that we are now in one of the best periods of the recent history of music. Further work is necessary to elaborate on these hypotheses. In addition, the combination of audio and textual features is still an open problem, not only for classification but also for the study of the evolution of music. We expect the released dataset will be explored in multiple ways for the development of multimodal research approaches in MIR. In conclusion, the main contribution of this work is a demonstration of the utility of applying systematic linguistic processing on texts about music. Furthermore, we foresee our method to be of interest for musicologists, sociologists and humanities researchers in general.

Part II

Multimodal Deep Learning Approaches

CHAPTER 9

Framework for Multimodal Deep Learning

9.1 Introduction

CHAPTER 10

Cold-start Music Recommendation

10.1 Introduction

CHAPTER 11

Multimodal Music Genre Classification

11.1 Introduction

CHAPTER 12

Summary and future perspectives

12.1 Introduction

In this thesis we have described a number of computational approaches for helping the users of online sharing platforms to better annotate the content they generate. Our approaches are meant to be a step towards increasing the value of resources shared in online sharing platforms by improving their descriptions and enabling better organisation, browsing and searching functionalities.

12.2 Summary of contributions

12.3 Directions for future research

Sergio Oramas Martin, Barcelona, 11 March 2017.

Bibliography

Adomavicius, G. & Kwon, Y. (2012). Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. *{IEEE} Trans. Knowl. Data Eng.*, *24*(5), 896–911.

Aghdam, M. H., Hariri, N., Mobasher, B., & Burke, R. D. (2015). Adapting Recommendations to Contextual Changes Using Hierarchical Hidden Markov Models. In *Proceedings of the 9th {ACM} Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015*, pp. 241–244.

Alcorta, C. S., Sosis, R., & Finkel, D. (2008). Ritual harmony: Toward an evolutionary theory of music. *Behavioral and Brain Sciences*, *31*(5), 576–+.

Alleyne, M. R. & Dunbar, S. (2012). *The Encyclopedia of Reggae: The Golden Age of Roots Reggae.*

Anand, S. S., Kearney, P., & Shapcott, M. (2007). Generating semantically enriched user profiles for Web personalization. *ACM Trans. Internet Technol.*, *7*(4).

Bach, N. & Badaskar, S. (2007). A Review of Relation Extraction. *Literature review for Language and Statistics II.*

Ballesteros, M. & Nivre, J. (2013). Going to the Roots of Dependency Parsing. *Computational Linguistics*, *39*(1), 5–13.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007a). Open Information Extraction for the Web. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 7, pp. 2670–2676.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007b). Open Information Extraction from the Web. In *International Joint Conferences on Artificial Intelligence*, pp. 2670–2676.

Bansal, T., Belanger, D., & McCallum, A. (2016). Ask the GRU: Multi-task Learning for Deep Text Recommendations. *RecSys*, pp. 107–114.

Baral, C. & De Giacomo, G. (2015). Knowledge Representation and Reasoning: What's Hot. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 4316–4317.

Beliakov, G., Calvo, T., & James, S. (2015). *Recommender Systems Handbook*, chap. Aggregation Functions for Recommender Systems, pp. 777–808. Boston, MA: Springer US.

Bellog\'\in, A., Cantador, I., & Castells, P. (2010). A Study of Heterogeneity in Recommendations for a Social Music Service. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '10, pp. 1–8. ACM.

Blas Vega, J. & Ríos Ruiz, M. (1988). *Diccionario enciclopédico ilustrado del flamenco.* Madrid: Cinterco.

Bogdanov, D., Porter, A., Herrera, P., & Serra, X. (2016). Cross-collection evaluation for music classification tasks. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR).*

Bogdanov, D., Wack, N., & Others (2013). ESSENTIA: an Open-Source Library for Sound and Music Analysis. In *ACM International Conference on Multimedia (MM{\textquoteright}13)*, pp. 855–858.

Bohnet, B. (2010). Very High Accuracy and Fast Dependency Parsing is Not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pp. 89–97. Stroudsburg, PA, USA: Association for Computational Linguistics.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pp. 1247–1250. New York, NY, USA: ACM.

Bonnin, G. & Jannach, D. (2014). Automated Generation of Music Playlists: Survey and Experiments. *ACM Comput. Surv.*, *47*(2), 26:1—-26:35.

Bouayad-Agha, N., Burga, A., Casamayor, G., Codina, J., Nazar, R., & Wanner, L. (2014). An Exercise in Reuse of Resources: Adapting General Discourse Coreference Resolution for Detecting Lexical Chains in Patent Documentation. In *Proccedings of the Language Resources and Evaluation 775 Conference (LREC)*, pp. 3214–3221.

Bovi, C. D., Espinosa-Anke, L., & Navigli, R. (2015a). Knowledge Base Unification via Sense Embeddings and Disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 726–736.

Bovi, C. D., Telesca, L., & Navigli, R. (2015b). Large-Scale Information Extraction from Textual Definitions through Deep Syntactic and Semantic

Analysis. *Transactions of the Association for Computational Linguistics (TACL)*, *3*, 529–543.

Bunescu, R. & Pasca, M. (2006). Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proceesings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, pp. 9–16. Trento, Italy.

Bunescu, R. C. & Mooney, R. J. (2005). A Shortest Path Dependency Kernel for Relation Extraction. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pp. 724–731.

Bunke, H. & Shearer, K. (1998). A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, *19*(3-4), 255–259.

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, *12*(4), 331–370.

Cambria, E. & White, B. (2014). Jumping NLP Curves: A Review of Natural Language Processing Research. *Computational Intelligence Magazine, IEEE*, *9*(2), 48–57.

Cantador, I., Bellogín, A., & Castells, P. (2008). A multilayer ontology-based hybrid recommendation model. *AI Commun. Special Issue on Rec. Sys.*, *21*(2-3), 203–210.

Carlson, A., Betteridge, J., Wang, R. C., Hruschka Jr, E., & Mitchell, T. M. (2010). Coupled Semi-Supervised Learning for Information Extraction. In *Proceedings of the third ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 101–110.

Celma, Ò., Cano, P., & Herrera, P. (2006). Search Sounds An audio crawler focused on weblogs. In *7th International Conference on Music Information Retrieval (ISMIR)*.

Celma, Ò. & Herrera, P. (2008). A new approach to evaluating novel recommendations. In *Proceedings of the 2008 ACM conference on Recommender systems*, pp. 179–186. ACM.

Celma, Ò. & Serra, X. (2008). FOAFing the music: Bridging the semantic gap in music recommendation. *Web Semantics*, *6*, 250–256.

Chim, H. & Deng, X. (2008). Efficient phrase-based document similarity for clustering. *Knowledge and Data Engineering, IEEE Transactions on*, *20*(9), 1217–1229.

Choi, K., Fazekas, G., & Sandler, M. (2016a). Automatic tagging using deep convolutional neural networks. *International Society for Music Information Retrieval Conference*, pp. 805–811.

Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2016b). Convolutional Recurrent Neural Networks for Music Classification. *arXiv preprint arXiv:1609.04243*.

Choi, K., Lee, J. H., & Downie, J. S. (2014). What is this song about anyway?: Automatic classification of subject using user interpretations and lyrics. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, pp. 453–454.

Cohen, W. W. & Fan, W. (2000). Web-collaborative filtering: recommending music by crawling the Web. *Computer Networks*, *33*, 685–698.

Cornolti, M., Informatica, D., Ferragina, P., Informatica, D., & Ciaramita, M. (). No Title. *Proceedings of the International World Wide Web Conference (WWW) (Practice & Experience Track), ACM (2013)*.

Culotta, A. & Sorensen, J. (2004). Dependency Tree Kernels for Relation Extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *JAsIs*, *41*(6), 391–407.

Di Noia, T., Mirizzi, R., Ostuni, V. C., & Romito, D. (2012a). Exploiting the Web of Data in Model-based Recommender Systems. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pp. 253–256. New York, NY, USA: ACM.

Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D., & Zanker, M. (2012b). Linked open data to support content-based recommender systems. In *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS '12, pp. 1–8. New York, NY, USA: ACM.

Dieleman, S., Brakel, P., & Schrauwen, B. (2011). Audio-based music classification with a pretrained convolutional network. In *12th International Society for Music Information Retrieval Conference (ISMIR-2011)*, pp. 669–674. University of Miami.

Dieleman, S. & Schrauwen, B. (2014). End-to-end learning for music audio. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pp. 6964–6968. IEEE.

Dong, R., O'Mahony, M. P., & Smyth, B. (2014). Further Experiments in Opinionated Product Recommendation. In *ICCBR'14*, pp. 110–124. Cork, Ireland.

Dong, R., Schaal, M., O'Mahony, M. P., & Smyth, B. (2013). Topic Extraction from Online Reviews for Classification and Recommendation. IJCAI'13, pp. 1310–1316.

Downie, J. S. & Hu, X. (2006). Review mining for music digital libraries: phase II. *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, p. 196.

Eichler, K., Hemsen, H., & Neumann, G. (2008). Unsupervised Relation Extraction from Web Documents. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*. ELRA.

Ellis, D. P. W., Ellis, D. P., Whitman, B., Berenzweig, A., & Lawrence, S. (2002). The quest for ground truth in musical artist similarity. In *Proc. International Symposium on Music Information Retrieval (ISMIR 2002)*, pp. 170–177.

Esuli, A. & Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, vol. 6, pp. 417–422. Citeseer.

Fader, A., Soderland, S., & Etzioni, O. (2011). Identifying relations for open information extraction. *Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP '11*, pp. 1535–1545.

Fernández, L. (2004). *Teoría musical del flamenco*. Madrid: Acordes Concert.

Fernández-Tob\'\ias, I., Cantador, I., Kaminskas, M., & Ricci, F. (2011). A generic semantic-based framework for cross-domain recommendation. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, pp. 25–32. New York, NY, USA: ACM.

Ferragina, P. & Scaiella, U. (2010). Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 1625–1628. ACM.

Ferragina, P. & Scaiella, U. (2012). Fast and accurate annotation of short texts with Wikipedia pages. *Software, IEEE*, *29*(1).

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 363–370.

Font, F. & Oramas, S. (2014). Extending Tagging Ontologies with Domain Specific Knowledge. *Interantional Semantic Web Conference (ISWC 2014)*, pp. 1–4.

Gamallo, P., Garcia, M., & Fernández-Lanza, S. (2012). Dependency-based Open Information Extraction. In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, ROBUS-UNSUP '12, pp. 10–18.

Gamboa, J. M. (2005). *Una historia del flamenco*. Madrid: Espasa-Calpe.

Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., & Schmidt-Thieme, L. (2010). Learning Attribute-to-Feature Mappings for Cold-Start Recommendations. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pp. 176–185. Washington, DC, USA: IEEE Computer Society.

Garcia-Silva, A., Corcho, O., Alani, H., & Gomez-Perez, A. (2012). Review of the state of the art: discovering and associating semantics to tags in folksonomies. *The Knowledge Engineering Review*, *27*(01), 57–85.

Getoor, L. (2012). Entity Resolution : Theory , Practice & Open Challenges. *Tutorial at AAAI-12*, pp. 2018–2019.

Gruhl, D., Nagarajan, M., Pieper, J., Robson, C., & Sheth, A. (2009). Context and Domain Knowledge Enhanced Entity Spotting In Informal Text. In *The Semantic Web-ISWC*, pp. 260–276. Springer.

Hariri, N., Mobasher, B., & Burke, R. D. (2012). Context-aware music recommendation based on latenttopic sequential patterns. In *Sixth {ACM} Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012*, pp. 131–138.

Havasi, C., Speer, R., & Alonso, J. (2007). ConceptNet 3: A Flexible, Multilingual Semantic Network for Common Sense Knowledge. In *Proceedings of Recent Advances in Natural Language Processing*, pp. 27–29. Citeseer.

Heitmann, B. & Hayes, C. (2010). Using Linked Data to Build Open, Collaborative Recommender Systems. In *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*.

Ho, C.-H. & Lin, C.-J. (2012). Large-scale linear support vector regression. *Journal of Machine Learning Research*, *13*, 3323–3348.

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., & Weld, D. S. (2011). Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations. *Network*, pp. 541–550.

Hu, M. & Liu, B. (2004). Mining Opinion Features in Customer Reviews. In *AAAI'04*, pp. 755–760. San Jose, California.

Hu, X. & Downie, J. (2006). Stylistics in customer reviews of cultural objects. *SIGIR Forum*, pp. 49–51.

Hu, X., Downie, J., West, K., & Ehmann, A. (2005). Mining Music Reviews: Promising Preliminary Results. In *ISMIR*, pp. 536–539.

Hu, X., Zhang, X., Lu, C., Park, E. K., & Zhou, X. (2009). Exploiting Wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 389–396. ACM.

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pp. 263–272.

Iacobacci, I., Pilehvar, M. T., & Navigli, R. (2015). SensEmbed: Learning Sense Embeddings for Word and Relational Similarity. In *Proceedings of ACL*, pp. 95–105.

Isabelle Augenstein, D. M. & Ciravegna, F. (2014). Relation Extraction from the Web using Distant Supervision. In *In proceedings of the 19th International Conference, EKAW 2014*.

Jain, H., Prabhu, Y., & Varma, M. (2016). Extreme Multi-label Loss Functions for Recommendation, Tagging, Ranking & Other Missing Label Applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 935–944. ACM.

Jannach, D., Lerche, L., & Kamehkhosh, I. (2015). Beyond "Hitting the Hits": Generating Coherent Music Playlist Continuations with the Right Tracks. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pp. 187–194. New York, NY, USA: ACM.

Jeh, G. & Widom, J. (2002). SimRank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 538–543. ACM.

Jiang, J. & Zhai, C. (2007). A Systematic Exploration of the Feature Space for Relation Extraction. In *HLT-NAACL*, pp. 113–120.

Juslin, P. N. & Västfjäll, D. (2008). Emotional responses to music: the need to consider underlying mechanisms. *The Behavioral and brain sciences*, *31*(5), 559–621.

Kaminskas, M. & Ricci, F. (2012). Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, *6*(2-3), 89–119.

Khrouf, H. & Troncy, R. (2013). Hybrid Event Recommendation Using Linked Data and User Diversity. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pp. 185–192. New York, NY, USA: ACM.

Knees, P. & Schedl, M. (2013). A Survey of Music Similarity and Recommendation from Music Context Data. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, *10*(1).

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer*, *42*(8), 42–49.

Laurier, C., Grivolla, J., & Herrera, P. (2008). Multimodal music mood classification using audio and lyrics. In *Machine Learning and Applications, 2008. ICMLA'08. Seventh International Conference on*, pp. 688–693. IEEE.

Leal, J. P., Rodrigues, V., & Queirós, R. (2012). Computing Semantic Relatedness using DBPedia. *1st Symposium on Languages, Applications and Technologies, SLATE 2012*.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., & Bizer, C. (2014). {DBpedia} - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*.

Libeks, J. & Turnbull, D. (2011). You can judge an artist by an album cover: Using images for music annotation. *IEEE MultiMedia*, *18*(4), 30–37.

Liu, H. & Wang, P. (2014). Assessing Text Semantic Similarity Using Ontology. *Journal of Software*, *9*(2), 490–497.

Logan, B. & Ellis, D. P. W. (2003). Toward Evaluation Techniques for Music Similarity. *SIGIR 2003: Workshop on the Evaluation of Music Information Retrieval Systems*, pp. 7–11.

Logan, B. & Others (2000). Mel Frequency Cepstral Coefficients for Music Modeling. In *ISMIR*.

Lux, M. & Granitzer, M. (2005). A Fast and Simple Path Index Based Retrieval Approach for Graph Based Semantic Descriptions. *In Proceedings of the Second International Workshop on Text-Based Information Retrieval*.

Mausam, Schmitz, M., Bart, R., Soderland, S., & Etzioni, O. (2012). Open Language Learning for Information Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

McAuley, J., Pandey, R., & Leskovec, J. (2015a). Inferring Networks of Substitutable and Complementary Products. *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'15)*, p. 12.

McAuley, J., Targett, C., Shi, Q., & Hengel, A. V. D. (2015b). Image-based Recommendations on Styles and Substitutes. *Proceeding of 38th ACM SIGIR*, pp. 1–11.

McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pp. 1097–1101. New York, NY, USA: ACM.

Mendes, P. N., Jakob, M., García-Silva, A., & Bizer, C. (2011). DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pp. 1–8. ACM.

Middleton, S. E., Roure, D. D., & Shadbolt, N. R. (2009). Ontology-Based Recommender Systems. *Handbook on Ontologies*, *32*(6), 779–796.

Mihalcea, R. & Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 233–242. ACM.

Mikolov, T., Yih, W.-t., & Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. In *HLT-NAACL*, pp. 746–751.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Commun. ACM*, *38*(11), 39–41.

Mobasher, B., Jin, X., & Zhou, Y. (2004). Semantically Enhanced Collaborative Filtering on the Web. In B. Berendt, A. Hotho, D. Mladenic, M. Someren, M. Spiliopoulou, & G. Stumme (Eds.) *Web Mining: From Web to Semantic Web*, *Lecture Notes in Computer Science*, vol. 3209, pp. 57–76. Springer Berlin Heidelberg.

Moïsi, D. (2010). *The Geopolitics of Emotion: How Cultures of Fear, Humiliation, and Hope are Reshaping the World.* New York, NY, USA: Anchor Books.

Montero, C. S., Munezero, M., & Kakkonen, T. (2014). No Title. *Computational Linguistics and Intelligent Text Processing*, pp. 98–114.

Mooney, R. J. & Roy, L. (1999). Content-Based Book Recommending. *Proceedings of the SIGIR-99 Workshop on Recommender Systems: Algorithms and Evaluation*, (August).

Moro, A., Cecconi, F., & Navigli, R. (2014a). Multilingual Word Sense Disambiguation and Entity Linking for Everybody. In *Proceedings of the 13th Internation Conference on Semantic Web (P&D).*

Moro, A. & Navigli, R. (2012). WiSeNet: Building a Wikipedia-based Semantic Network with Ontologized Relations. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 1672–1676.

Moro, A. & Navigli, R. (2013). Integrating Syntactic and Semantic Analysis into the Open Information Extraction Paradigm. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2148–2154. AAAI Press.

Moro, A., Raganato, A., & Navigli, R. (2014b). Entity Linking meets Word Sense Disambiguation : a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL).*

Moro, A., Raganato, A., & Navigli, R. (2014c). Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, *2*, 231–244.

Musto, C., Semeraro, G., Lops, P., & de Gemmis, M. (2014). Combining Distributional Semantics and Entity Linking for Context-Aware Content-Based Recommendation. In *User Modeling, Adaptation, and Personalization - 22nd International Conference, {UMAP} 2014, Aalborg, Denmark, July 7-11, 2014. Proceedings*, pp. 381–392.

Nadeau, D. & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3–26.

Nakashole, N., Weikum, G., & Suchanek, F. M. (2012). PATTY: A Taxonomy of Relational Patterns with Semantic Types. *EMNLP-CoNLL*, (July), 1135–1145.

Navarro, J. L. & Ropero, M. (1995). *Historia del flamenco*. Sevilla: Ed. Tartessos.

Navigli, R. & Ponzetto, S. P. (2010). BabelNet : Building a Very Large Multilingual Semantic Network. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, (July), 216–225.

Navigli, R. & Ponzetto, S. P. (2012). BabelNet: The Automatic Construction, Evaluation and Application of a Wide-coverage Multilingual Semantic Network. *Artificial Intelligence*, *193*, 217–250.

Neumayer, R. & Rauber, A. (2007). Integration of text and audio features for genre classification in music information retrieval. In *European Conference on Information Retrieval*, pp. 724–727. Springer.

Ning, X. & Karypis, G. (2012). Sparse Linear Methods with Side Information for Top-n Recommendations. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pp. 155–162. New York, NY, USA: ACM.

Oramas, S. (2014). Harvesting and Structuring Social Data in Music Information Retrieval. *Extended Semantic Web Conference (ESWC). Lecture Notes in Computer Science*, *8465*, 817–826.

Oramas, S., Espinosa-anke, L., Sordo, M., Saggion, H., & Serra, X. (2016). ELMD : An Automatically Generated Entity Linking Gold Standard Dataset in the Music Domain. In *In Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016.*

Oramas, S., Gómez, F., Gómez, E., & Mora, J. (2015a). FlaBase: Towards the Creation of a Flamenco Music Knowledge Base. In *Proceedings of the International Society for Music Information Retrieval Conference.*

Oramas, S., Sordo, M., & Espinosa-anke, L. (2015b). A Rule-Based Approach to Extracting Relations from Music Tidbits. In *2nd Workshop in Knowledge Extraction from Text, WWW'15.*

Oramas, S., Sordo, M., Espinosa-Anke, L., & Serra, X. (2015c). A Semantic-based Approach for Artist Similarity. *Proceedings of 16th the International Society for Music Information Retrieval Conference*, (October), 100–106.

Oramas, S., Sordo, M., & Serra, X. (2014). Automatic Creation of Knowledge Graphs from Digital Musical Document Libraries. In *Conference in Interdisciplinary Musicology.* Berlin.

Ostuni, V. C., Di Noia, T., Di Sciascio, E., & Mirizzi, R. (2013). Top-N Recommendations from Implicit Feedback Leveraging Linked Open Data. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pp. 85–92. New York, NY, USA: ACM.

Ostuni, V. C., Di Noia, T., Di Sciascio, E., Oramas, S., & Serra, X. (2015). A Semantic Hybrid Approach for Sound Recommendation. In *Proceedings of the 24th International Conference on World Wide Web (Companion Volume)*, pp. 85–86. International World Wide Web Conferences Steering Committee.

Ostuni, V. C., Di Noia, T., Mirizzi, R., & Di Sciascio, E. (2014). A Linked Data Recommender System using a Neighborhood-based Graph Kernel. In

The 15th International Conference on Electronic Commerce and Web Technologies, Lecture Notes in Business Information Processing. Springer-Verlag.

Pachet, F. & Cazaly, D. (2000). A taxonomy of musical genres. In *Content-Based Multimedia Information Access-Volume 2*, pp. 1238–1245. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.

Passant, A. (2010). dbrec: music recommendations using DBpedia. In *Proc. of 9th Int. Sem. Web Conf.*, ISWC'10, pp. 209–224.

Pons, J., Lidy, T., & Serra, X. (2016). Experimenting with musically motivated convolutional neural networks. In *Content-Based Multimedia Indexing (CBMI), 2016 14th International Workshop on*, pp. 1–6. IEEE.

Porter, A., Bogdanov, D., Kaye, R., Tsukanov, R., Serra, X., Group, M. T., Fabra, U. P., & Foundation, M. (2015). Acousticbrainz: a community platform for gathering music information obtained from audio. *16th International Society for Music Information Retrieval Conference (ISMIR 2015)*, pp. 786–792.

Ratcliff, J. W. & Metzener, D. (1988). Pattern matching: The gestalt approach. *Dr. Dobb's Journal*, *13*, 46–72.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pp. 452–461. Arlington, Virginia, United States: AUAI Press.

Rizzo, G., van Erp, M., & Troncy, R. (2014). Benchmarking the Extraction and Disambiguation of Named Entities on the Semantic Web. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Rorvig, M. (1999). Images of similarity: A visual exploration of optimal similarity metrics and scaling properties of TREC topic-document sets. *Journal of the American Society for Information Science*, *50*(8), 639–651.

Rowe, M. (2014). SemanticSVD++: incorporating semantic taste evolution for predicting ratings. In *2014 {IEEE/WIC/ACM} International Conferences on Web Intelligence, {WI} 2014*.

Ruiz-Casado, M., Alfonseca, E., Okumura, M., & Castells, P. (2008). Information extraction and semantic annotation of wikipedia. *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pp. 145–169.

Saggion, H. & Gaizauskas, R. (2004). Multi-document summarization by cluster/profile relevance and redundancy removal. In *Proceedings of the Document Understanding Conference*, pp. 6–7.

Sanden, C. & Zhang, J. Z. (2011). Enhancing Multi-label Music Genre Classification Through Ensemble Techniques. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pp. 705–714. New York, NY, USA: ACM.

Saveski, M. & Mantrach, a. (2014). Item cold-start recommendations: learning local collective embeddings. *RecSys '14 Proceedings of the 8th ACM Conference on Recommender systems*, pp. 89–96.

Schedl, M., Gómez, E., & Urbano, J. (2014). Music Information Retrieval: Recent Developments and Applications. *Foundations and Trends® in Information Retrieval*, *8*(2-3), 127–261.

Schedl, M., Hauger, D., & Urbano, J. (2013). Harvesting microblogs for contextual music similarity estimation: a co-occurrence-based framework. *Multimedia Systems*, *20*(6), 693–705.

Schedl, M., Knees, P., & Widmer, G. (2005). A Web-Based Approach to Assessing Artist Similarity using Co-Occurrences. In *Proceedings of the 4th International Workshop on Content-Based Multimedia Indexing {(CBMI'05)}*.

Schindler, A. & Rauber, A. (2015). An audio-visual approach to music genre classification through affective color features. In *European Conference on Information Retrieval*, pp. 61–67. Springer.

Seyerlehner, K., Schedl, M., Pohle, T., & Knees, P. (2010). Using block-level features for genre classification, tag classification and music similarity estimation. *Submission to Audio Music Similarity and Retrieval Task of MIREX, 2010*.

Shawe-Taylor, J. & Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press.

Soon, W. M., Ng, H. T., & Lim, D. C. Y. (2001). A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational linguistics*, *27*(4), 521–544.

Sordo, M., Oramas, S., & Espinosa-Anke, L. (2015). Extracting Relations from Unstructured Text Sources for Music Recommendation. In *Proceedings of the 20th International Conference on Applications of Natural Language to Information Systems, NLDB 2015*, pp. 369–382. Cham: Springer International Publishing.

Steck, H. (2013). Evaluation of recommendations: rating-prediction and ranking. In *RecSys*, pp. 213–220.

Sturm, B. L. (2012). A survey of evaluation in music genre recognition. In *International Workshop on Adaptive Multimedia Retrieval*, pp. 29–66. Springer.

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pp. 697–706. ACM.

Tata, S. & Di Eugenio, B. (2010). Generating Fine-Grained Reviews of Songs from Album Reviews. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1376–1385. Association for Computational Linguistics.

Tesnière, L. (1959). *Elements de Syntaxe Structurale.* Editions Klincksieck.

Tsoumakas, G. & Katakis, I. (2006). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, *3*(3).

Turnbull, D., Barrington, L., & Lanckriet, G. (2008). Five approaches to collecting tags for music. *Proceedings of 9th the International Society for Music Information Retrieval Conference*, pp. 225–230.

Tzanetakis, G. & Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, *10*(5), 293–302.

Usbeck, R., Ngomo, A.-C. N., Auer, S., Gerber, D., & Both, A. (2014a). Agdistis-agnostic disambiguation of named entities using linked open data. In *International Semantic Web Conference*, p. 2.

Usbeck, R., Ngomo, A.-c. N., Michael, R., Gerber, D., Coelho, S. A., & Both, A. (2014b). AGDISTIS - Graph-Based Disambiguation of Named Entities using Linked Data. *The Semantic Web – ISWC 2014.*

Usbeck, R., Röder, M., Ngomo, A.-C. N., Baron, C., Both, A., Brümmer, M., Ceccarelli, D., Cornolti, M., Cherix, D., Eickmann, B., Ferragina, P., Lemke, C., Moro, A., Navigli, R., Piccinno, F., Rizzo, G., Sack, H., Speck, R., Troncy, R., Waitelonis, J., & Wesemann, L. (2015). GERBIL – General Entity Annotator Benchmarking Framework. *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pp. 1133–1143.

van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. *Electronics and Information Systems department (ELIS)*, p. 9.

Vigliensoni, G. & Fujinaga, I. (2014). Identifying Time Zones in a Large Dataset of Music Listening Logs. In *Proceedings of the First International Workshop on Social Media Retrieval and Analysis*, SoMeRA '14, pp. 27–32. New York, NY, USA: ACM.

Voskarides, N. & Meij, E. (2015). Learning to Explain Entity Relationships in Knowledge Graphs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pp. 564–574.

Wang, F., Wang, X., Shao, B., Li, T., & Ogihara, M. (2009). Tag Integrated Multi-Label Music Style Classification with Hypergraph. In *ISMIR*, pp. 363–368.

Wang, M. (2008). A Re-examination of Dependency Path Kernels for Relation Extraction. In *IJCNLP*, pp. 841–846.

Whitman, B. & Lawrence, S. (2002). Inferring descriptions and similarity for music from community metadata. In *Proceedings of the 2002 International Computer Music Conference*, pp. 591–598.

Zhang, X., Liu, Z., Qiu, H., & Fu, Y. (2009). A Hybrid Approach for Chinese Named Entity Recognition in Music Domain. *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pp. 677–681.

Zhou, G. & Su, J. (2002). Named entity recognition using an HMM-based chunk tagger. In *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 473–480. Association for Computational Linguistics.

Ziegler, C.-N., Lausen, G., & Schmidt-Thieme, L. (2004). Taxonomy-driven computation of product recommendations. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pp. 406–415. New York, NY, USA: ACM.

Appendix A: Datasets and Knowledge Bases

Introduction

Appendix B: publications by the author

In press

Journal papers

Oramas S., Espinosa-Anke L., Sordo M., Saggion H. & Serra X. (2016). Information Extraction for Knowledge Base Construction in the Music Domain. *Data & Knowledge Engineering, Volume 106*, Pages 70-83.

Oramas S., Ostuni V. C., Di Noia T., Serra, X., & Di Sciascio E. (2016). Music and Sound Recommendation with Knowledge Graphs. *ACM Transactions on Intelligent Systems and Technology, Volume 8*, Issue 2, Article 21.

Oramas S., Sordo M. (2016). Knowledge is Out There: A New Step in the Evolution of Music Digital Libraries. *Fontes Artis Musicae, Vol 63, no. 4*.

Conference papers

Oramas S., Espinosa-Anke L., Lawlor A., Serra X., & Saggion H. (2016). Exploring Music Reviews for Music Genre Classification and Evolutionary Studies. *In Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR 2016).*

Oramas S., Espinosa-Anke L., Sordo M., Saggion H., & Serra X. (2016). ELMD: An Automatically Generated Entity Linking Gold Standard in the Music Domain. *In Proceedings of the 10th Conference on Language Resources and Evaluation (LREC 2016).*

Espinosa-Anke, L., Oramas S., Camacho-Collados J., & Saggion H. (2016). Finding and Expanding Hypernymic Relations in the Music Domain. *In Proceedings of the 19th International Conference of the Catalan Association for Artificial Intelligence (CCIA 2016).*

Oramas S., Sordo M., Espinosa-Anke L., & Serra X. (2015). A Semantic-based approach for Artist Similarity. *In Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015).*

Oramas S., Gómez F., Gómez E., & Mora J. (2015). FlaBase: Towards the creation of a Flamenco Music Knowledge Base. *In Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015).*

Ostuni V. C., Oramas S., Di Noia T., Serra, X., & Di Sciascio E. (2015). A Semantic Hybrid Approach for Sound Recommendation. *In Proceedings of the 24th International World Wide Web Conference (WWW 2015, Poster track).*

Oramas S., Sordo M., & Espinosa-Anke L. (2015). A Rule-based Approach to Extracting Relations from Music Tidbits. *In Proceedings of the 2nd Workshop on Knowledge Extraction from Text (KET 2015).*

Sordo, M., Oramas S., & Espinosa-Anke L. (2015). Extracting Relations from Unstructured Text Sources for Music Recommendation. *In Proceedings of the 20th International Conference on Applications of Natural Language to Information Systems (NLDB 2015).*

Oramas S., Sordo M., & Serra X. (2014). Automatic Creation of Knowledge Graphs from Digital Musical Document Libraries. *In Proceedings of the 9th Conference on Interdisciplinary Musicology (CIM 2014).*

Oramas S. (2014). Harvesting and Structuring Social Data in Music Information Retrieval. *In Proceedings of the Extended Semantic Web Conference (ESWC 2014, PhD Symposium).*

Font, F., Oramas, S., Fazekas, G., & Serra, X. (2014). Extending Tagging Ontologies with Domain Specific Knowledge. In *Proceedings of the International Semantic Web Conference (ISWC 2014, Poster track).*

Tutorials and Challenges

Oramas S., Espinosa-Anke L., Zhang S., Saggion H., & Serra X. (2016). Natural Language Processing for Music Information Retrieval. *17th International Society for Music Information Retrieval Conference (ISMIR 2016).*

Conference presentations

Oramas, S. (2017). Discovering Similarities and Relevance Ranking of Renaissance Composers. *The 63rd Annual Meeting of the Renaissance Society of America (RSA)*, Chicago.

Oramas S. (2015). Information Extraction for the Music Domain. *The 2nd International Workshop on Human History Project: Natural Language Processing and Big Data*, CIRMMT, Montreal.

Oramas, S., & Sordo M. (2015). Knowledge Acquisition from Music Digital Libraries. *The International Association of Music Libraries and International Musicological Society Conference (IAML/IMS 2015)*, New York.