Mining Cross-Domain Rating Datasets from Structured Data on Twitter

Simon Dooms
iMinds-Ghent University
G. Crommenlaan 8, box 201
Ghent, Belgium
Simon.Dooms@UGent.be

Toon De Pessemier
iMinds-Ghent University
G. Crommenlaan 8, box 201
Ghent, Belgium
Toon.DePessemier@UGent.be

Luc Martens
iMinds-Ghent University
G. Crommenlaan 8, box 201
Ghent, Belgium
Luc1.Martens@UGent.be

ABSTRACT

While rating data is essential for all recommender systems research, there are only a few public rating datasets available, most of them years old and limited to the movie domain. With this work, we aim to end the lack of rating data by illustrating how vast amounts of ratings can be unambiguously collected from Twitter. We validate our approach by mining ratings from four major online websites focusing on movies, books, music and video clips. In a short mining period of 2 weeks, close to 3 million ratings were collected. Since some users turned up in more than one dataset, we believe this work to be amongst the first to provide a true cross-domain rating dataset.

Categories and Subject Descriptors

E.0 [**Data**]: General; H.1.2 [**Information Systems**]: Models and Principles—*User/Machine Systems, Human Information Processing*

Keywords

Dataset, mining, Twitter, Goodreads, YouTube, Pandora, IMDb

1. INTRODUCTION

Researchers are in constant need of data. They need data for analysis reasons, the validation of algorithms or to drive experiments. This is especially true in the field of recommender systems, where rating data serves as input data and is therefore a crucial component to calculate recommendations. Despite their importance, rating datasets are not abundantly available. Ratings are often considered private user data and therefore most online platforms do not make them publicly available. The lack of rating datasets has contributed to the growing gap between academia and industry. While academic researchers are working on algorithms and often lack user data for testing, industry has data and users, but needs the algorithms to improve their services.

To circumvent the lack of public ratings, most recommender systems research employs one of the few datasets that are available e.g., the MovieLens [1] or the Netflix dataset [2, 3]. Both datasets are slowly becoming outdated (i.e., originating from 2005 and 2007) and furthermore are restricted to the movie item domain. The most recent movie in the MovieLens 100K dataset, was released as far back as 1998. Nevertheless, this dataset is still widely used in recent literature (e.g., [4, 5]).

We aim to end the lack of up-to-date rating data by illustrating how large amounts of ratings can be unambiguously collected from current-day popular websites by mining the Twitter platform for specific pre-formated *tweets*. In previous work [6], we presented a new movie rating dataset called 'MovieTweetings', based on a similar approach. With this work, we generalize our method to other item domains as books, music and video clips, and provide the tools to allow researchers to collect and build cross-domain rating datasets for themselves.

2. STRUCTURED DATA ON TWITTER

With over 300 billion tweets in total[1], the data available through the Twitter platform is enormously abundant. Twitter users (at the time of writing) post an average of 500 million tweets every day, thereby exchanging opinions, ideas, links, jokes and many other types of information. In this work, we mine rating datasets by filtering and extracting relevant preference indicators contained in these tweets. We avoid dataset ambiguity and the need for complex natural language processing by focusing on structured tweets originating from the *social share* feature integrated on many popular online platforms. We illustrate our method focusing on four major online platforms with very divergent item domains: IMDb[2] (movies), Goodreads[3] (books), Pandora[4] (music), YouTube[5] (video clips).

2.1 Movies - IMDb

The first online platform focuses on movies. IMDb (Internet Movie Database) is one of the biggest and most popular websites concerning movies. It offers a wide variety of movie content information and lets users rate movies on a 10-star scale. Although users have the option to make their ratings public, few users actually do this, and so individual rating information is usually not accessible. Users of the IMDb mobile app for iOS, are however provided with the option to tweet their rating. The app proposes a pre-formatted tweet with the following structure.

```
I rated <Title> <Rating>/10 #IMDb
<Link to IMDb movie page>
```

[1] http://bit.ly/19YSAmo
[2] http://www.imdb.com
[3] http://www.goodreads.com
[4] http://www.pandora.com
[5] http://www.youtube.com

Querying the Twitter API[6] for the term 'I rated #IMDb' will result in tweets containing movie ratings originating from IMDb. From the tweet the following data fields can be extracted:

- User (Twitter user identifier)
- Movie title
- Rating (10-star scale)
- IMDb URL of the movie

The URL contains the unique IMDb identifier for the movie, which allows easy disambiguation of the movies and furthermore enables extraction of additional metadata from the movie's IMDb page. In previous work [6], we discussed how the Twitter API could be queried on a daily basis for such tweets, which resulted in a publicly available movie rating dataset called 'MovieTweetings'. The dataset offers the extracted ratings in a format similar to the MovieLens dataset and also integrates movie genre data.

2.2 Books - Goodreads

The second online platform we extract ratings from (through Twitter), is Goodreads. Goodreads is one of the largest websites for books discussion and discovery. Readers can review books and receive recommendations based on their personal taste. This website also offers to tweet about the review or rating a reader has provided. On Goodreads the following tweet template structure is used.

```
<Rating> of 5 stars to <Title> by
<Author> <Link to review on Goodreads>
```

To obtain the tweets originating from the Goodreads website, we query the Twitter API for 'of 5 stars to'. From the tweet the following information can be extracted:

- User (Twitter user identifier)
- Rating (5-star scale)
- Book title
- Book author
- Goodreads URL of the review

Since the URL refers to the review on the Goodreads website posted by the same user the tweet originates from, additional metadata fields (e.g. Goodreads user id, book id) can easily be extracted. Ratings provided by users on Goodreads are publicly available, so if the Goodreads user id is known, (although not implemented in this work) all of that user's ratings could additionally be extracted from the website to expand the ratings dataset even further.

2.3 Music - Pandora

Music is the third item domain this work focuses on. There are many online services for music, one of which is the online radio service Pandora. Using Pandora, users can easily stream music and receive song recommendations. The website offers a similiar social share feature as we found for IMDb and Goodreads. Users can tweet about the song they are currently listening to, using the following predefined tweet format.

[6] https://dev.twitter.com/docs/api/1.1

```
I'm listening to "<Title>" by <Artist> on
Pandora <Link to song on Pandora> #pandora
```

Very similar to the data available from tweets originating from the Goodreads website, the data fields available in this tweet format are:

- User (Twitter user identifier)
- Song title
- Song artist
- Pandora URL of the song

The difference here, is the lack of an explicit rating value. Pandora users do not rate the music, they either listen to it, or they do not. The tweets originating from the Pandora platform should therefore be considered implicit feedback. The query we use to get the Pandora tweets from the Twitter API is 'I am listening on #pandora'. The Pandora URL is available, so again additional metadata (e.g., music genre) can be extracted.

2.4 Video clips - YouTube

The fourth and final platform is YouTube. YouTube is currently the biggest provider of short video clips on the Internet and is widely famous and well-known all worldwide. While YouTube used to have a 5-star rating scale, in 2009 it was replaced with a thumbs up/down system because it more closely aligned with typically observed rating behavior[7]. When users watch videos on YouTube they can rate them by clicking a like or dislike button and tweet about it. The pre-formatted tweet in this situation shows the following structure.

```
I liked a @YouTube video [from @uploader]
<Link to YouTube video> <Title>
```

To restrict the Twitter API to results originating from these kinds of YouTube related tweets, we employ the query 'I liked a @YouTube video'. The data fields that can be extracted from the resulting tweets are:

- User (Twitter user identifier)
- The @handle of the video owner (optional)
- YouTube URL of the video

While there is no star-rating involved in this scenario, the feedback gathered is explicit feedback (i.e., the user explicitly expressed that she liked the video). The URL contains the unique YouTube identifier for the video which can be used to request additional content data (e.g., tags) from the YouTube API.

2.5 Other platforms

Although in this work we focus on the previously discussed platforms, other websites could be considered for rating extraction as well. In table 1 we also list some other online platforms (and suggested Twitter API search queries) we found to be integrating a social share feature for posting structured rating data to Twitter.

[7] http://youtube-global.blogspot.be/2009/09/five-stars-dominate-ratings.html

Table 1: Platforms available for rating extraction

Platform	Domain	Twitter API query
IMDb	movies	I rated #IMDb
Goodreads	books	of 5 stars to
Pandora	music	I am listening on #Pandora
YouTube	video clips	I liked a @YouTube video
Amazon	e-commerce	I just bought via @amazon
Spotify	music	#nowplaying on spotify
Last.fm	music	I'm listening to via @lastfm

3. MINING EXPERIMENT

To validate our method of extracting rating datasets from Twitter, we set up an experiment to automatically build four rating datasets, one for each of the online platforms discussed in the previous section.

3.1 Experimental setup

For each of the online platforms (i.e., IMDb, Goodreads, Pandora and YouTube.) we queried the Twitter API at fixed time intervals (between 5 and 30 minutes) to download all tweets containing the aforementioned preference indicators. The frequency of querying the Twitter API depended on the typical number of tweets associated with the specific website. YouTube tweets were much more numerous than tweets from IMDb, so we had to query the Twitter API more frequently (every 5 minutes) to capture all the relevant tweets, while respecting the Twitter API limitations.

The data fields as discussed in the previous section were extracted by means of a series of specific regular expressions and stored line by line in dataset files. We mined ratings over a period of 2 weeks (from December 19, 2013 to January 2, 2014) and processed all the captured tweets in 4 resulting datasets. The (Python) scripts used for the downloading and processing of the files, and the resulting datasets are available on the Github platform[8].

3.2 Results

Table 2 lists the basic characteristics for each of the 4 collected datasets. While each dataset was mined on Twitter for the exact same period of time, the number of extracted ratings is significantly different. The most ratings were collected from the YouTube platform, the fewest from Pandora (about 2000 times less). For all of the datasets the sparsity, calculated as equation 1, turned out to be very high. This is to be expected since the collected datasets are unfiltered. Often public datasets only integrate users with a minimum number of ratings while our datasets contain every user with at least one tweet containing a rating. The high sparsity values indicate high numbers of users and items with only little rating information to link them, which can be a major problem for collaborative filtering recommender systems known as the *sparsity problem* [7, 8].

$$sparsity = 1 - \frac{\#\ ratings}{(\#\ users) \times (\#\ items)} \quad (1)$$

On the other hand, filtering a dataset may cause the introduction of a systematic bias which may prevent the ability to generalize any obtained experimental results to real-life scenarios[9]. Furthermore, the natural datasets that come from our approach allow the simulation of a real-life recommender system, which often has to take into account many users and items with very few ratings or feedback.

[8] `https://github.com/sidooms/Twitter-ratings`

Table 2: Dataset statistics (2 week mining period)

	IMDb	Goodreads	Pandora	YouTube
#ratings	9,297	43,960	1,468	2,867,182
#users	3,412	19,680	1,039	420,373
#items	2,689	27,403	425	1,112,292
avg rat./day	664	3,140	105	204,799
sparsity	0.99899	0.99992	0.99668	0.99999

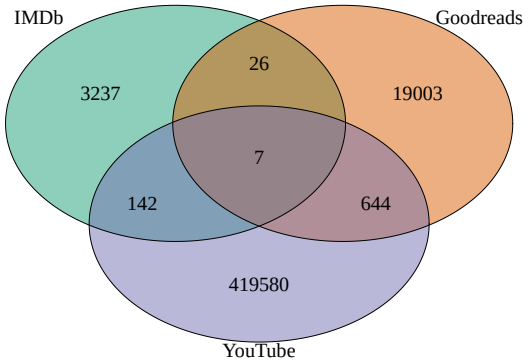

Figure 2: Venn diagrams indicating the numbers of unique users for the IMDb, Goodreads and YouTube datasets and their intersections.

Fig. 1 again illustrates the extreme difference in numbers of ratings collected from each platform during our mining period. The figure shows the daily number of ratings which varies day by day mostly depending on the day of the week (more activity in weekends).

An emerging research topic in the recommender systems area is cross-domain recommendation [10]. In cross-domain recommendation scenarios the sparsity problem is usually alleviated by integrating data from another domain e.g., recommending books based on previous movie ratings. One of the main challenges the domain faces is the lack of cross-domain rating datasets, which forces researchers to work with artificially generated datasets. With our approach we find ourselves in the unique position of linking rating data originating from the same (Twitter) user across multiple item domains (books, movies, music, etc.). We analyzed the intersections of the 4 collected datasets, more specifically the intersection of users (i.e., find Twitter user ids that have ratings in more than one dataset).

Fig. 2 shows the result of the intersection analysis for YouTube, Goodreads and IMDb (Pandora was omitted because of the low number of collected ratings). Many users turned out to actually rate across more than one domain. In total, 7 users were even found to rate across all three of the datasets. Considering the short period of data collection (2 weeks), these results look very promising for the creation of cross-domain rating datasets.

4. CONCLUSIONS

In this work we generalized our method of mining rating datasets from structured data on Twitter. We focused on 4 different online platforms each in another item domain:

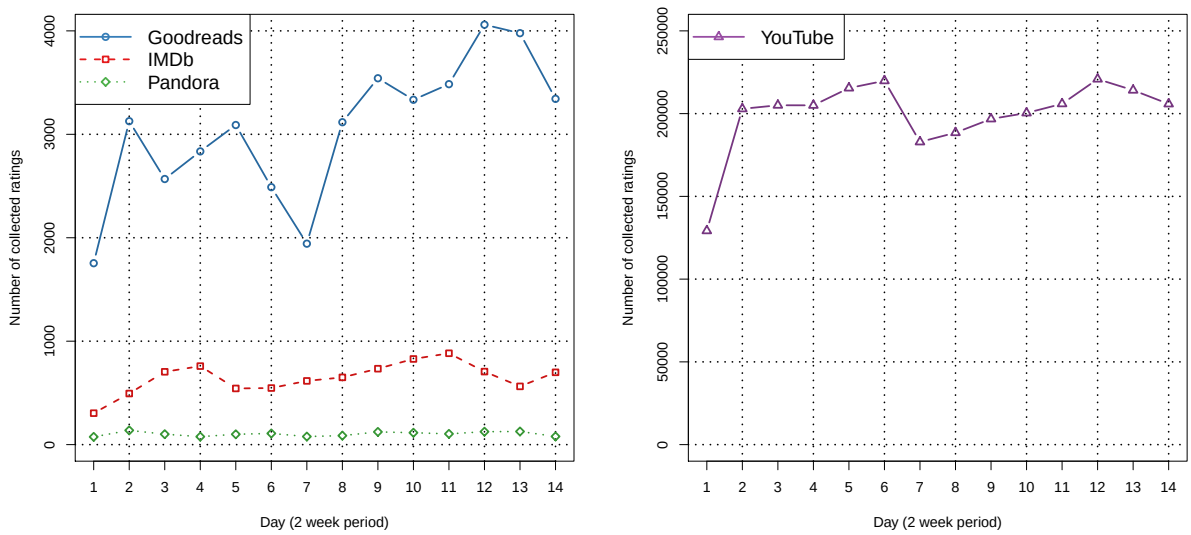

Figure 1: The daily number of collected ratings from Goodreads, IMDb, Pandora and YouTube. YouTube is displayed separately because of the large difference in Y-axis scale.

IMDb (movies), Goodreads (books), Pandora (music) and YouTube (video clips). We illustrated the application of our approach for each of these platforms and validated our method by mining ratings into 4 datasets over a period of 2 weeks. In total, almost 3 million ratings were collected, with some users even appearing in multiple datasets, paving the way towards cross-domain datasets. We hope this work ends the lack of rating data in the recommender systems area by assisting researchers in tapping into the vast amounts of information that can be easily extracted through social media.

5. ACKNOWLEDGMENTS

The described research activities were funded by a PhD grant to Simon Dooms of the Agency for Innovation by Science and Technology (IWT Vlaanderen).

6. REFERENCES

[1] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. 22nd annual int. ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM, 1999.

[2] Andreas Töscher, Michael Jahrer, and Robert M Bell. The bigchaos solution to the netflix grand prize. *Netflix prize documentation*, 2009.

[3] Martin Piotte and Martin Chabbert. The pragmatic theory solution to the netflix grand prize. *Netflix prize documentation*, 2009.

[4] Rui Ping Song, Bo Wang, Guo Ming Huang, Qi Dong Liu, Rong Jing Hu, and Rui Sheng Zhang. A hybrid recommender algorithm based on an improved similarity method. *Applied Mechanics and Materials*, 475:978–982, 2014.

[5] Yong Liu, Fan Jia, and Wei Cao. A degree-based method to solve cold-start problem in network-based recommendation. In *Advanced Technologies, Embedded and Multimedia for Human-centric Computing*, pages 897–904. Springer, 2014.

[6] Simon Dooms, Toon De Pessemier, and Luc Martens. Movietweetings: a movie rating dataset collected from twitter. In *Workshop on Crowdsourcing and Human Computation for Recommender Systems, CrowdRec at RecSys 2013*, 2013.

[7] Bin Li, Qiang Yang, and Xiangyang Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *IJCAI*, volume 9, pages 2052–2057, 2009.

[8] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. Transfer learning in collaborative filtering for sparsity reduction. In *AAAI*, volume 10, pages 230–235, 2010.

[9] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011.

[10] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. Cross-domain recommender systems: A survey of the state of the art. In *Proc. 2nd Spanish conf. on Information Retrieval. CERI*, 2012.