

A Rule-Based Approach to Extracting Relations from Music Tidbits

Sergio Oramas
Music Technology Group
Universitat Pompeu Fabra
sergio.oramas@upf.edu

Mohamed Sordo
Center for Comp. Science
University of Miami
msordo@miami.edu

Luis Espinosa-Anke
TALN Group
Universitat Pompeu Fabra
luis.espinosa@upf.edu

ABSTRACT

This paper presents a rule based approach to extracting relations from unstructured music text sources. The proposed approach identifies and disambiguates musical entities in text, such as songs, bands, persons, albums and music genres. Candidate relations are then obtained by traversing the dependency parsing tree of each sentence in the text with at least two identified entities. A set of syntactic rules based on part of speech tags are defined to filter out spurious and irrelevant relations. The extracted entities and relations are finally represented as a knowledge graph. We test our method on texts from *songfacts.com*, a website that provides tidbits with facts and stories about songs. The extracted relations are evaluated intrinsically by assessing their linguistic quality, as well as extrinsically by assessing the extent to which they map an existing music knowledge base. Our system produces a vast percentage of linguistically correct relations between entities, and is able to replicate a significant part of the knowledge base.

Categories and Subject Descriptors

I.2.7 [Computing Methodologies]: Artificial intelligence—*Natural language processing*

Keywords

Open information extraction; relation extraction; music

1. INTRODUCTION

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 [23]. In the music domain, scant research has taken advantage of Information Extraction techniques [19]. The Web is full of user generated content about music, and all this information is mainly

present as non structured text. Extraction and curation of this knowledge may be very useful for several tasks, such as Music Recommendation, or to populate and enrich existing knowledge bases.

In this paper we propose a method that exploits unstructured music text sources from the web in order to create or extend music knowledge graphs. The method does so by identifying music-related entities in the text (such as *song*, *band*, *person*, *album* and *music genre*) and extracting relations between these entities using a rule based approach that exploits dependency trees and part-of-speech tags. In our system, candidate relations are obtained by traversing the dependency tree of each sentence between identified named entities. After that, irrelevant relations are filtered out using a set of syntactic rules based on part-of-speech tags.

We design a two-fold evaluation that focuses on the linguistic quality of the extracted relations as well as on the extent to which these relations replicate the information contained in an existing music knowledge base. As for the linguistic evaluation, we assess the correctness of the system by comparing its output to manual gold-standard annotation. In addition, a finer-grained comparison is carried out in a word-level evaluation. With regard to the data-driven evaluation, our system generates a knowledge graph that can be mapped with high accuracy to MusicBrainz, an existing knowledge base in the music domain. In both cases we obtain very promising results, which suggest that this is a valid line of research for creating or extending music-related knowledge bases with arbitrary relations extracted among entities.

The remainder of this paper is structured as follows: Section 2 provides a brief overview of Relation Extraction. We present the proposed method in Section 3. The dataset and the experimental setup are described in Section 4. Finally Section 5 draws some conclusions and points to potential future lines of work.

2. RELATED WORK

In traditional Relation Extraction (RE), the vocabulary of extracted relations is defined a priori, i.e. in a domain ontology or an extraction template. 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 [14]. More promising approaches, called semi-supervised approaches, bootstrapping approaches, or distant supervision approaches do not need a complete hand labeled training corpus. These approaches

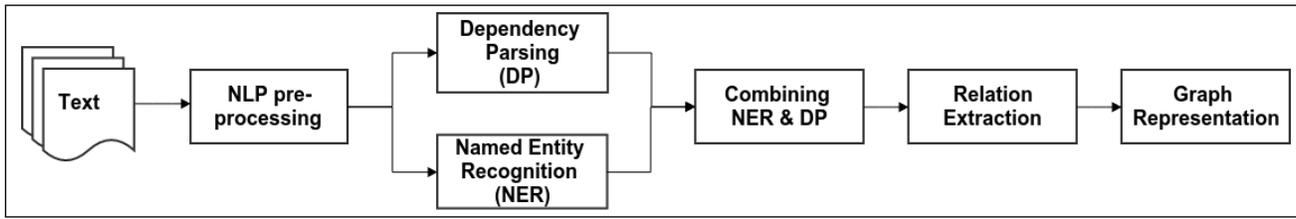


Figure 1: Workflow of the proposed method.

often use an existent knowledge base to heuristically label a text corpus (e.g., [14, 7]) More recently, multi-instance learning approaches are combined with distant supervision to combat the problem of ambiguously-labeled training data for the identification of overlapping relations [14, 24].

In contrast with traditional Relation Extraction, Open Information Extraction methods do not require a pre-specified vocabulary, as they aim to discover all possible relations in the text [3]. However, these methods have to deal with uninformative and incoherent extractions. In ReVerb [11] 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 [17] to increase the number of possible extracted relations.

Unsupervised approaches do not need any annotated corpus. In [10] 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 [1]. The aim of this paper is to show that when the analyzed unstructured text sources are domain specific (in this case music) and reviewed by a group of domain experts, unsupervised approaches using simple rules can map extracted relations with an existing knowledge base with a high precision.

3. METHODOLOGY

3.1 NLP Pre-processing

Figure 1 depicts the workflow of the proposed method. Given a text input (e.g., a collection of web documents) the pre-processing module segments it into single sentences. Each sentence is subsequently divided into a sequence of words or tokens. Our method uses the Stanford NLP implementation¹ of the tokenizer.

3.2 Named Entity Recognition (NER)

Although Named entity recognition (NER) is not a solved problem [16], there are many available tools with good enough performance ratios [12]. Among these tools, we selected DBpedia Spotlight, a system for automatically annotating text documents with DBpedia URIs. DBpedia Spotlight is shared as open source and deployed as a Web service freely available for public use². It has a competitive performance and evaluations show an F-measure around 0.5 [18].

Our NER module receives a list of sentences as input and uses DBpedia Spotlight to find DBpedia entities in the sen-

tences. The entities are then annotated with their corresponding URI and type. In the current approach we only consider 5 different types that are relevant to the music domain: *song*, *band*, *person*, *album* and *music genre*. The rest of the recognized entities are ignored.

Identification of song titles in text is a challenge, since titles are often short and ambiguous [13]. Fortunately, due to the nature of the dataset (explained in Section 4.1) the complexity of the identification process of song titles is reduced considerably, under the assumption that ambiguity is less probable in this scenario. Such assumption is possible since we know a priori which song is each tidbit talking about. Thus, apart from using DBpedia Spotlight, each song title is searched in its tidbits. Moreover, further analysis showed that the song in question is usually referred using expressions such as “the song” or “this song”. Therefore, we also looked for these structures and treat them as detected song entities.

3.3 Dependency Parsing (DP)

Dependency Parsing provides a tree-like syntactic structure of a sentence based on the linguistic theory of Dependency Grammar [22]. One of the outstanding features of Dependency Grammar is that it represents binary relations between words [2], where there is a unique edge joining a node and its parent node (see Figure 2 for the full parsing of an example sentence). Dependency relations have been successfully incorporated to RE systems. For example, [6] describe and evaluate a RE system based on shortest paths among named entities.

Our Dependency Parsing module uses the implementation by [4] and produces a tree for each sentence. Each node in the tree represents a single word of the sentence, together with additional linguistic information like part-of-speech³ and syntactic function. For instance, in Figure 2 the word *Freedom* is the subject (SBJ) of the root word *was*. The definition of all these syntactic functions is given in [21]. In our case, however, we want to find relations between music-related entities, which can consist of more than one word. The next module takes care of this.

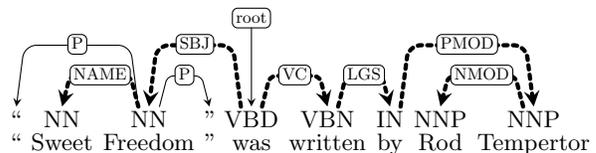


Figure 2: Example sentence with dependency parsing tree.

¹<http://nlp.stanford.edu/software/>

²<https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki/Web-service>

³http://ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.htm

3.4 Combining NER & DP

The aim of this module is to combine the output of the two previous modules. For each recognized music-related entity in the NER module (Section 3.2), the combination module merges all the nodes in the dependency tree of the sentence that correspond to that entity into a single node. Figure 3 shows how the example sentence from Figure 2 is modified by merging the nodes that correspond to the recognized entities (in this case, the album “*Sweet Freedom*” and the person *Rod Temperton*) into single nodes.

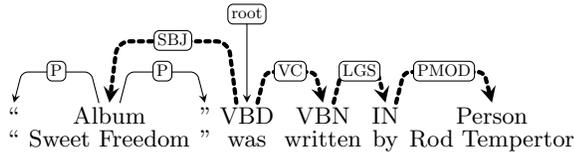


Figure 3: Example sentence with its modified dependency tree, after merging nodes that correspond to an entity.

3.5 Relation Extraction (RE)

The Relation Extraction module analyzes the modified dependency trees from the combination module (Section 3.4), and extracts relations between pairs of recognized music-related entities⁴. Two entities (nodes) in a tree are considered to be related if there is a path between them that does not contain any other entity. Since dependency trees are directed trees, there is no guarantee in finding a path. Therefore we use an undirected version of the tree to obtain the path between the aforementioned pair of entities. Interestingly, the nodes that are part of the path between the two entities are considered by our method to represent the actual relation between the entities. In the example of Figure 3, the resulting path between “*Sweet Freedom*” and *Rod Temperton* contains the words *was*, *written*, *by*. These words are used to define the relation label between “*Sweet Freedom*” and *Rod Temperton*.

During our experimentation with this module using the Songfacts dataset (explained in Section 4.1), we analyzed the output of the relation extraction. For each pair of type of entities (e.g.: Album-Person), we counted the frequency of the different paths between them. Instead of taking the plain words of the path, we used their corresponding part-of-speech tags. We call this a relation class. For instance, the previous example (Figure 3) has a relation between an entity of type *Album* (“*Sweet Freedom*”) and an entity of type *Person* (*Rod Temperton*). The relation class *Album-Person* in this case contains the terms *VBD*, *VC* and *IN*, which are part of speech labels meaning verb past tense, verb past participle and preposition, respectively. After manually checking the most frequent relations between pairs of entity types, we observed that while many relations make sense, some others were not providing any useful information. In order to filter this list of relations, we applied some constraints. First, we removed all the relation classes with a frequency less than 10. From the remaining relation classes

⁴The current method only considers sentences with two or more entities. Sentences with no entities or only one entity are ignored, since our goal is to relate pairs of entities.

we picked manually the best classes that represented the relation between pairs of entity types, in other words, relations that seemed to make sense linguistically. The list of rules is depicted in Figure 4.

3.6 Graph Representation

After the list of relations between entities is filtered, the method creates a graph representation of it, where the nodes are the music-related entities and the edges represent the relations (i.e., the path) between pairs of entities. The graph contains five chosen types of nodes corresponding to the 5 music-related types: *song*, *band*, *person*, *album* and *music genre*.

4. EVALUATION

We tested our method against a dataset gathered from *songfacts.com* (Section 4.1). The output of the method has been evaluated from two different standpoints, namely: (1) a linguistically motivated evaluation of the extracted relations and (2) a data-driven evaluation of the extracted knowledge. The linguistic evaluation quantifies the correctness of a relation by comparing it to a reference annotation manually crafted by a Computational Linguistics expert. Data-driven evaluation compare the extracted knowledge with a reference knowledge-base. The following subsections provide a detailed description of the dataset and the experimental results.

4.1 Dataset

Songfacts⁵ is an online database that collects, stores and provides facts and stories about songs. The site contains information about more than 30,000 songs belonging to nearly 6,000 artists. Songfacts tidbits are little pieces of information telling stories about a song, such as what the song is about or who wrote it, who produced it, who collaborated with whom or who directed the video clip, etc. Therefore, a huge amount of information about the actors involved in the creative process of a song is present in the aforementioned tidbits. We crawled the whole song dataset from Songfacts in mid-January 2014. We chose only songs whose title had been recognized by our system as a song entity and only those among them who were involved in at least one relation with another recognized entity. Finally, we obtained 12,838 entities and 12,306 relations. Among the detected entities, 6,116 were songs, and those songs were related to 1,483 different artists.

4.2 Linguistic Evaluation

The evaluation of the approach described in this paper was inspired by previous work in Relation Extraction [11, 17]. In general, the literature agrees in evaluating the automatically extracted relations in terms of correctness according to human judgement. However, a finer-grained analysis is preferred in [3], where relations are judged as being *concrete* or *abstract*. Building up on this well-established methodology, we analyzed a sample of 205 relations extracted from 155 randomly selected sentences. Two human judges marked a relation as “correct” if the information contained in the sentence implied or connoted that the relation was true. An “incorrect” label was assigned otherwise.

⁵<http://www.songfacts.com>

```

('Album', 'Band'): ['(VBD)(NN)', '(VBZ)(NN)', '(VBD)(NN)(IN)(NN?)', '(NN?) (IN)(NN?)', '(VBD)(VBN)(IN)'],
('Album', 'Person'): ['(NN)(NN?)', '(VBD?) (NN)', '(NN?) (IN)(NN?)', '(VBD)(VBN?) (IN)(NN?)'],
('Album', 'Song'): ['(NN?) (VBD)', '(VBD)(NN)(IN)'],
('Band', 'Album'): ['(VBD?) (NN)', '(VBD)(IN)', '(NN)(VBD)', '(NN)(IN)(NN)', '(NN?) (VBD)(IN)(NN)'],
('Band', 'MusicGenre'): ['(VBZ)(NN)', '(VBD)(NN)', '(VBP)(NN)'],
('Band', 'Person'): ['(NNS)', '(NN)(NN?)', '(NN)(VBD)', '(NN)(CC)(NN)', '(VBD)(IN)(NN?)', '(VBD)(NN)'],
('Band', 'Song'): ['(NN?) (VBD)', '(VBD?) (NN)(IN)', '(NN)(VBD)(NN)(IN)'],
('MusicGenre', 'Band'): ['(NN)(NN?)'],
('MusicGenre', 'Person'): ['(NN)(NN?)', '(NN)(CC)(NN)'],
('Person', 'Album'): ['(NN)(NN?)', '(NN)(IN)(NN?)', '(NN?) (VBD)(IN?) (NN?)'],
('Person', 'Band'): ['(NN?) (IN)', '(VBD)(IN?) (NN?)', '(VBD)(NN)(IN)', '(NN)(IN)(NN)'],
('Person', 'Person'): ['(NN)', '(NN)(CC)(NN)', '(VBD)(IN)(NN?)', '(VBD)(CC)(VBD)'],
('Person', 'Song'): ['(VBD)(IN?)', '(VBD)(IN?)', '(NN)(IN)', '(VBD)(IN)(VBN?)', '(NN)(VBZ)(IN)', '(VBZ)(NN)(IN)',
                    '(NN)(VBD)(NN?) (IN?)', '(VBD)(IN?) (NN)(IN?)'],
('Song', 'Album'): ['(VBD)(VBN)', '(VBZ)(IN)(NN)', '(VBD)(VBN?) (IN)(NN?)'],
('Song', 'Band'): ['(VBZ)(NN)', '(IN)(NN)', '(VBD)(VBN?) (IN)(NN?)', '(VBZ)(VBN?) (IN)(NN?)'],
('Song', 'MusicGenre'): ['(VBZ)(NN?)', '(VBZ)(NN)(IN)(NN?)', '(VBD)(VBN)(IN)(NN?)'],
('Song', 'Person'): ['(VBZ)(NN)', '(IN)(NN?)', '(VBD?) (NN)', '(VBD)(VBN?) (IN)(NN?)', '(VBZ)(NNS)(IN)',
                    '(VBZ)(NN)(IN)(NN?)', '(NN)(VBD)(VBN)(IN)'],

```

Figure 4: Part-of-speech rules that represent the relation between pairs of music entity types.

The results obtained were very high with regard to the observed agreement. Specifically, our results indicate that out of 205 relations, both evaluators agreed in judging 146 relations as correct and 23 as incorrect. In terms of agreement, this yields 82.43% for the whole set of relations, and 71.21% in the correct ones. We also computed Cohen’s Kappa [8] agreement measure in order to have an additional viewpoint of the reliability of the human evaluation. Our computation of Cohen’s Kappa was 40.68, which is considered to be within the *fair* agreement boundaries [15]. We should also note that Cohen’s Kappa strongly punishes potential agreement by chance, which in an experiment with two evaluators and two classes is the likeliest of the scenarios. Let us illustrate the above results with an example that showcases a case of agreement (1) and a case of disagreement (2) in the same sentence.

(1) **Sentence:** [Weezer] frontman [Rivers Cuomo] wrote this song for and about Jamie Young, the band’s first lawyer.

Entities: Band ↔ Person

Extracted Relation: Weezer (frontman) Rivers Cuomo

(2) **Sentence:** Weezer frontman [Rivers Cuomo] wrote [this song] for and about Jamie Young, the band’s first lawyer.

Entities: Person ↔ Song

Extracted Relation: Rivers Cuomo (frontman wrote) Weezer_–_Jamie

In the first example both evaluators agreed in assigning a “correct” label to the relation. However, in the second example one evaluator found it to be incorrect. This could be due to the distracting role of the word “frontman”, which can be considered to be a property of the first entity, rather than an element of the relation. Although this dichotomy has been addressed in previous work (e.g. [11] evaluated a relation to be correct where critical information was dropped from the relation but included in the second argument), we found that a word-level quantitative evaluation would reflect better the extent to which essential words in a relation are missing, and vice-versa. Specifically, this lexical approach compares the relations extracted by the system with those that would be extracted by a human expert. We computed Precision, Recall and F-Measure by looking at the overlap

between the words included in human-extracted and system-extracted relations. For instance, in the above case, the relation in the pair Person ↔ Song would get a score of P=0.5 and R=1 because the human evaluator extracted the relation *Rivers Cuomo (wrote) Weezer_–_Jamie*. “Frontman” would be a false positive. Table 1 provides results for the full evaluation dataset and for each pair of entity types.

It is worth noting how our approach has performed very well in certain pairs, specifically in the MusicGenre ↔ Band, Person ↔ Band, Band ↔ Song and Song ↔ MusicGenre pairs. This might be due to the many straightforward one-word relations among these entities, as shown in the following examples:

Sentence: The [Christian Metal] band [Stryper] recorded this song for their 1990 album Against the Law and made a video for it.

Entities: MusicGenre ↔ Band

Sentence: Jessie Lacey of Brand New’s girlfriend cheated on him with [John Nolan] of [Taking Back Sunday].

Entities: Person ↔ Band

Lower scores were obtained in relations like Person ↔ Album or Band ↔ Album. A closer look at these relations shows that there are many cases where an album is preceded by a significant number of adjectives and other noun modifiers. These modifiers are often described as sibling nodes of the relation in the dependency tree, and thus do not appear in the path between the two related entities. An improvement of the relation extraction module of our method — especially the way we generate a path between entities in the dependency tree — would significantly improve the recall scores, and therefore constitutes a clear avenue for future work. For example, by specifying for each token in the relation, whether it is a constituent of a relation between entities, an attribute of one of them, or none of them.

4.3 Data-driven evaluation

The output of our system can be regarded as a knowledge base of music related information. This knowledge base consists of entities and relations, two building blocks of a simple, non-taxonomic ontology. According to [9], a learned ontology can be evaluated in three different ways: in the context of an application, by domain experts or by comparing

	P - P	P - S	B - P	B - A	P - A	B - S	S - A	M - B	M - P	S - M
Precision	59.09	95.52	88.88	94.87	81.13	95.74	95.55	85.18	62.5	96.55
Recall	48.14	60.95	49.23	31.62	45.74	55.55	47.25	71.87	66.66	57.14
F-Score	53.55	74.41	63.36	47.43	58.49	70.36	63.23	77.96	64.51	71.79

Table 1: Results per pairs for the token-wise evaluation. The analyzed entities are: P (person), S (song), B (band), A (album), M (Music Genre)

it with a predefined reference ontology (i.e., a Gold Standard). In this section we use the latter approach as it allows a certain amount of automation of the evaluation process [20]. The Gold Standard with which to evaluate our learned knowledge base was obtained from MusicBrainz⁶, the most complete and accurate open knowledge base of music information. We extracted a subset of the MusicBrainz database containing all the entities that could be mapped to entities in our knowledge base, along with their corresponding relations. This mapping was accomplished as follows: for those entities in our knowledge base with a DBpedia URI (such as entities of type *person*, *band* and *album*) we obtain their MusicBrainz ID. Regarding entities of type *song*, since we do not have a URI, we query the MusicBrainz API⁷ by using song and artist name strings. Entities of type *musicgenre* were not considered for this evaluation as there is no corresponding concept in MusicBrainz. Finally, relations between the mapped entities were obtained using the aforementioned MusicBrainz API.

Of the 12,838 entities in our knowledge base we could map 11,740 entities in MusicBrainz, which represent a 91.4%. In order to evaluate both knowledge bases we removed those entities that could not be mapped to MusicBrainz. To facilitate the evaluation process we represented both our knowledge base and the Gold Standard as graphs, where nodes correspond to musical entities and edges represent relations between those entities. Some pairs of entities could have more than one relationship. For example, *artist* “Bob Ezrim” is related to *album* “The Wall” as *orchestration* and *producer* in MusicBrainz. In our case we simplified this by merging all these relations into a single edge with multiple labels. Thus from now on, when we mention a relation, we are in fact referring to an edge in the graph. To refer to a single relation we use the expression “relation term”.

As a first evaluation we calculated the overlap of edges between the two graphs, regardless of the labels (i.e, the relation concepts) of those edges. We obtained an overlap of 5,236 edges, which represented a 49.4% of the Gold Standard relations and a 39.8% of our extracted knowledge base. Once this overlap is obtained, the next step is to assess how our knowledge base “fits” the MusicBrainz Gold Standard [5]. Evaluating two knowledge bases, is an arduous task. Typical Information Retrieval evaluation measures such as precision and recall cannot be easily used in their strict sense, as there is no clear definition of what knowledge is acquired [5]. The main problem in our case is that the vocabularies used in the two knowledge bases are different. Nevertheless, even though the vocabularies are different, many of their terms refer to similar music-related concepts. Hence, finding a conceptual equivalence between relation terms in our knowledge base and the MusicBrainz Gold Standard is fundamental in

order to evaluate our approach in terms of precision and recall. Of the overlapping relations (i.e., edges), we selected all the distinct combinations of the MusicBrainz relation terms and our knowledge base relation terms that co-occur in the same edges and grouped them. A closer look at the Gold Standard graph shows that many relations in this graph do not have labels. For example, many artists are related to recordings in MusicBrainz without any explicit relation concept. We thus decided to ignore these relations from our evaluation, keeping 1,143 overlapping edges.

The grouping of the relation terms in relation types resulted in 727 different combinations, for which an equivalence had to be computed. A MusicBrainz relation type is considered to have an equivalent relation type in our knowledge base if their relation terms are conceptually similar. For example, the relation term *married* in MusicBrainz is conceptually implicit in the relation term *husband* in our knowledge base. Furthermore, MusicBrainz also organizes its relation terms in tree-like taxonomies, where conceptually similar terms are grouped in the same tree branch⁸. This can be used to decide whether a term in our knowledge base can be mapped to a term in the MusicBrainz relation taxonomies. In order to compute the equivalence of the 727 combinations we asked three human annotators to vote whether the two relation terms are conceptually similar⁹.

Once this equivalence is obtained, we can compute precision and recall at a relation level (i.e, edge level). For this evaluation we only use a subset of the graphs. The subset is defined by all the overlapping edges in both graphs with at least one relation term. For each edge in the graphs, precision refers to how many relation terms in our knowledge base edge have an equivalence in the Gold Standard edge, whilst recall refers to how many relation terms in the Gold Standard edge have an equivalence in our knowledge base edge. Lets use the previous example of *artist* “Bob Ezrim” related to *album* “The Wall” as *orchestration* and *producer* in MusicBrainz. The relation between “Bob Ezrim” and “The Wall” in our knowledge base is defined by the single term *producer*. In this case, precision will be 1, but recall will be 0.5. We computed the average precision and recall over the 1,143 total overlapping edges and obtained a score of 0.74 and 0.72, respectively¹⁰. These scores show a high correlation between MusicBrainz and our approach, which can confirm the veracity of many relations in the *songfacts.com* website. This could suggest that a combination of both knowledge bases might increase the completeness of metadata in MusicBrainz. The assessment of such assumption is though left for future work.

⁶<http://musicbrainz.org/>

⁷http://musicbrainz.org/doc/Development/XML_Web_Service/Version_2

⁸<https://musicbrainz.org/relationships>

⁹the votings are available at <http://goo.gl/u0Gjlo>

¹⁰the individual precision and recall scores are available at <http://goo.gl/C4Coj3>

5. CONCLUSIONS

In this paper we presented a method for knowledge acquisition for music that exploits information extracted from unstructured text sources. The method identifies music-related entities in the text and extracts relations between these entities using a rule based approach. We tested our method with a dataset gathered from *songfacts.com*, an online database of facts and stories about songs. We evaluated the extracted relations from a linguistic perspective and the extracted knowledge by comparing it with an existing knowledge base. Experimental results showed that our method is able to extract relations with a high linguistic and conceptual precision.

Still, there are many avenues for future work. Although the evaluation of our relation extraction system shows good values in terms of precision, recall is low between several pairs. We find that an exploration of additional techniques to extract and represent relations is a motivating and encouraging field of research. A successful system should contribute dramatically to the improvement or creation from scratch of music knowledge bases, just as the MusicBrainz example described in this paper. All in all, this method can constitute the foundations for more elaborate approaches to domain-specific knowledge extraction and formalization.

6. ACKNOWLEDGMENTS

The authors would like to thank Miguel Ballesteros for his valuable advice.

7. REFERENCES

- [1] I. Augenstein, D. Maynard, and F. Ciravegna. Relation extraction from the web using distant supervision. In *In proceedings of the 19th International Conference, EKAW 2014*, 2014.
- [2] M. Ballesteros and J. Nivre. Going to the roots of dependency parsing. *Computational Linguistics*, 39(1):5–13, 2013.
- [3] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open Information Extraction from the Web. In *International Joint Conferences on Artificial Intelligence*, pages 2670–2676, 2007.
- [4] B. Bohnet. Very high accuracy and fast dependency parsing is not a contradiction. In *Proc. of the 23rd International Conference on Computational Linguistics*, pages 89–97, 2010.
- [5] C. Brewster, H. Alani, S. Dasmahapatra, and Y. Wilks. Data Driven Ontology Evaluation. *International Conference on Language Resources and Evaluation*, 2004.
- [6] R. C. Bunescu and R. J. Mooney. A shortest path dependency kernel for relation extraction. In *Proc. of the conference on HLT/EMNLP*, pages 724–731, 2005.
- [7] A. Carlson, J. Betteridge, R. C. Wang, E. Hruschka Jr, and T. M. Mitchell. Coupled Semi-Supervised Learning for Information Extraction. In *Proc. of the third ACM WSDM*, pages 101–110, 2010.
- [8] J. Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213, 1968.
- [9] K. Dellschaft and S. Staab. On How to Perform a Gold Standard Based Evaluation of Ontology Learning. *Lecture Notes in Computer Science. The Semantic Web ISWC 2006*, 4273, 2006.
- [10] K. Eichler, H. Hensen, and G. Neumann. Unsupervised relation extraction from web documents. In *Proc. of the 6th International Conference on Language Resources and Evaluation*. ELRA, 2008.
- [11] A. Fader, S. Soderland, and O. Etzioni. Identifying Relations for Open Information Extraction. In *Empirical Methods in Natural Language Processing*, 2011.
- [12] A. Gangemi. A Comparison of Knowledge Extraction Tools for the Semantic Web. In *The Semantic Web: Semantics and Big Data*, pages 351–366. 2013.
- [13] D. Gruhl, M. Nagarajan, J. Pieper, C. Robson, and A. Sheth. Context and Domain Knowledge Enhanced Entity Spotting In Informal Text. In *The Semantic Web-ISWC*, pages 260–276. 2009.
- [14] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 541–550, 2011.
- [15] J. Landis and G. Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.
- [16] M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. M. Gómez-Berbís. Named Entity Recognition: Fallacies, challenges and opportunities. *Computer Standards & Interfaces*, (5):482–489, 2013.
- [17] Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni. Open Language Learning for Information Extraction. In *Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012.
- [18] P. N. Mendes, M. Jakob, A. García-silva, and C. Bizer. DBpedia Spotlight : Shedding Light on the Web of Documents. *Proc. of the 7th International Conference on Semantic Systems*, 2011.
- [19] S. Oramas, M. Sordo, and X. Serra. Automatic creation of knowledge graphs from digital musical document libraries. In *Conference in Interdisciplinary Musicology*, Berlin, 2014.
- [20] I. Serra, R. Girardi, and P. Novais. Evaluating techniques for learning non-taxonomic relationships of ontologies from text. *Expert Systems with Applications*, 41, 2014.
- [21] M. Surdeanu, R. Johansson, A. Meyers, L. Màrquez, and J. Nivre. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proc. of the 12th Conference on Computational Natural Language Learning*, pages 159–177, 2008.
- [22] L. Tesnière. *Elements de syntaxe structurale*. Editions Klincksieck, 1959.
- [23] M. Wang. A re-examination of dependency path kernels for relation extraction. In *IJCNLP*, pages 841–846, 2008.
- [24] W. Xu, R. Hoffmann, L. Zhao, and R. Grishman. Filling knowledge base gaps for distant supervision of relation extraction. In *In Proc. of the Annual Meeting of the Association for Computational Linguistics*, 2013.