Norwegian University of Science and Technology

Project Thesis

Improving Movie Recommendations with Unpersonal Social Media Data

Author:
Jonas Myrlund

Supervisor:
Prof. Heri Ramampiaro

November 2013

NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Abstract

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Computer and Information Science

Master of Technology

Improving Movie Recommendations with Unpersonal Social Media Data

by Jonas Myrlund

This is a project thesis about movie recommendations, specifically answering the question: *can movie recommendations be significantly improved by unpersonal social media data?* We look at how we can improve a set of recommendations – through filtering and annotation – by analyzing *the sentiment in social media.* By selecting a set of movies, and comparing the results to those found in open movie rating datasets, we consider whether the sentiment extracted from social media is of significant use in improving movie recommendations.

@TODO What do we find???

Acknowledgements

@TODO: Acknowlegde.

Contents

Abstract i

Acknowledgements ii

Contents iii

List of Figures v

List of Tables vi

Abbreviations vii

Glossary viii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Research Questions . . . 2
- 1.3 Overview and Summary . . . 3

2 Survey 4
- 2.1 Relevant literature . . . 4
- 2.2 Similar applications . . . 5

3 Requirements 6
- 3.1 Relevant Qualities of Twitter Data . . . 6
- 3.2 Twitter as a Data Source . . . 7

4 Design 8

5 Implementation 9
- 5.1 Data retrieval . . . 10
- 5.2 Sentiment analysis . . . 10
- 5.3 Evaluating sentiment . . . 11

6 Evaluation 12
- 6.1 Evaluating against Netflix rating data . . . 12

7 Conclusion 14

A Requirements **15**

B Design Documents **16**

C Evaluation Results **17**

Bibliography **18**

List of Figures

4.1 The basic architecture of a system described in this thesis. We mainly describe the component labelled *A*. 8
4.2 The layout of the system within component A. 8

List of Tables

3.1 Classification of microblogging behavior 7

Abbreviations

SaaS	**S**oftware **a**s **a** **S**ervice
SNS	**S**ocial **N**etwork **S**ervice
SVM	**S**upport **V**ector **M**achine
MSE	**M**ean **S**quare **E**rror

Glossary

Tweet	A Twitter message
DatumBox	A machine learning SaaS

Chapter 1

Introduction

1.1 Motivation

Recommendation systems, as seen in commercial products in 2013, seldom provide any context to go with its recommended products other than statements like "Because you watched X you might like Y."

Netflix, the movie subscription service, quite recently began suggesting movies based on what your Facebook friends had watched – thereby taking a solid step in the direction of social recommendations. These suggestions, however, have one underlying assumption that may not always apply: what the displayed friend has watched is relevant to your choice of content.

Furthermore, the social components in Netflix's recommendations are *personal*. This approach has some clear advantages in that it uses *your* social network as a basis for suggesting content, but there are also some downsides to this:

- They don't take *novelty* into consideration.
- They don't take *hype* into consideration.
- They leverage only a microscopic portion of the available opinions on content that is available in social media.

These personal suggestions seems to be a trend in contemporary approaches to social recommendations. We'll have a look at something a bit different.

There are other sources of social data that do not fit into the "personal" model. In this thesis we'll have a look at one of the largest sources of unpersonal social media content today: Twitter.

Micro-blogging services such as Twitter have enormous amounts of data on almost every topic imaginable. Content is limited in length, and users react to each others' content by "re-tweeting", "favoriting" or "replying to" it. This leaves us with a source of textual data that is:

Instant Users express reactions to events as they experience them.

Weighted Users weigh each others' content by interacting with it.

Concise Due to limitations on content length, users must express themselves concisely.

Furthermore, the Twitter search API[1] supports returning both *popular* and *recent* content, or *a mix* of the two. This enables two interesting approaches to both filtering and annotating the recommended content, in that we can treat popular and recent comments separately.

1.2 Research Questions

In this thesis, I'll mostly examine *improving a set of recommendations* taking social media data into account, and not so much try to generate new recommendations in their own right – although I might make a go of it if the data should prove agreeable.

I will rather look at ways of using copious[2], unpersonal[3] social media data to *filter* sets of recommendations.

More specifically, the aim is to find answers to the following questions:

1. Can sentiment analysis of large quantities of unpersonal social media data be used to effectively filter or provide context to recommendations?

2. Can unpersonal social media data in any way generate reliable ratings of its own?

[1] https://dev.twitter.com/docs/api/1.1/get/search/tweets
[2] *Copious* in the sense that the size of the data source is arbitrarily large.
[3] *Unpersonal* in the sense that the data is not related to a single hypothetical user of the system.

3. How do we evaluate our efforts in order to answer the above questions?

1.3 Overview and Summary

I will look at improving an intermediate step in a hypothetical recommendation pipeline, filtering and/or annotating recommended content. For a more detailed look at the system design, and some reasoning around the parts of the system being touched upon, see chapter 4.

@TODO Add more overview and summary information as it comes into existence.

Chapter 2

Survey

2.1 Relevant literature

The task of improving recommendations through sentiment analysis of social media data requires digging into several fields of study.

Some people have attempted the same task as in this thesis, albeit with another type of social data. Singh et al. [1] investigated a "formulation, where [they] combined the content-based approach with a sentiment analysis task to improve the recommendation results." Their approach is very similar to our approach, but differs in two important ways:

1. It uses user reviews from IMDB as content source[1], and not a more general source of sentiment-carrying content – as in our case, with Twitter.
2. It is not designed to enhance presupplied recommendations, but rather to generate its own based on genre input.

As we're looking at data from Twitter, the sentiment analysis task is a bit different than usual, as it needs to operate on texts that are all less than 140 characters long. More often than not, we will in fact need to work with texts that are merely one or two sentences long. Cho & Kang [2] "propose a method of classifying tendencies and opinions in texts of multiple sentence length extracted from social media and covering

[1] IMDB does not provide open API access at the time of writing.

both formal and informal vocabularies". Among the things the more unusual things they condider when analysing content is posision of each sentence and emotion icons, which is quite important in short, concise text like ones found on Twitter. We'll be taking a closer look at this method for the actual sentiment analysis task.

@TODO More.

2.2 Similar applications

What spawned the idea of using an unpersonal social service like Twitter as a content source for filtering content is that Netflix recently rolled out personal social recommendations of their content. For this they use Facebook.

One huge limitation to the Facebook approach is that Facebook doesn't expose how "close" you are to your various friends.

Content sharing patterns (Social influence and the diffusion of user-created content): [3].

Chapter 3

Requirements

Although the methods described in this paper aims to be as agnostic as possible with regards to what type of sentiment carrying input is used, we have selected the microblogging service Twitter – and the available Twitter data has some specific qualities we'll try to make use of to improve the quality of our results.

3.1 Relevant Qualities of Twitter Data

As previously mentioned, "Tweets" can be *favorited*, *retweeted*, and *replied to*. Additionally, we can tell how big reach an author has by counting the number of *followers* he/she has, and use this as another indication of content popularity.

We want to be able to use the data as a source of implicit ratings. To be able to, we need to quantify the significance of these verbs. Oard and Kim [4, 5] and Kelly and Teevan [6] have developed a framework for classification of implicit feedback. They define three major categories for implicit feedback: examination, retention, and reference.

We adapt it to the domain of Twitter data, and wind up with table 3.1.

To clarify the classifications of table 3.1, let's break the terms down.

Tweet A user posts content to Twitter, in the form of a new post. A Tweet can have a maximum of 140 characters. Due to the size restrictions, tweets often contain links to websites.

Original	**Ours**		**Action**
Examine	Consume	→	Follow
Annotate	Evaluate	→	Reply
Retain	Endorse	→	Favorite
Reference	Forward	→	Retweet

Table 3.1: Classification of microblogging behavior

Follow Users consume each others' content by following each other. The number of followers users have range from 0 to more than 40 million. Following is a one-way relationship, and there is often a big difference in the number of users following and being followed by a user.

Reply Users can mention each other in tweets by prepending a username with "@". This same mechanism is used to reply to others' content. When replying, the content the Tweet was replying to is stored along with the reply, forming a conversation tree.

Favorite Users can favorite content, which notifies the content owner and boosts the content in search results etc. It is also trivial to extract all content a particular user has favorited.

Retweet When a user chooses to retweet content, that content is "forwarded" to the user's followers, and boosts the content in search results etc.

3.2 Twitter as a Data Source

There are vast opportunities in managing to understand a data source with the qualities discussed above, but alas – the diversity and free nature of Twitter as a publishing platform comes at a price: *there is a lot of noise*. That is not to say that there is little relevant information, but when the service is designed to have such a low threshold for contributing content, a low signal-to-noise ratio seems inevitable.

This problem of finding content carrying relevant information turns out to perhaps be the biggest challenge of the entire study.

Chapter 4

Design

The basic architecture is outlined in figure 4.1. Our work will be aimed at the component labelled A.

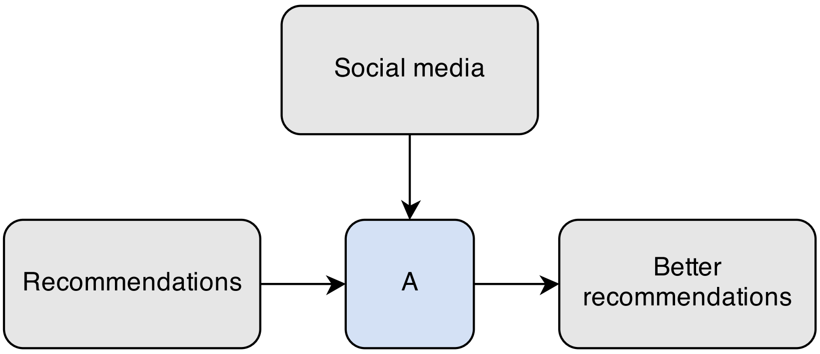

FIGURE 4.1: The basic architecture of a system described in this thesis. We mainly describe the component labelled A.

The A component is structured as illustrated in figure 4.2.

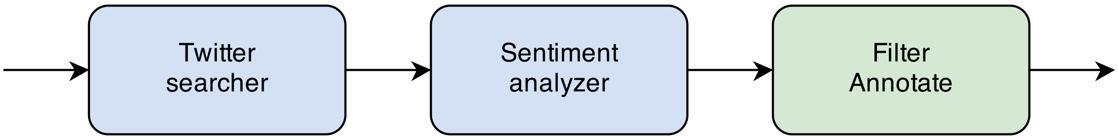

FIGURE 4.2: The layout of the system within component A.

Chapter 5

Implementation

The system does not directly employ any specific algorithms or data structures worth mentioning. The problem lies in determining the applicability of the data harvested from Twitter.

A central part of the system is the sentiment classification of the Twitter results. This sentiment analysis could well have been implemented locally, but for simplicity's sake it has been offloaded to an external service called DatumBox[1], as it seems to employ a reasonable choice of algorithm, and performs well enough for our needs on the short Twitter messages. See section 5.2 for a more in-depth look at the techniques the DatumBox service utilizes.

The system is implemented as a pipeline of sorts, consisting of two main steps, and one evaluation step:

1. Data retrieval – retrieves, cleans and packages the Twitter data into simple classes for easier subsequent use.
2. Sentiment analysis – analyzes each tweet, classifying it as either positive, neutral, or negative.
3. Evaluation – maps sentiment to a final score, and compares these results to those found in external datasets.

[1] `datumbox.com`

5.1 Data retrieval

As discussed in section 3.2, Twitter has a disturbingly low signal-to-noise ratio, so we had to examine every opportunity to refine the data retrieval step. Luckily, the Twitter API, at the time of writing, has a quite extensive search interface, with many ways of tweaking the results in a desired direction.

After much trial and failure, the following settings seem to yield the best results:

- Ensure that the title is searched for in its entirety, not the individual words.
- Exclude tweets containing the following terms[2]: "download", "stream", "nw", "nowwatching", and "RT".

Then remained the choice between the two modes of search: "popular", "recent", or the optional combination of the two.

5.2 Sentiment analysis

As mentioned above, the task of sentiment analysis is offloaded to a SaaS called `DatumBox`. They outline the techniques applied in an article on their service blog [7].

With the approach taken, texts are classified as either positive, neutral, or negative. A training set of 1.2 million tweets were tokenized "by extracting their bigrams and by taking into account the URLs, the hash tags, the usernames and the emoticons", and were subsequently fed into a Mutual Information algorithm for feature selection. The classifier in use is the Binarized Naïve Bayes, after having outperformed SVM, Max Entropy and others on the test set.

With a 10-fold cross-validation, the best performing classifier allegiedly achieves an accuracy of 83.26%.

When calling the DatumBox API, the best results were achieved when removing the movie title itself from the query. Quite a lot of titles have sentiment-carrying words in their titles[3], and this obviously confused the classifier quite a bit.

[2] Any time a set of irrelevant results shared a common term, it would be added to the list. There are probably many ways of fine-tuning this further.

[3] "Breaking Bad" consequently scoring way below "Cheers" was a rather clear cut case.

5.3 Evaluating sentiment

Chapter 6

Evaluation

- When rating popular and well-known movies[1] the predicted ratings achieve a correlational coefficient of 0.75 with regard to average Netflix ratings for the same movie.
- B-movies or older less-known movies rarely collect enough Twitter search results to warrant any further analysis.
- Movies with titles that are fairly common words or expressions in their own right achieve very low precision, and often return only noise. This is hard to detect without manual interference. Need to perform some sort of Named Entity Disambiguation, maybe something like the techniques outlined in Cucerzan [8] or Sarmento [9] (is elaboration needed?).

6.1 Evaluating against Netflix rating data

To find out how the Twitter-based predictions fare, we will use the average of the available Netflix ratings of the same titles as a benchmark.

With predictions p, and benchmark ratings r, we will compute the MSE (Mean Square Error) of N sample movies in the following way:

[1]The sample in question consisted of "Pulp Fiction", "The Shining", "Mission: Impossible", "The Matrix", "The Godfather", "Forrest Gump", and "A Clockwork Orange".

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(p_i - r_i)^2 \tag{6.1}$$

As a baseline metric, we'll compare against the MSE of the average of all the benchmark ratings:

$$\text{MSE}_{\text{baseline}} = \frac{1}{N}\sum_{i=1}^{N}(\bar{r} - r_i)^2 \tag{6.2}$$

Chapter 7

Conclusion

I thought that the data would contain too much noise to be usable in any other way than to annotate and – in the best of cases – adjust ratings of content where the social sentiment disagreed strongly with the proposed rating.

However, for certain kinds of content, the predictions generated by Twitter were more or less spot on the same as the ones from Netflix, as shown in chapter 6.

This leads me to believe that Twitter as a data source for recommendations can have a greater role than that of augmenting recommendations coming from elsewhere. (@TODO More specifically....)

Moving towards a conclusion:

1. Twitter's strongest suite lies in its abundance of novel content.
2. Some of the traditional CF systems' weakest points relate to recommending novel content.
3. Augmenting new content, with extremely sparse user ratings, might well be a good application.

Suggestions to further work:

- Applying NED to Twitter entities.
- Further improving sentiment analysis of informal texts.

Appendix A

Requirements

Appendix B

Design Documents

Appendix C

Evaluation Results

Bibliography

[1] Vivek Kumar Singh, Mousumi Mukherjee, and Ghanshyam Kumar Mehta. Combining a content filtering heuristic and sentiment analysis for movie recommendations. In K.R. Venugopal and L.M. Patnaik, editors, *Computer Networks and Intelligent Computing*, volume 157 of *Communications in Computer and Information Science*, pages 659–664. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-22785-1. doi: 10.1007/978-3-642-22786-8_83. URL http://dx.doi.org/10.1007/978-3-642-22786-8_83.

[2] Sang-Hyun Cho and Hang-Bong Kang. Statistical text analysis and sentiment classification in social media. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 1112–1117, 2012. doi: 10.1109/ICSMC.2012.6377880.

[3] Eytan Bakshy, Brian Karrer, and Lada A. Adamic. Social influence and the diffusion of user-created content. In *Proceedings of the 10th ACM conference on Electronic commerce*, EC '09, pages 325–334, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-458-4. doi: 10.1145/1566374.1566421. URL http://doi.acm.org/10.1145/1566374.1566421.

[4] Douglas Oard and Jinmook Kim. Implicit feedback for recommender systems. In *in Proceedings of the AAAI Workshop on Recommender Systems*, pages 81–83, 1998.

[5] Douglas W. Oard and Jinmook Kim. Modeling information content using observable behavior, 2001.

[6] Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, September 2003. ISSN 0163-5840. doi: 10.1145/959258.959260. URL http://doi.acm.org/10.1145/959258.959260.

[7] Vasilis Vryniotis. How to build your own twitter sentiment analysis tool, September 2013. URL http://blog.datumbox.com/how-to-build-your-own-twitter-sentiment-analysis-tool/.

[8] Silviu Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D/D07/D07-1074.

[9] Luís Sarmento, Alexander Kehlenbeck, Eugénio Oliveira, and Lyle Ungar. An approach to web-scale named-entity disambiguation. In *Proceedings of the 6th International Conference on Machine Learning and Data Mining in Pattern Recognition*, MLDM '09, pages 689–703, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-03069-7. doi: 10.1007/978-3-642-03070-3_52. URL http://dx.doi.org/10.1007/978-3-642-03070-3_52.