Common Lisp for a Common Cultural Recommendation System

Creating a Cultural Recommendation & Exploration Engine Using Semantic Web

Emre Sevinç
İstanbul Bilgi University
emres@cs.bilgi.edu.tr

Remzi Emre Başar
İstanbul Bilgi University
reb@cs.bilgi.edu.tr

Boran Puhaloğlu
İstanbul Bilgi University
boranp@cs.bilgi.edu.tr

Abstract

Current recommendation systems such as Last.FM, Pandora, Amazon.com, etc. use a limited amount of meta-data and collaborative filtering in order to recommend new items to their users within a single domain, be it music albums, books or movies. There is also progress in the field of combining semantic web technologies such as FOAF and music ontologies to create better and more useful recommendation systems.

We think it is possible to go beyond these approaches and create a broad cultural recommendation system. For example, in the domain of music, by taking album and track based meta-data, using RDF and OWL to describe them and then using semantic reasoning techniques, a much more sophisticated and interesting cultural recommendation system can be designed. Such a system will not only recommend new musical items but also related cultural items such as books and movies. In addition to the final result of recommending an item, the system will also provide a justification, showing the meaningful links between those cultural items. The system will also be adaptive, it will be able to adapt itself according to every user with the help of the users (explicitly) or using inductive logic programming methods (implicitly). In our paper we provide justification for our system and summary of how such a cultural recommendation system can be built using Common Lisp, existing semantic web and logical reasoning technologies. In our live demo we will also give examples using the prototype based on the mentioned ideas.

Categories and Subject Descriptors Demonstration [*subcategory*]: application

Keywords semantic web, music, domain-specific languages, entertainment technologies

1. Introduction

Recommendation systems became a very important part of our digital and interconnected life. Netflix, Amazon.com, Last.fm, and Pandora are just some of the more famous ones who rely on various recommendation algorithms to achieve higher levels of user satisfaction. The main theme of those systems is recommending more items that will hopefully be accepted by the user as similar and relevant to the item that the user is engaged with. The recommendations produced by those systems are limited to a single domain most of the time: Neither Amazon.com recommends music albums based on your reading history, nor Last.fm invites you to read some relevant books by looking at your listening trends or Netflix recommends some books to you based on your DVD rental history. This is pretty reasonable for those systems; they focus on a domain and try to sell more and more items in that domain. Most of them, probably excluding Amazon.com, have got nothing to do with the general concept of a fully transparent, adaptive, cultural recommendation system. However businesses such as Amazon.com can make use of such a system in order to gain more customers and drive their satisfaction level to a higher level leading to more profits.

Cultural artifacts have semantic connections based on the production relationships of artists who create them. For example, screenplays are adapted from novels, poems and their translations are used as lyrics, musicians collaborate on various musical albums, they create soundtracks for movies, are influenced by and influence other musicians, authors, poets, directors, etc. Whenever a person interacts a cultural artifact such as a book, a musical track or a movie she may be made aware of vast connections to other items within the same domain and other domains as well.

2. Drawbacks of Collaborative Recommendation

Even though collaborative recommendation systems are known to provide high levels of user satisfaction in various cases, there are also some disadvantages using those statistical techniques, such as the cold-start problem (what to recommend to a new user without any history in the sys-

tem), the novelty detection problem, the item popularity bias (popular items being recommended solely on the grounds of being popular), and the amount of huge data needed to get reasonable and relevant results [1].

Then there is the problem of **groupthink** which is connected to the item popularity bias above and which leads to stereotypes, thus preventing cultural diversity. A cultural recommendation system which can explore and present items to its users can help them to have a broader cultural perspective.

3. Properties of a Cultural Recommendation System

Currently we are developing a common cultural recommendation system with keeping the following properties in mind:

- A user at a given time is engaged with one of the following cultural items: a book, a film or a musical piece.
- The system finds related items, e.g. a book related to a music track, another album related to the current album listened, etc.
- Results are grouped according to the domain and sorted according to the distance between the original item and the found items.
- The system presents the results to the user.
- The user may provide feedback to the system by providing information regarding what he or she thinks about the results.
- The user can direct the system to explore more of the cultural network and provide feedback about these extra results.
- The user may provide any number of cultural items and ask the system if they are connected in any way and if so, how they are connected.

4. Building and Querying a Semantic Knowledge base

4.1 Tools and concepts

The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications which is designed to encode statements about digital resources [2]. The basic idea is to encode the statements as triples: subject, predicate, and object. For example, one way to represent the statement “Bach is the composer of Brandenburg concertos” in RDF is to use a triple: a subject denoting “Bach”, a predicate denoting “is the composer of”, and an object denoting “Brandenburg concertos”. A collection of RDF-encoded statements forms a labeled, directed multi-graph which is a very suitable representation for arbitrary facts in the real world. RDF can be and is also used to encode ontologies which are formal representations of sets of concepts within specific domains and the relationships between those concepts. Once an ontology is created it is possible to do automatic semantic reasoning using powerful methods such as ones based on Description Logic [3].

A knowledge base that is encoded according to the rules of an ontology which uses RDF triples can be stored in a file, a relational database or a triplestore. A triplestore is a kind of database engine designed and optimized for the storage and retrieval of many RDF triples. Such a database engine can use its proprietary query language or one of the standard query languages proposed by W3C.

In order to build a cultural recommendation system with the properties mentioned in previous section we need a system that is capable of dealing with ontologies, querying semantic networks that are encoded using those ontologies and doing logical reasoning over them. AllegroGraph is a modern, high-performance Common Lisp based graph and semantic web system. It uses disk-based storage, enabling it to scale to billions of triples. AllegroGraph supports SPARQL, RDFS++, and Prolog reasoning within Lisp programs [1]. It is a fully functional graph processing system which is not limited to semantic web but can handle any kind of graph, e.g. doing social network analysis over graphs which represent relationships among peoples and social groups (such as page-rank centrality [2]). It is our tool of choice to create a cultural recommendation engine in Lisp.

4.2 A simple query example

To clarify our ideas, here is an example which does not include any special ontology or RDF syntax (for the sake of simplicity and which can be trivially converted into such a format) and solely relies on Prolog capabilities of Allegro Common Lisp and AllegroGraph: We took a famous Turkish music group “Ezginin Günlüğü” who released an album “Oyun”. That album contained a track titled “Vazgeçtim” and the Turkish lyrics of that track were actually a translation of a sonnet from Shakespeare (Sonnet 66 from Sonnets, translated by a Turkish poet, Can Yücel). Let’s take a very small knowledge base which includes facts about the cultural world of books, authors, albums and artists:

```
(<-- (book sonnets))
(<-- (part-of sonnet-66 sonnets))
(<-  (book the-old-man-and-the-sea))
(<-- (author-of shakespeare sonnets))
(<-  (author-of hemingway the-old-man-and-the-sea))
(<-- (album oyun))
(<-- (artist-of ezginin-gunlugu oyun))
(<-- (track-of vazgectim oyun))
(<-  (track-of bekle-beni oyun))
(<-  (track-of bir-eflatun-vakti oyun))
(<-  (track-of kimbilir-ne oyun))
(<-  (track-of kul-vakti oyun))
```

[1] http://agraph.franz.com/allegrograph/

[2] http://agraph.franz.com/support/documentation/current/reference-guide.html#page-rank-centrality

```
(<-- (translation-of vazgectim sonnet-66))
(<-- (translator-of can-yucel sonnet-66))
```

Now let's define an explicit rule which tries to find connections to the related authors, given an album. We encode the rule in the following manner: if the given album exists, and there's a track of an album, and that track is a translation of some cultural item and that item is a part of a book and that book has an author then the author is connected the album we are interested in:

```
(defun find-related-authors-for-album (album)
  (let ((e))
    (prolog
     (lisp ?album album)
     (track-of ?track ?album)
     (translation-of ?track ?translation)
     (part-of ?translation ?book)
     (author-of ?author ?book)
     (lisp (setq e (cons ?book ?author))))
    e))
```

Running this simple query gives the result:

```
CL-USER(140): (find-related-authors-for-album
                            'oyun)
(SONNETS . SHAKESPEARE)
```

The above example is only one of the possibilities that can be realized on that kind of semantic knowledge base. We can do different things and ask different questions: We can ask the system if there is a connection between the album we listen and an author we know, e.g. Shakespeare; we can ask if that connection is direct and if not how many and what kind of relationships it takes to connect both; we can use depth-first or breadth-first search to investigate possible connections between tracks we listen, books we may read and DVDs we may watch.

Since there is a lively and fruitful effort to encode data related to music, literature and cinema domains in a machine readable format as exemplified in [4], it is possible to combine various data sources and provide a uniform interface for asking interesting questions.

4.3 Current Data set and Implementations

Our present data set contains approximately 400 (mostly Turkish) music albums, 5500 Turkish movies, 58000 books in Turkish (original and translation) and 45000 Turkish and foreign artists encoded as RDF triples. There are about 400000 triples in our triple store. Music albums have the most detailed metadata in terms of person, album and track relationships whereas metadata regarding movies and books are more limited.

Every relationship is given a weight as an integer value based on its frequency in our knowledge base, e.g. the "composer" and "artist" relationships have higher weights than "backvocals" and "synthesizer programming". These weight values are used to calculate weighted paths that connect two cultural items based on production relationships.

Currently we are using a simple graph traversal function which, given a node (as a cultural item), collects neighbouring nodes and then prunes the collected nodes based on their weights, e.g. if a musical track is given and there is a relationship with an artist and the relationship is of "congo drums" type and the weight of the relationship is below a critical threshold then this is eliminated. The function continues to work for a predetermined number of levels to collect recommendation paths along with their sum of weights. They are later sorted by their weights and then the end nodes of the paths are given as recommended along with showing the explicit relationship with the initial items.

In addition to giving weights to relationships we are creating a metric of centrality for artists in our triple store so that we can utilize how connected an artist is while constructing recommendation paths. As for the weights of the relationships we also plan to get user feedback and integrate it to the system so that semantic relationships between cultural items and artists can have different values for different users. Even though representing user preferences and doing reasoning about them are known to have issues regarding comparability, combinatorial complexity we think it is important to have how the recommendations produced by the system are evaluated by the users [6].

5. Conclusion

For a full fledged prototype implementation we are using a limited subset of the music ontology defined in [5] [3]. In addition to that we combine book and movie items using bibliographic ontology specification [4] and IMDB ontology specification [5]. There are many publicly available RDF-encoded musical album and culture knowledge bases which can be augmented by the data we compiled for Turkish albums, e.g. dbpedia.org [6] and dbtune.org [7].

We also plan to investigate search strategies as well as user adaptation strategies and recommendation performance of our system based on some metrics such as cultural diversity, recommendation relevancy, run-time performance, the effect of knowledge base contents on the overall quality of recommendation.

We are going to discuss the results of our ongoing work during and after the demo of our prototype in the conference.

[3] Online at `http://purl.org/ontology/mo`

[4] Online at `http://purl.org/ontology/bibo/`

[5] Online at `http://www.csd.abdn.ac.uk/ ggrimnes/dev/imdb/IMDB.rdfs`

[6] Online at `http://dbpedia.org`

[7] Online at `http://dbtune.org`

References

[1] Celma, O. 2006. “Foafing the Music: Bridging the Semantic Gap in Music Recommendation.” In Lecture Notes In Computer Science, 2006, Springer-Verlag, Germany, NUMBER 4273, 927-934

[2] Lassila, O., and Swick. 1998 “Resource description framework (RDF) model and syntax specification.” 1998. [Online] Available: http://citeseer.ist.psu.edu/article/lassila98resource.html

[3] Baader, F., Calvanese, D., McGuiness D.L., Nardi, D., Patel-Schneider, P.F. 2003 “The Description Logic Handbook: Theory, Implementation, Applications”. Cambridge University Press, Cambridge, UK, 2003.

[4] Passant, A., and Raimond, Y. 2008. “Combining Social Music and Semantic Web for Music-related Recommender Systems” In Proceedings of the First Social Data on the Web Workshop, Karlsruhe, Germany, October 27, 2008, CEUR Workshop Proceedings, ISSN 1613-0073, online CEUR-WS.org/Vol-405/paper-5.pdf.

[5] Raimond, Y., Abdallah, S., Sandler, M., Giasson, F. 2007. “The Music Ontology”. Proceedings of the International Conference on Music Information Retrieval, 2007.

[6] Walsh, T. 2007. “Representing and Reasoning with Preferences”. AI Magazine, Vol. 28, Number 4, 2007.