Skipforward—a lightweight ontology-based peer-to-peer recommendation system

Malte Kiesel
DFKI GmbH, Kaiserslautern, Germany
malte.kiesel@dfki.de

Sven Schwarz
DFKI GmbH, Kaiserslautern, Germany
sven.schwarz@dfki.de

ABSTRACT

Skipforward is a distributed recommendation system using a lightweight ontology approach for formalizing opinions about item features. Items can be things such as songs or board games; describing and discriminating item features are the genre of a song or the degree of chance in a board game. Every user of the system is free to add new items and statements about existing items into the system. Naturally, opinions may differ between users—the system even encourages people to express dissent by supporting negation for item features. Skipforward allows discussions for any item feature as well as displaying these discussions in a way similar to web forums.

Introduction

One thing the world wide web is freqently used for is finding out which items (songs, books, movies, programs) might be interesting for somebody—it acts as a vast database containing lots of reviews and product information. Vendor information and shop websites get complemented by lots of forums that provide means to voice yourself and store reviews written by you. Typically using a search engine with the appropriate terms reports at least dozens of hits regardless of the publicity of the item you were searching.

Unfortunately, in the last years things have been becoming more complicated. Searching for reviews for an item is not as easy as it has been before as nearly every single webshop allows to write reviews, polluting the relevant keywords even if no single review actually exists in the shop, making existing reviews hard to find. If one has found actual reviews, they often use different metrics or miss certain aspects. In any case, one has to manually wade through dozens of websites, reading every page, and try to integrate the information manually. Once this is done, you probably have a good overview of the things you are interested; what if you found out some things other people obviously missed? Say you found a wrong fact in a web forum. The best thing you can do about that is adding a reply to that posting, explaining why it's wrong. However, there is no guarantee that your comment will not be overlooked in the future (keep in mind it's probably way down in the discussion thread).

What is needed is a more formalized approach to write, digest, and correct reviews. It should be possible to correct and add to other people's reviews in a way that additions automatically get aggregated, making it superfluous to read through lots of reviews manually. Also, a way to find interesting items without lots of manual work is desirable.

In the following, we outline the *Skipforward* system[1] which is available as open source. Its main purpose is to allow discussing items (songs, books, movies, games, etc.) by voicing opinions about item features. The system as a whole can act as a recommendation system, similar to for example Pandora[2]. It combines ontology-based personal annotations with personalized recommendations, using a peer-to-peer metadata storage system.

Skipforward system outline

Skipinions, the top level ontology. – This is the main building block for formalizing opinions in Skipforward by ways of assigning *features* to *items*. For example, (song) features are `Jazz (Genre)` or `Aggressive (Mood)`. Technically, feature types are represented RDFS classes and, therefore, use a hierarchical structure. Feature instances (in other words, *opinions*) are RDF instances of the respective class. For every feature instance, the user can give an applicability and a confidence rating as well as a link to another feature instance and a comment. This way, dissent with another user's opinion can be expressed, and simple conversation threads about features can be established: If user A creates a feature instance of type `Aggressive (Mood)` and applicability +1, user B may counter this by creating another instance of the same type, but applicability –1. This also mirrors the open world assumption Skipforward embraces on application level.

Fact databases based on Skipinions can be merged easily, keeping the provenance of facts through URI conventions.

Domain ontologies. – The Skipinions ontology only defines basic concepts and cannot be used by itself for any real items. Instead, domain ontologies extending Skipinion's concepts are used. Typically, domain ontologies reuse existing ontologies (e.g., the Music Ontology[3]). Using domain ontologies, expressing facts such as *This song features an*

[1] http://skipforward.opendfki.de/ — The system's name is due to its initial focus as a music recommendation system.
[2] http://www.pandora.com/
[3] http://musicontology.com/

electric guitar solo or *The element of chance is very prominent in this board game* is possible. Feature types are RDFS classes and use a class hierarchy; in the system, RDFS inference is used, so a song associated with a feature instance of type `Very aggressive (Mood)` is also found when searching for `Aggressive (Mood)` (assuming appropriate subclass relations exist). Extending domain ontologies by creating subclasses is possible within the system.

Metadata exchange protocol. – A Skipforward node/client is expected to run on the user's machine. A number of clients form a peer-to-peer-system. XMPP[4] is used as a basis for metadata exchange. Peer nodes are the users present in the XMPP contact list. Every user runs a Skipforward node that connects to the user's XMPP server. The Skipforward client can request and send metadata using the Skipforward metadata exchange protocol, which, in turn, is based on XMPP. We plan to make the system's metadata also available via HTTP, implementing the *Linked Data*[5] paradigm.

Recommendation component. – Users may express liking or disliking items and item features (e.g., *I like this song* or *I do not like the feature* `Endless Screaming`). Based on this information, a weighted set of item features is built. The system then can compute the difference of this set to the aggregated set of features present for each item. The items that match best are being recommended to the user as being potentially interesting (*content-based filtering*). In the demo, aggregating feature instances of the same type (e.g., several people commented on the feature `Aggressive (Mood)`) for one item is done in a naive way, just calculating the arithmetic mean of applicability ratings for each instance. This component is being extended to compute and take into consideration peer competence optionally: Opinions by users who created annotations that match my annotations get a higher weight when calculating recommendations than opinions by people typically disagreeing with me. This allows handling subjective item features and is expected to discourage spamming.

Skipforward user interface. – The Skipforward user interface is implemented as a web application using the Dojo Javascript framework[6]. We chose the web-based approach since this allows running the client on an (own) web server continuously, avoiding problems with off-line peers[7]. The user interface allows browsing items stored in the system, annotating items, "replying" to annotations created by other users of the system, searching for items by features (*Which games use dice?*), configuring user preferences (*I like the game* Settlers *and in general games that use hex-tiled boards, but I do not like card games*), and presenting recommendations generated on basis of these preferences. Additionally to the hierarchical feature type view, Skipforward allows grouping features types into *certificate sets*. This allows displaying icons when certain annotation levels have been reached, e.g., *All basic metadata—author, material, publisher—of this item has been entered.* It also helps breaking down entering metadata into smaller steps, e.g., *In order to get the next certificate for this item, enter statements regarding feature types X and Y.* The user interface supports internationalization, especially for the metadata entered.

Related work

DBin[8] is similar to Skipforward but more generic and heavyweight. For example, it comes with its own messaging API, a plug-in architecture for its user interface, and needs dedicated metadata servers as well as a Java client.

SIOC[9] is a metadata format that can be used to represent information of web forums (discussions, users, ...) in RDF. In Skipforward, this is currently done in a simplistic way (feature instances double as discussion entries). Using SIOC for modelling discussions while keeping connections to features/opinions will be considered in Skipforward's future.

Revyu[10] is a metadata format and website where users can post reviews. Reviews can be tagged with keywords; absolute ratings can be given to items. Metadata is available as RDF. Compared with Skipforward, Revju does not feature fine-grained reviews (*User A says item I has feature X; user B disagrees*).

Foafing the Music[11] is a recent music recommendation system that combines FOAF profiles, RSS feeds, and last.fm data to generate music recommendations. However, this is a kind of read-only mashup while Skipforward heavily empathizes user contributions. Also, Skipforward focuses on content based filtering whereas Foafing the Music focuses on collaborative filtering.

Bouillon 2[12] implements a "peer-to-peer wiki" using Jabber/XMPP. Wiki text parts you give a good rating are weighted high for your friends (i.e., people in your contact list) and vice versa. Thus, there is no 'canonical' view on the contents of this wiki but there exist many subjective views, one for each 'community', which is similar to the idea Skipforward pursues.

Traditional music recommender systems such as *last.fm* typically only feature collaborative filtering (*Users who liked this song also liked song X*). Content-based filtering, possible with the more rich metadata available in Skipforward, allows more informed recommendations and more directed search. Compared to content-based systems Skipforward allows user participation while, for example, *Pandora* employs a rigid annotation model and a facts database that only a number of chosen experts can contribute to.

[4]Extensible Messaging and Presence Protocol (formerly *Jabber*), a protocol for routing arbitrary XML, primarily used for instant messaging. http://www.xmpp.org/
[5]http://esw.w3.org/topic/LinkedData
[6]http://dojotoolkit.org/
[7]Off-line peers are no showstopper in Skipforward but still present a slow start problem.

[8]http://dbin.org/
[9]http://www.w3.org/Submission/2007/02/
[10]http://revyu.com/
[11]http://foafing-the-music.iua.upf.edu/
[12]http://oc-co.org:8000/