COMP6803 — Project Proposal

Scott Jackson (41794142)
Due: April 1, 2011

1. Topic Definition

This project deals with how we might be able to use the semantics behind what people Like online to provide them with better recommendations for new content. The rest of this section of the proposal is dedicated to explaining exactly what that means.

1.1 Liking

As the internet use has grown more and more in recent years, websites have become more advanced. One particular kind of website that has exploded in recent years is the social network. A common pattern on social networks is the practice of “Liking” a unit of content. The terminology varies between sites (on Facebook and Tumblr, users Like things; on Twitter and Flickr, users Favorite things; on Last.fm, users Love things), but the idea is the same — users click a button next to a thing that they like and the website registers that they like that thing. For the remainder of this proposal, “Like” is considered to mean the general action, applicable to any of the sites mentioned above (unless a specific site is being discussed at the time).

As web technologies have become more mature, social sites have begun implementing Application Programming Interfaces (APIs) which expose a site’s functionality so that others may build some of the site’s functionality in to their own site. By way of example, APIs allow someone who runs a news site to put a Like button that connects with Facebook on their site. Now, Facebook users who browse news articles on the news site can click that Like button and have those Likes show up on Facebook. APIs allow social sites to reach outside their own domain. People online are Liking things in droves, in no small part thanks to the APIs that have made Like buttons ubiquitous. As the number of Likes online continues to grow it is becoming more clear each day: Liking content is a phenomenon that is here to stay.

1.2 Recommendations

Whether it’s seeing a movie because a friend tells you they enjoyed it, or reading a book because you enjoyed its author’s previous book, recommendations have always played a part in how people discover new content. With the wealth of information online, we can start to make some sophisticated recommendations —

we can recommend a movie not just based on what one of your friend likes, but what all of your friends like. In addition, we can base recommendations not just on information about what's available at your local (on-the-brink-of-extinction) video store or showing at your local cinema, but on vast archives of information about seemingly every motion picture ever made.

1.3 Introducing Semantics to Likes

One interesting question to ask is: why do people Like the things that they do? The obvious answer is right there in the question — a person Likes something because they like something. However, there are more nuanced motivations at play. Perhaps someone Likes a joke they found funny on Twitter. Maybe someone else Likes a photo their daughter posted on Flickr because they haven't seen each other for a couple of years. Perhaps someone Likes their friend's blog post on Tumblr because they empathise with their situation. Maybe someone else Likes a quote from a TV show that their friend posted to Facebook to let them know that they get the reference, that they like the show too. In each of these cases, each person took the same action, but did so as a way of showing something different. What would be four vastly different reactions in the real-world[1] have been reduced to a single, one-dimensional response — you Like it.

The situation gets even more interesting when we consider content not generated by people — in each of the situations above, Liking content can be viewed as a way of giving feedback to the author of the content. What happens when there *is* no author? Who are you giving feedback to when you say that you Like Coca-Cola on Facebook? What end is Liking Coca-Cola a means to? I certainly don't know, but 23.9 million people (and counting) Like Coca-Cola on Facebook, so surely *some* of them are doing it for a reason.

If we have a recommendation system that recommends someone content based on what other people Like (see section 4 — Review of Background and Related Work), knowing why people Liked the things that they Liked could provide us more insight into their taste, and perhaps allow us to provide better recommendations. That is the overall topic of this project.

---

[1] in order: laughing at a joke a friend makes, missing a daughter when she sends you a picture, commiserating with a friend, and acknowledging a shared interest in a TV show

2. Goal

The goal of this project is to explore how we can use semantics as a way of enhancing social filters for recommending new content to users. In addition to typical recommendation strategies like social and content filters (see Review of Background and Related Work), we try to take into account not only *what* content people Like online, but *why* they Like that content. By incorporating some semantics into the recommendation algorithm, we aim to provide better, more nuanced recommendations to users.

In terms of deliverables, the project is made up of two major products.

The first major product is a webapp that learns a user's taste. This product lets users opt-in to having the things that they Like tracked. The webapp consists of two subcomponents: the "tracker" and the survey. The tracker passively watches and records what users Like across various social sites (users opt-in to however many sites they feel comfortable with), providing us with a large set of data describing *what* people Like. What's more, modeling users' taste on what they Like across multiple sites allows us to get a more broad sense of what they Like. The second subcomponent, the survey, takes a more active approach to gathering information. The survey ideally consists of a browser extension[2] that asks the user a couple of questions whenever it detects that they Liked something on a social site. The questions aim to find out why the user Liked that content. If, for some reason, this automatic behaviour isn't technically possible to implement, the survey will consist of a bookmarklet that the user activates whenever they Like something. The bookmarklet asks them the same questions as the browser extension does. The main difference between the browser extension approach and the bookmarklet approach is that the bookmarklet approach requires users to actively remember to click the bookmarklet when they Like something, whereas the browser extension shows them the questions whenever they Like something. The bookmarklet is a fallback because the user response rate may be lower.

The second major product is a webapp that recommends content to a user. This webapp allows users to sign in with various social sites and have new content recommended to them (with their history on those sites taken into account for the recommendations). The recommendation webapp uses data gathered from the taste-tracking webapp as a base for its recommendations. Building and tweaking this product will make up the majority of the work in the second semester.

---

[2] for Safari, Firefox, Chrome, and (time permitting) Internet Explorer

3. Relevance

This project's relevance grows by the day — with more information online than ever before, wading through content to find the good stuff is getting more difficult[3]. As a result, users need to employ filters to allow them to find what they're looking for.

In addition to the need for such a filter, the capabilities of the tools with which we can build filters are growing as well. With widely accepted standards like XML and JSON at our disposal, we can make a whole lot of social sites speak the same language. This wasn't imaginable, let alone implementable, as little as five years ago. There has never been a better time to build web applications that harness data acquired from social sites.

All of this is probably obvious, though. The real relevance of this project has to do with the limitations of current recommendation systems. Current recommendation systems that rely on social filters use what a user's friends Like to show users new content. The problem is that, as we saw before, not all Likes are created equally. Different Likes mean different things, but recommendation systems don't take this into account. In addition, most (if not all) current recommendation systems base their recommendations on data within their own domain — Amazon recommends products based only your Amazon browsing history, for example. The recommendation system developed in this project doesn't have a vested interest in recommending content from any one particular site, so we can recommend content from a range of sites.

4. Review of Background and Related Work

Though the tools we deal with in this project are cutting-edge, the problems surrounding recommendations have been studied for a long time. Two prevailing methods of choosing which content to recommend to users are social filtering and content filtering. In this section of the proposal, we look at some of the work surrounding both of these methods.

[3]The situation only gets worse when we also consider how many spammers and scammers are out there. The problem of filtering content is serious enough without taking into account people with malicious goals.

Hill, Stead, Rosenstein and Furnas (1995) defined social filtering as "using other people as filters and guides: filters to strain out potentially bad choices and guides to point out potentially good choices." People have been manually using social filtering as a way of discovering new content for years — when we ask a friend with similar taste to our own what movie we should see, they make a recommendation, and we trust that recommendation because the two of us are like-minded (Basu, Hirsh, Cohen, 1998). However, before the internet, geography was a limitation on the effectiveness of social filtering. The community of people that would act as a social filter was only made up of the people that you knew, and you were limited to only knowing the people around you. The internet changed this by allowing the community of people used as a social filter to be made up of people with similar interests, not similar home addresses. By removing limitations like geography and having to personally know other members of the community, "social information filtering essentially automates the word-of-mouth recommendations. Except that instead of having to ask a couple friends about a few items, a social filtering agent can ask thousands of other people, and consider thousands of different items, all happening autonomously and automatically" (Shardanand, 1994).

In a general sense, a social filter works in the following way: "the user of the system provides ratings of some artifacts or items and the system makes informed guesses about what other items the user may like" (Basu, Hirsh, Cohen, 1998). Our project's recommendation system will relieve the user of having to actively provide ratings for known content — we simply use the history of what they have Liked across various social sites as a basis for the informed guesses about what other content they might like.

One early example of an online social filter was Hill, Stead, Rosenstein, and Furnas' videos@bellcore.com, an email service that would recommend movies to users. The system worked by first sending users a list of 500 movies. The user then gives the movies on the list that they've seen a rating between 1 and 10 (1 being lowest, 10 being highest) and sends those ratings back to the system. The system then adds the user's ratings to the existing database of ratings for those 500 movies, and computes some recommendations for new movies to watch based on what what other people who similarly rated the same movies enjoyed. The system sends back a list of the movies recommended for the user, as well as a predicted rating for each movie. (Hill, Stead, Rosenstein, Furnas, 1995)

videos@bellcore.com was tested in two ways: user feedback and a cross-validated correlation study. The cross-validated correlation study is of particular interest to us, since it is a method of testing recommendations that came up a lot in review of literature. Basically, we remove a small portion of user's ratings from the pool of data and see what the recommendation system predicts users' ratings for that removed content. We compare the predicted ratings to the actual ratings — if the ratings are similar, the recommendation system can predict the users' taste. (Hill, Stead, Rosenstein, Furnas, 1995)

videos@bellcore.com was a success — over a sample size of 291 participants and 55,000 ratings, three out of four recommendations were highly rated by users. The authors of the system compared videos@bellcore.com with the effectiveness of recommendations by nationally-known movie critics, and videos@bellcore.com came out on top. In addition, 32 out of 51 voluntary open-ended feedback responses were positive. (Hill, Stead, Rosenstein, Furnas, 1995)

A second example of a social filter in use was Shardanand's "Ringo," an online music recommendation service developed at MIT in 1994. Like videos@bellcore.com, Ringo users used email to join the service. Once signed up, users could access the system through a graphical web browser. As with videos@bellcore.com, users needed to initially rate some content (in this case, musical artists) for the system to learn what they like. Once the user had given some initial ratings, Ringo created a profile for the user (featuring content they'd already rated) and allowed users to ask for suggestions of new artists to listen to.

Being a publicly available service, Ringo drew more users than videos@bellcore.com — within a month of being advertised on USENET, over one thousand users had signed up and provided ratings for artists. Ringo used a variation on the Pearson *r* algorithm to measure similarity between users, as Shardanand (1994) found it to be the most effective in predicting how users would rate new content. One key lesson learned in Shardanand's experiment was the importance of critical mass:

> *The system must reach a certain critical mass of people in any particular genre before it works well overall. Likewise, a certain number of "key" artists must be rated by someone before Ringo has a good fix on that person's tastes. The amount of data will vary from problem domain to problem domain. [...] Based on user feedback, we reached "critical mass" for mainstream music after 250 users were in the system.* (Shardanand, 1994)

Overall, Ringo showed that social filtering is a good strategy for determining which content to recommend to users, albeit with the drawback that it needed a certain amount of existing data to provide good recommendations.

A more modern case in which the virtues of social filtering was shown was Kristina Lerman's 2006 experiment involving social news site Digg. One of Digg's interfaces is a "Friends" view (now called "My News") that shows users news stories that their friends Dugg rather than the traditional view (up and coming stories that are becoming popular across the site). In that study, Lerman (2006) found that Digg users tended to Digg stories submitted, read, and Dugg by their friends more often than other stories, thus showing that the social filtering created by the "Friends" interface was a good way of recommending content. Lerman's paper showed that social networks like Digg (and by extension in 2011, Facebook, Twitter, Tumblr, Flickr, etc.) can be used as social filters, a view that was corroborated by Lerman's later paper on Flickr (Lerman, Jones, 2006). These papers and their findings are particularly relevant to this project because they use social networks as the social filter — in other words, it is the taste of the user's friends that is drawn on to recommend new content to the user, not the taste of like-minded people from the entire community.

As we saw in Shardanand's Ringo paper, the major drawback of social filtering is that it relies on a sufficiently large overlap of ratings. Specifically, enough users need to have rated the same content. That way, the system has a method of determining how similar or dissimilar different users are. Basu, Hirsh, and Cohen (1998) use the following example to illustrate the drawbacks of social filtering without an overlap of ratings:

> *Consider the case of an artifact for which no ratings are available, such as when a new movie comes out. Since there will be a period of time when a recommendation system will have little ratings data for this movie, the recommendation system will initially not be able to recommend this movie reliably.* (Basu, Hirsh, Cohen, 1998)

With this drawback in mind we can safely say that social filtering is *generally* a good approach for recommendations. The kind of situation in which social filtering doesn't provide accurate recommendations is when not enough people have rated a common set of content. We now turn our attention to another method of filtering, one that doesn't rely on ratings.

Instead of relying on pre-existing ratings from other users, content filtering (sometimes called collaborative filtering) uses information inherent in the content to recommend similar new content to users. Movies, for example, have all sorts of attributes: genre, director, writer, cast, year of release, etc. By recommending movies with similar attributes to movies the user has already seen and liked, we are using the content's metadata as a filter.

In 2007, Groh and Ehmig pitted social filtering against content filtering to see which would provide more satisfying recommendations. In their empirical study, they found that "social recommenders perform as good (sic) as the best collaborative filter approaches (in some situations (e.g. sparsity) and under some aspects (e.g. novel predictions) even clearly better" (Groh, Ehmig, 2007).

While content filtering is a viable option in recommendation systems built around recommending things like music or movies, a serious problem with content filtering is that it requires content to have a standardised set of attributes (or that such a set can easily be parsed). Since most web content doesn't have a large, reliable set of shared metadata (there's no IMDb equivalent for websites — all we could really use in our recommendation is a domain name[4]), content filtering doesn't really work in unstructured environments. As a result, in our project, we won't expect to deal with content filtering very often. However, in cases where social filtering may not produce great results, we may need to resort to using content filtering. Because of this, it deserves a mention here.

5. Project Plan

The following is a list of tasks that make up the project, complete with a risk assessment and schedule for each task.

Task 1: Production of a tracker
Description: The tracker is a web application that tracks what users Like across various social sites. The tracker is made up of seven subcomponents, one for each of the following sites:

• Facebook (facebook.com)
• Twitter (twitter.com)

[4] That is, we could recommend content based on the fact that it comes from the same site as content a user is known to have Liked.

- Tumblr (tumblr.com)
- Flickr (flickr.com)
- Reddit (reddit.com)
- Last.fm (last.fm)
- Hacker News (news.ycombinator.com)

Each of these subcomponents track what a user likes on a particular site. Once completed, the subcomponents then need to be integrated into a larger web application that periodically checks each site to find out what each user that has opted in to be tracked has liked. This bulk data accumulates over time and is used as a basis for the recommendation engine.
Schedule: Subcomponents of the tracker should be built over the course of the first semester, and the tracker itself should be operational by the end of the first semester of the project.

Task 2: Production of a survey
Description: The survey is a browser extension (for Safari, Firefox and Chrome) that automatically asks users who have the extension installed questions whenever they Like something on various social sites (see the list of sites in the tracker). The aggregation of this data provides some insight into the meaning behind the tracker's bulk data about what people Like. If it turns out to be infeasible to track users in this automated manner, a bookmarklet will be used instead, requiring users to manually bring up the questions. This approach would increase the amount of effort required to answer the questions, and so would probably result in a lower response rate from users. Even if the browser extension is unable to be implemented, the bookmarklet uses almost all of the same code to allow the user to answer questions. The only difference is in how the user is shown those questions (manually in the case of the bookmarklet, automatically for the browser extension). If neither the extension nor the bookmarklet can be implemented, a recommendation engine can still be built that only uses data from the tracker. However, the resulting recommendation engine wouldn't allow us to draw any conclusions as to how taking meaning into account changes the effectiveness of recommendations, giving us nothing to analyse.
Schedule: As with the tracker, the survey needs to be ready to be tested by the end of the first semester.

Task 3: Delivery of a progress seminar
Description: a seminar is to be given detailing the project's progress over the course of the first semester.

Schedule: The seminar needs to be completed by and will be delivered during the week starting May 16, 2011.

Task 4: Gathering of data
Description: The recommendation engine needs data to base its recommendations on, so a period of data gathering has to take place. To gather this data, a group of test users with accounts on various social sites will have their online activity tracked by the tracker for a period of two weeks. In addition, a second group of users will install the browser extension and take a survey whenever they Like something on a social site over the course of two weeks. The data gathered is sent to us and stored on a server, ready for analysis and use in the recommendation engine.
Schedule: Gathering of data will take place over the mid-year holidays.

Task 5: Production of a Recommendation Engine
Description: The recommendation engine is a webapp that recommends new content to users based on what the users have Liked previously across various social sites.
Schedule: The recommendation engine should be built by the mid-semester 2 holidays, so as to allow time for testing.

Task 6: Testing of the recommendation engine
Description: At this stage, the recommendation engine needs to be adapted based on user feedback. The recommendation engine will be tested and modified iteratively — we will generate some recommendations for test users, get their feedback on the recommendations, and try to incorporate what we learn from that feedback into the recommendation engine to make it provide better recommendation. This process repeats either until we are satisfied with the quality of the recommendations or the project is over. The recommendation engine will be tested through two methods — user feedback and a cross-validated correlation study (see Review of Background and Related Work).
Schedule: Testing of the recommendation engine will take place between the mid-semester 2 holidays and needs to be completed by the deadline for the thesis, November 7, 2011.

Task 7: Poster and abstract submission
Description: The poster is a graphical representation of the findings of the project to be presented to other students and academics, and possibly industry and the public. The software produced over the course of the project should be

demonstrated. The abstract should clearly state the project aims and goals, methods used and project outcomes.
Schedule: The poster and abstract need to be completed by October 7, 2011.

Task 8: Oral presentation
Description: The oral presentation is a seminar detailing the key content of the research undertaken in the project, including progress made.
Schedule: The oral presentation needs to be completed by October 27, 2011.

Task 9: Thesis
Description: The thesis is the compilation of all work done during the project, including material on the problems and goals of the project, applicable methods, the approach taken, major decisions and reasons for those decisions, results, and evaluation of the results.
Schedule: The project thesis needs to be completed by November 7, 2011.

6. OHS Risk Assessment

Since this is a research project, it automatically passes an OHS risk assessment.

References

Basu, Chumki; Cohen, William; Hirsh, Haym. “Recommendation as Classification: Using Social and Content-Based Information in Recommendation.” Proceedings of the Fifteenth AAAI Conference on Artificial Intelligence. Madison, WI, USA: AAAI, 1998.

Ehmig, Christian; Groh, Georg. “Recommendations in Taste Related Domains: Collaborative Filtering vs. Social Filtering.” Proceedings of the 2007 international ACM conference on supporting group work. New York, NY, USA: AcM Press, 2007. 127-136.

Hill, Will; Stead, Larry; Rosenstein, Mark; Furnas, George. “Recommending and Evaluating Choices in a Virtual Community of Use.” Proceedings of the SIGCHI conference on Human factors in computing systems. Piscataway, NJ, USA: ACM Press, 1995. 194-201.

Lerman, Kristina. “Social Networks and Social Information Filtering on Digg.” CoRR (Computing Research Repository) abs/cs/0612046 (2006).

Lerman, Kristina; Jones, Laurie. “Social Browsing on Flickr.” CoRR (Computing Research Repository) abs/cs/0612047 (2006).

Shardanand, Upendra. Social Information Filtering For Music Recommendation. Massachusetts, USA: Massachusetts Institute of Technology, 1994.