A Large Multilingual and Multi-domain Dataset for Recommender Systems

Giorgia Di Tommaso, Stefano Faralli*, Paola Velardi
Department of Computer Science, *Unitelma
Sapienza, University of Rome, Italy
{ditommaso,velardi}@di.uniroma1.it, stefano.faralli@unitelmasapienza.it

Abstract

This paper presents a multi-domain interests dataset to train and test Recommender Systems, and the methodology to create the dataset from Twitter messages in English and Italian. The English dataset includes an average of 90 preferences per user on music, books, movies, celebrities, sport, politics and much more, for about half million users. Preferences are either extracted from messages of users who use Spotify, Goodreads and other similar content sharing platforms, or induced from their "topical" friends, i.e., followees representing an interest rather than a social relation between peers. In addition, preferred items are matched with Wikipedia articles describing them. This unique feature of our dataset provides a mean to derive a semantic categorization of the preferred items, exploiting available semantic resources linked to Wikipedia such as the Wikipedia Category Graph, DBpedia, BabelNet and others.

Keywords: social mining, recommender systems, Twitter, users' interest dataset

1. Introduction

Recommender systems are widely integrated in online services to provide suggestions and personalize the on-line store for each customer. Recommenders identify preferred items for individual users based on their past behaviors or on other similar users. Popular examples are Amazon (Linden et al., 2003) and Youtube (Davidson et al., 2010). Other sites that incorporate recommendation engines include Facebook, Netflix, Goodreads, Pandora and many others.

Despite the vast amount of proposed algorithms, the evaluation of recommender systems is very difficult (Fouss and Saerens, 2008). In particular, if the system is not operational and no real users are available, the quality of recommendations must be evaluated on existing datasets, whose number is limited and what is more, they are focused on specific domains (i.e, music, movies, etc.). Since different algorithms may be better or worse depending on the specific purpose of the recommender, the availability of multi-domain datasets could be greatly beneficial. Unfortunately, real-life cross-domain datasets are quite scarce, mostly gathered by "big players" such as Amazon and eBay, and they not available to the research community[1].

In this paper we present a methodology for extracting from Twitter a large dataset of user preferences in multiple domains and in two languages, Italian and English. To reliably extract preferences from users' messages, we exploit popular services such as Spotify, Goodreads and others. Furthermore, we infer many other preferences from users' friendship lists, identifying those followees representing an interest rather than a peer friendship relation. In this way we learn, for any user, several interests concerning books, movies, music, actors, politics, sport, etc. The other unique feature of our dataset, in addition to multiple languages and domains, is that preferred items are matched with corresponding Wikipedia pages, thus providing the possibility to generalize users' interests exploiting available semantic resources linked to Wikipedia, such as the Wikipedia Category Graph, Babelnet, DBpedia, and others.

The paper is organized as follows: Section 2. summarizes previous research on creating datasets for recommender systems, Section 3. describes our methodology to collect data, and Section 4. presents and evaluates our results. Finally, in Section 5. we draw conclusions providing some directions for future research.

2. Related work

Most recommender systems (hereafter RS) are based on one of three basic approaches (Felfernig et al., 2014): collaborative filtering (Schafer et al., 2007) generates recommendations collecting preferences of many users, content-based filtering (Pazzani and Billsus, 2007) suggests items similar to those already chosen by the users, and knowledge-based recommendation (Trewin, 2000) identifies a semantic correlation between user's preferences and existing items. Hybrid approaches are also widely adopted (Burke, 2002). All approaches share the need of sufficiently large datasets to learn preferences and to evaluate the system, a problem that is one of the main obstacles to a wider diffusion of RS (Gunawardana and Shani, 2009) since only a small number of researchers can access real users data, due to privacy issues.

To overcome the lack of datasets, challenges as RecSys have been lunched[2], and dedicated web sites have been created (e.g., SNAP[3] or Kaggle[4]), where researchers can upload their datasets and make them available to the community. However it is still difficult to find appropriate data for novel types of recommenders, as the majority is focused on a single topic, like music (Dror et al., 2012), (Shepitsen et al., 2008),) food ((Kamishima and Akaho, 2010), (Sawant and Pai, 2013)), travel ((Wang et al., 2010), (Mavalankar et al., 2017)) and more (Çano and Morisio, 2015). Fur-

[1] https://recsys.acm.org/wp-content/uploads/2014/10/recsys2014-tutorial-cross_domain.pdf

[2] https://recsys.acm.org/

[3] http://snap.stanford.edu/data

[4] https://www.kaggle.com/datasets/?sortBy=hottest&group=all

thermore, while a small number of large datasets are available, such as Movielens (Harper and Konstan, 2016), Million song dataset (McFee et al., 2012) and Netflix Prize Dataset (Bennett et al., 2007), many others are quite small and based on very focused experiments.

Concerning the source of data for extracting preferences, social networks are often used, since their content is available with more or less severe restrictions. For example, (Chaabane et al., 2012) use Facebook, perhaps the most appropriate platform for this type of study, as it provides incentive mechanisms for sharing interests and content. However, a disadvantage is the difficulty in extensively accessing profiles due to user privacy issues. In (Chaabane et al., 2012), 104,000 public and 2000 private profiles (obtained by volunteers) have been extracted. Another study (Yan et al., 2014) presents an overview of users' interests derived from multiple platforms to which the same user is registered (e.g., Twitter, Youtube, etc.). To find the same account on multiple social networks, Google+ is used, where users are encouraged to share and link the addresses of their accounts. Overall 143,259 accounts were collected, of which, 11,850 provided multiple accounts. Many other studies use Twitter (eg. (Gesualdo et al., 2013), (Adamopoulos and Tuzhilin, 2014)) as a platform for extracting users' information, although existing restrictions limit the amount of freely accessible traffic to 1%.

Data extraction from Twitter messages is expensive since it requires natural language processing techniques to analyze the text. To overcome this difficulty, a number of studies exploited platforms (e.g., Youtube, Spotify) that integrate among their services the ability to post the user's personal content on the most popular social network sites, such as movies that users are watching. Sharing this information is done in a simple and predefined way. Depending on the social network chosen, the content, for example a Youtube video, will be shared with a pre-formatted message formed by the video name, a link, a self-generated text and, if provided, a numerical rating (eg. "How It's Made: Bread" `https://youtu.be/3UjUWfwWAC4` via YouTube). The message can also be enriched and personalized by the user. In (Pichl et al., 2015) this types of messages are extracted from Twitter, to detect music interests. The dataset is based on 100,000,000 tweets with the #nowplaying main tag. Tweets are extracted via Twitter APIs over 3-years and next, MusicBrainz and Spotify are used to add more details. Other studies extract data about music (Schinas et al., 2013) or sport (Nichols et al., 2012) events. However, all the datasets generated in this way concern only one domain of interest.

To the best of our knowledge, the only really multi-domain dataset[5] is presented in (Dooms et al., 2014), where pre-structured tweets about three domains - movies, books and video-clips - are extracted respectively from IMDb (Internet Movies Database), Youtube and Goodreads. With respect to this work, we collect a much wider number of interests, since in addition to pre-formatted messages based on a number of available services, we reliably extract many additional types of interests exploiting users' followees lists. Furthermore, as shown in Section 4., we collected many interest *types* for each user, while the dataset released in (Dooms et al., 2014) includes only 7 users with at least 3 types of interests.

[5]Another is the ConcertTweets `https://github.com/padamop/ConcertTweets`, however it is focused on music events.

3. Workflow and data sources

This section summarizes the data sources and workflow to create our multi-domain dataset. We extract preferences from a user's messages and from his/her friendship list, identifying those followees which represent an interest rather than a peer friendship relationship. The process is in three steps:

1. *Extracting interests from users' textual communications*. The first step is to extract preferences from Twitter messages. Using textual features extracted from users' communications, profiles or lists seems a natural way for modeling their interests. However, this information source has several drawbacks when applied to large data streams, such as the set of Twitter users. First, it is computationally very demanding to process millions of daily tweets in real time; secondly, the extraction process is error prone, given the highly ungrammatical nature of micro-blogs. To reliably extract preferences from users' messages, in line with other works surveyed in Section 2., we use a number of available services, described hereafter, that allow to share activities and preferences in different domains - movies, books etc. - using pre-formatted expressions (e.g, for Spotify: #NowPlaying) followed by the url of a web site, from which we can extract information without errors. The drawback is that a relatively small number of users access these services and in addition, preferences are extracted only in few domains.

2. *Extracting interests from users' friendship lists*. In (Barbieri et al., 2014) the authors argue that users' interests can also be implicitly represented by the authoritative (*topical*) friends they are linked to. This information is available in users' profiles and does not require additional textual processing. Furthermore, interests inferred from topical fiends are less volatile since, as shown in (Myers and Leskovec, 2014), "common" users tend to be rather stable in their relationships. Topical friends are therefore both relatively stable and readily accessible indicators of a user's interest. Another advantage is that average Twitter users have hundreds of followees, many of which, rather than genuine friends, are indicators of a variety of interests in different domains, such as entertainment, sport, art and culture, politics, etc.

3. *Mapping interests onto Wikipedia pages*. The final step is to associate each interest, either extracted from messages or inferred from friendship relations, with a corresponding Wikipedia page, e.g., @nytimes $\Rightarrow$ WIKI:EN:The_New_York_Times (in this example, @nytimes is a Twitter account extracted from a user's friendship list). Although not all interests can be mapped on Wikipedia, our experiments show that this

is possible in a large number of cases, since Wikipedia articles are created almost in real-time in correspondence with virtually any popular entity, either book, or song, actor, event, etc.

We applied this workflow to two Twitter streams in two languages, English and Italian, as detailed in what follows.

3.1. Extracting preferences from messages

Everyday a huge number of people uses on-line platforms (eg. Yelp, Foursquare, Spotify, etc.) that allow to share activities and preferences on different domains on a social network in a standard way. Among the most popular services accessed by Twitter users, we selected those providing pre-formatted messages, as detailed hereafter.

- **Spotify:** Spotify is a music service offering on-demand streaming of music, both desktop and mobile. Users can also create playlists, share and edit them in collaboration with other users. In addition to accessing the Spotify web site, users can retrieve additional information such as the record label, song releases, date of release etc.. Since 2014, Spotify is widely used in America, Europe and Australia. Spotify is among the services allowing to generate self-generated content shares in Twitter. An example of these tweets is: "#NowPlaying The Sound Of Silence by Disturbed `https://t.co/d8Sib5EDVf`". The standard form of these tweets is:

  `#NowPlaying <title> by <artist > <URL>`

  By filtering the tweets stream and using Twitter APIs for hashtag detection, we generated a stream of all the users who listened music using Spotify.

- **Goodreads and aNobii:** Similarly to Spotify, a number of platforms allows to share opinions and reviews on books. In these platforms, users can share both titles and ratings. Similarly to Spotify, generated tweets have a predefined structure and point to an URL. In the book domain, we use Goodreads (10 million users and 300 million books in the database) and for Italian, the more popular aNobii service.

- **IMDb and TVShowTime:** In the domain of movies, currently there are no dominant services. Popular platforms in this area are Flixter, themoviedb.org and iCheckMovies. However, many of these platforms use the IMDb database, owned by Amazon, which handles information about movies, actors, directors, TV shows, and video games. We also use the TvShowTime service for Italian tweets.

First, we collect in a Twitter stream all messages including a hashtag related to one of the above mentioned services (#NowPlaying, #IMDb ..). Next, we extract from tweets the music, movie and book preferences for a set of users U who accessed these services. Unlike (Dooms et al., 2014), we avoid parsing tweets using specific regular expressions, since users are free to insert additional text in the pre-formatted message. Rather, we exploit an element that all these pre-formatted tweets have: the URL, as in (Pichl et al., 2014). Every URL points to the website containing all the information, such as, e.g., title, author and publisher for books. Since the URL in the tweets is a short URL, we first extend the original URL so that all URLs belonging to a given platform can be identified (for example, all Goodreads URLs contain the "goodreads.com" string). Next, we access the web site and scrape its content. The reason for extracting the information from the URL (which is computationally more demanding) rather than from the tweet itself is twofold:

1. Tweets can be ambiguous or malformed, and furthermore, users can insert additional text in the pre-formatted message, e.g, "*#NowPlaying Marty*. This guy is amazing.♫`http://t.co/jwxvLiNenW`". Scraping the html page at the URL address ensures that we extract data *without errors*, even for complex items such as book and movie titles;

2. The URL includes additional information (e.g., not only the title of a song, but also the singer and the record label), which provide us a context to reliably match the extracted entity (song, book, movie) with a Wikipedia article, as detailed in Section 3.3..

3.2. Extracting preferences from users' "topical" friends

We denote as *topical friends* those Twitter accounts in a user's followees list representing popular entities (celebrities, products, locations, events . . .). For example, if a user follows @David_Lynch, this means that he/she likes his movies, rather than being a genuine friend of the director.
There are several clues to identify topical friends in a friendship list: first, topical relationships are mostly *not reciprocated*, second, popular users have a high in-degree. However, these two clues alone do not allow to distinguish e.g., bloggers or very social users from truly popular entities.
To lean a model of topical friends we first collected a network of Verified Twitter Accounts. Verified accounts [6] are authentic accounts of public interest. We started from a set of seed verified contemporary accounts in 2016, and we then crawled the network following only verified friends, until no more verified accounts could be found. This left us with a network of 107,018 accounts of verified contemporary users (V), representing a "model" of authoritative users' profiles. Next, from the set U of users in our datasets (separately for the English and Italian streams), we collected the set F of Twitter accounts such that, for any $f \in F$ there is at least one $u \in U$ such that u follows f.
In order to identify candidate topical friends $F_t \subset F$, we learned a model of popularity, using the set V and a random balanced set of $\neg V$ users. For each account, we extracted three structural features (in degree, out degree and their ratio) and one binary textual feature (presence in the user's account profile of role words such as *singer, artist, musicians, writer*..). Then, we used 80% of these accounts to train a SVM classifier with Laplacian kernel and the remaining 20% for testing with cross-validation, obtaining a

[6] `https://developer.twitter.com/en/docs/api-reference-index`

total accuracy of 0.88 (true positive rate 0.95 and true negative rate 0.82). Finally, the classifier was used to select a subset $F_t \subseteq F$ of *authoritative* users representing "candidate" topical friends. The last filtering step to identify "true" topical friends in F_t, i.e., genuine users' *interests*, consists in determining which members of the set F_t have a matching Wikipedia page. The intuition is that, if one such match exists, the entity to which the Twitter account belongs is indeed "topical".

3.3. Mapping to Wikipages

Mapping interests extracted from users' messages to Wikipedia pages is a very reliable process, given the additional contextual information extracted from the URL (see Section 3.1.). For a complete example, see Section 3.4..
On the contrary, matching interests extracted from a user's friendship list with corresponding Wikipedia pages is far more complex, because of synonymy, polysemy and ambiguity, as argued in (Faralli et al., 2017). Furthermore, the information included in a user's Twitter profile is very sketchy and in some case misleading, therefore it may not provide sufficient context to detect a similarity with the correspondent Wikipedia article. For example, Bill Gate's description field[7] in his Twitter profile is: "*Sharing things I'm learning through my foundation work and other interests...*" which has little in common with his Wikipedia page: "*William Henry Gates III (born October 28, 1955) is an American business magnate, investor, author, philanthropist, humanitarian and co-founder of the Microsoft Corporation along with Paul Allen.*"
To find the Wikipedia page, if any, associated to a topical friend we used the methodology that was first presented in (Faralli et al., 2015) and improved in (Faralli et al., 2017), summarized in what follows:

1. *Selection of candidate senses*: For any f in F_t, find a (possibly empty) list of *candidate wikipages*, using BabelNet synonym sets (in BabelNet, each "BabelSynset" points to a unique Wikipedia entry (Navigli and Ponzetto, 2012));

2. *BoW Disambiguation*: Compute the bag-of-words (BoW) similarity between the user description in f's Twitter account and each candidate wikipage. The BoW representation for each wikipage is obtained from its associated BabelNet relations (Delli Bovi et al., 2015);

3. *Structural Similarity*: If no wikipages can be found with a sufficient level of similarity (as for the previous example of Bill Gates), select from f's friendship list those friends already mapped to a wikipage (if any), and compute the similarity between those wikipages and candidate wikipages.

3.4. Anecdotic examples and evaluation

We provide hereafter examples of the process outlined in previous Sections. For the sake of space, we consider only examples of interests extracted from users' messages.

[7] as retrieved on January 2018

1. **detection of "interesting" tweets**
   We collect all tweets containing the selected hashtags and discard those which do not include an url.
   accepted #NowPlaying High by James Blunt https://t.co/7EiepE2Bvz
   discarded #NowPlaying CaSh Out - Cashin' Out

2. **extraction of the url**
   Next, we retrieve the original url from short url. If the url does not contain the platform domain (eg. spoti.fi), we discard it.
   accepted: `https://t.co/oShYDc6DeL` → `http://spoti.fi/2cTPn0U`
   discarded: `https://tunein.com/radio/Pratt-Radio-s50434/`

   Then, we extract information about an item (movie, book or music) from the platform site through APIs, (when available) or web-scraping. For each platform we obtain the following data:

   (a) **Music**: <Title, Author (eg. singer, band)>

   (b) **Books**: <Title, Author>

   (c) **Movie**: <Title, Year of production, Type (eg. movie, tv series)>

3. **mapping to Wikipedia**
   Wikipedia mapping is obtained by a cascade of weighted boolean query on a Lucene Index.The index is based on a tdf-idf with vector space model.

   (a) Searching the Wikipage of an item
   $< TITLE \in WikiTitle >^{w_1}$
   $\wedge < AUTHOR \in WikiGloss >^{w_2}$
   $\wedge\big(\big(< WORDS \in WikiTitle >^{w_3}$
   $\vee < AUTHOR \in WikiTitle >^{w_4}$
   $\vee < WORDS \in WikiText >^{w_5} \big)$
   $\vee\neg\big(< WORDS \in WikiTitle >^{w_3}$
   $\vee < AUTHOR \in WikiTitle >^{w_4}$
   $\vee < WORDS \in WikiText >^{w_5} \big)\big)$
   Where
   w_i is a weigth assigned to a query
   $< WORDS >$ for music = {"song"}
   $< WORDS >$ for books = {"books", "novel", "saga"
   $< WORDS >$ for movie = {"film", "series", "TV series", "episode"}
   When the page doesn't exist or is not available we search the page of the item's author.

   (b) Searching the wikipage of the item's author
   $< AUTHOR \in WikiTitle >^{w_1}$
   $\wedge\big(\big(< WORDS \in WikiTitle >^{w_2}$
   $\vee < TITLE \in WikiText >^{w_3}$
   $\vee < WORDS \in WikiText >^{w_4} \big)$
   $\vee\neg\big(< WORDS \in WikiTitle >^{w_2}$
   $\vee < TITLE \in WikiText >^{w_3}$
   $\vee < WORDS \in WikiText >^{w_4} \big)\big)$
   Where

USER ID:787930***		
Source	**Interest**	**Wikipage**
IMDb	Eyes Wide Open - 2009 - movie	WIKI:EN:Eyes_Wide_Open_(2009_film)
	Okja - 2017 - movie	WIKI:EN:Okja
Goodreads	The Beautifull Cassandra - Jane Austen	WIKI:EN:Jane_Austen
	The Beach - Alex Garland	WIKI:EN:The_Beach_(novel)
Spotify	I Don't Know What I Can Save You From - Kings of Convenience	WIKI:EN:Kings_of_Convenience!
	Nothing Matters When We're Dancing - **The Magnetic Fields**	WIKI:EN:The_Magnetic _Fields
Topical friends	**@IMDb**	WIKI:EN:IMDb
	@UNICEF_uk	WIKI:EN:UNICEF_UK
	@TheMagFields	WIKI:EN:The_Magnetic _Fields
	@BarackObama	WIKI:EN:Barack_Obama
	@Spotify	WIKI:EN:Spotify

Table 1: Excerpt of a Twitter user's interests

w_i is a weigth assigned to a query
$< WORDS >$ for music = {"singer", "band", "artist", "songwriter", "composer", "musician", "record producer"}
$< WORDS >$ for books = {"writer", "novelist", "cartoonist", "journalist", "orator", "poet", "Japanese manga author"}

Examples of positive results when searching an item's title:
Tweet: I rated Arrow: Disbanded (S5.E18) 9/10 #IMDb `https://t.co/Oo4qu6tHl7`
Original url: `http://www.imdb.com/title/tt5607516/`
WikiPage: WIKI:EN:Arrow_(TV_series)
Note that WikiTitle contains the term "TV_series" in WORDS.

Examples with positive results when searching an item's author:
Tweet: 4 of 5 stars to Silken Prey by John Sandford `https://t.co/AyF5Iuyc9s`
Original url: `https://www.goodreads.com/review/show/2105147499`
WikiPage: WIKI:EN:John_Sandford_(novelist)
Note that WikiTitle contains the term "novelist" in WORDS.

Examples with incorrect results:
The system returns no results or incorrect results in 3 cases:

1. the title page of the item or item's author page doesn't exist:

   Tweet: #NowPlaying Cherry Garcia by Dingus `https://t.co/t7g4EQ3ucp`
   Original url: `https://open.spotify.com/track/2vfZpWZGUtfM2VYVomh7MZ`
   WikiPage: WIKI:EN:Eric_Dingus
   The wikipage of the correct singer (which is not Eric Dignus) doesn't exist.

2. the extracted data is wrong because information extraction fails for various reasons (eg. missing or poorly structured information in the platform):

   Tweet: 4 of 5 stars to Love, Rosie by Cecelia Ahern `https://t.co/EtS1RoCarK`
   Original url: `https://www.goodreads.com/review/show/1965312746`
   WikiPage: WIKI:EN:Love,_Rosie_(film)
   In this case Wikipedia shows the title page with the initial name of the book "WIKI:EN:Where_Rainbows_End", but the original name of the book was modified in the reprint.

3. the searched Wikipedia page does not contain enough context to match the query.

Overall, the methodology to extract and map preferences from messages proved to be very reliable. We evaluated the precision (with adjudication) on a randomly selected balanced sample of 1200 songs, books, and movies in English, obtaining a precision of 96%. For the Italian dataset, we evaluated 750 songs, books, and movies, obtaining a precision of 98%.

As far as the topical interests mapping performance is concerned, in (Faralli et al., 2017) the authors mention that inducing interests from topical friends and subsequent mapping to Wikipedia has an accuracy of 84%. Since our aim in this work is to generate a highly accurate dataset, we considered only the subset F'_t in F_t with indegree (with respect to our population U) higher than 40. In fact, we noted that less popular topical friends may still include bloggers or Twitter users for which, despite some popularity, a Wikipage does not exists. In these cases, our methodology may suggest false positives. When applying the indegree filter, the precision -manually evaluated with adjudication on 1250 accounts randomly chosen in this restricted population F'_t- is as high as 90%.
Finally, we note that we are not concerned here with measuring the recall, since our aim is to release a dataset with high precision and high coverage, in terms of number of interests per user, over the considered populations. To this end, the indegree threshold 40 was selected upon repeated experiments to obtain the best trade-off between the distribution of interests in the population U and precision of the mapping, as shown in Section 4..

message-based interests ($\|U\|$=444,744 English speaking users)	**Music**	**Books**	**Movie**	**Total**
platform	*Spotify*	*Goodreads*	*IMDb*	*All*
#crawled tweets (tweets with selected hashtags)	19,941,046	693,975	97,772	20,732,793
#cleaned tweets (tweets for which an URL was extracted)	2,519,166	139,882	88,355	2,747,403
# of unique interests with a mapping to a Wikipage	253,311	20,710	8,282	282,303
average #interests per user	6	8	6	6
average #users per interest	7	3	7	6
precision of Wikipedia mapping (on 3 samples of 400 items each)	94%	96%	97%	**96%**

Table 2: 6-months (April-September 2017) statistics on **message-based** interests extracted from English-speaking users

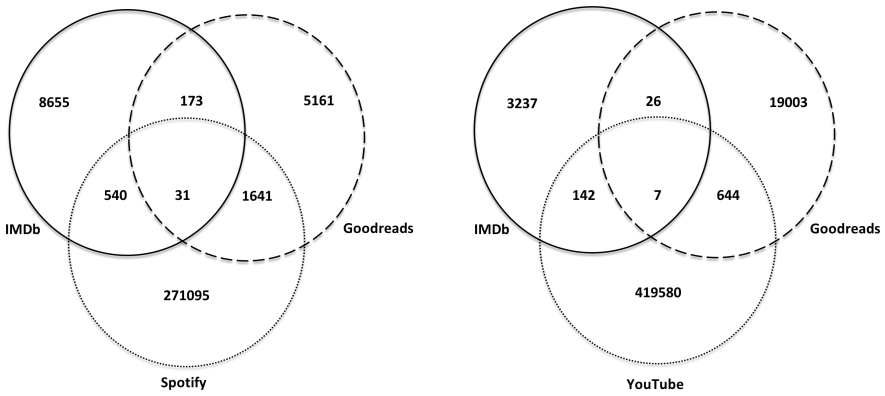

Figure 1: Venn Diagram of message-based interest *types* for our English dataset (left) and the dataset in Dooms et al.(2014)

message-based interests ($\|U\|$= 25,135 Italian speaking users)	**Music**	**Books**	**Movie**	**Total**
platform	*Spotify*	*ANobii*	*IMDb & TVShowTime*	*All*
#crawled tweets (tweets with selected hashtags)	273,256	12,198	2,229	287,683
#cleaned tweets (tweets fro which an URL was extracted)	70,330	12,193	2,119	84,642
# of unique interests with a mapping to a Wikipage	9,926	4,690	279	14,895
average #interests per user	3	9	7	6
average #users per interest	5	2	5	4
precision of Wikipedia mapping (on 3 samples of 250 items each)	96%	98%	100%	**98%**

Table 3: 6-months (April-September 2017) statistics on **message-based** interests extracted from Italian-speaking users

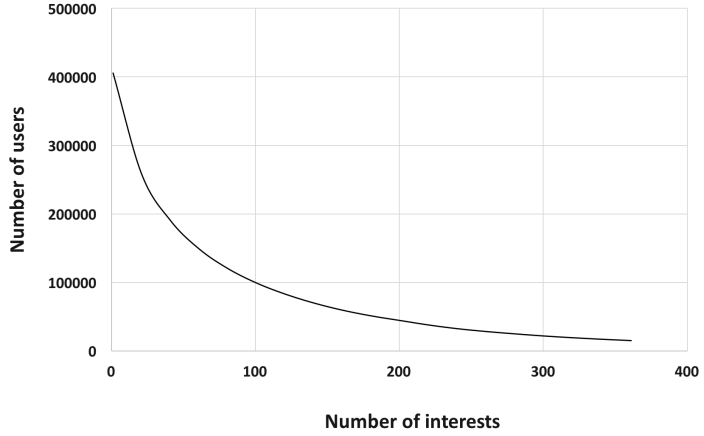

Figure 2: Distribution of interests induced from users's topical friends (English dataset)

Interests induced from topical friends ($\|U\|$=444,744 English speaking users)	
# of topical friends F'_t with indegree $\geq$ **40** in U	**409,743**
# of unique interests with a mapping to a Wikipage	58,789
average #interests per user	82
precision of Wikipedia mapping (tested on a sample of 1,250 items in F'_t)	90%

Table 4: 6-months (April-September 2017) statistics on interests induced from **topical friends** of English-speaking users

4. Description of the Dataset

The outlined process has been applied to two streams of Twitter data, in English and Italian, extracted during 6 months (April-September 2017) using Twitter APIs. We collected the maximum allowed Twitter traffic of English users mentioning service-related hashtags (e.g., #NowPlaying for Spotify), and the full stream of messages in Italian, since they do not exceed the maximum. As a final result, we obtained for a large number of users a variety of interests along with their corresponding Wikipedia pages. An excerpt of a Twitter user's interests is shown in Figure 1. In the example, we selected two interests from each of the four sources from which they have been induced:IMDb (movies), Goodreads (books), Spotify (music) and four interests from the user's topical friends. Although a detailed analytics of interest categories is deferred to further studies, the example shows the common trend that a user's interests, either extracted from his/her messages or from topical friends, are strongly related, and in same case identical. For example, the user in Table 1 frequently accesses the IMDb and Spotify services, and he/she is also a follower of the IMDb and Spotify Twitter account. Furthermore, his/her interest in the band The Magnetic Field emerges from both source types.

Overall, we followed 444,744 English-speaking and 25,135 italian-speaking users (the set U) who accessed at least one of the services mentioned in Section 3.1.. The general statistics of interests extracted from users' messages are shown in Tables 2 and 3.

In the English dataset we crawled more than 20M tweets from these users, of which, about 2.7M could be associated to the URL of a corresponding book, movie or music. On average, we collected 6 interests per user. What is more, several users have interests in at least two of the three domains. Figure 1 compares the Venn diagram of interest types in our dataset (left) with that reported in (Dooms et al., 2014) (right), to demonstrate the superior coverage of our dataset, even when considering only preferences extracted from users' messages.

The number and variety of extracted preferences is however mostly determined by the interests induced from users' topical friends, as shown in Table 4 (English dataset). Although, to ensure a high precision of the Wikipedia mapping step, we mapped only topical friends in $F'_t \subset F_t$ with a high in-degree from users in U (see Section 3.4.), the average number of interests induced for each user is as high as 82, and the distribution is shown in Figure 2. The Figure shows, e.g., that there are 100,000 users in U with ≥ 100 interests induced from their topical friends.

When merging the two sources of information, our dataset includes an average of 90 interests per user for about 450k users, in a large variety of domains. To the best of our knowledge, this is the largest multi-domain interest dataset reported in literature, and furthermore, we provide the unique feature of a reliable mapping to Wikipedia.

We release under creative commons license the dataset of English-speaking users along with their preferences. (`http://lmm.tweets.di.uniroma1.it/lmm/`). The dataset is in five files. Details are provided in the *readme* file. Further note, as we explained in Section 3., that the process of extracting interests from messages is almost free of errors (96% precision), while inducing interests from topical friends and subsequent mapping to Wikipedia has an estimated 10% error rate. However, as mentioned in Section 1., semantic techniques can be applied to reliably identify the main *categories* of interest for each user, an enhancement that we leave to future work.

5. Concluding Remarks

In this paper we presented a new dataset that captures, from Twitter messages and friendship lists, users' interests in multiple domains. We described the methodology to create the interests dataset and released a dataset extracted from an English Twitter stream collected during April-September 2017. The interests dataset can be extended to more languages and domains through the same methodology. Thus, the dataset and its extensions can be used in a number of applications in the domain of Recommender Systems, but not only. Although the dataset possibly includes extraction errors (which is a common problem in large, automatically extracted resources), the unique feature of mapping interests to Wikipedia articles and the large number of interests associated to each user, offer the possibility to identify for each user the "dominant" interest *categories*, on which Recommender Systems could rely when suggesting new items.

6. Acknowledgements

We greatly thank Dr.Giovanni Stilo, who developed the framework for extracting real-time data from Twitter streams, under the restrictions specified by Twitter. This work has been in part supported by the IBM Faculty Award #2305895190 (2017).

7. Bibliography

Adamopoulos, P. and Tuzhilin, A. (2014). Estimating the value of multi-dimensional data sets in context-based recommender systems. In *RecSys Posters*.

Barbieri, N., Bonchi, F., and Manco, G. (2014). Who to follow and why: link prediction with explanations. In *20th ACM SIGKDD*, pages 1266–1275. ACM.

Bennett, J., Lanning, S., et al. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370.

Çano, E. and Morisio, M. (2015). Characterization of public datasets for recommender systems. In *Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2015 IEEE 1st International Forum on*, pages 249–257. IEEE.

Chaabane, A., Acs, G., Kaafar, M. A., et al. (2012). You are what you like! information leakage through users? interests. In *19th (NDSS)*.

Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., Gupta, S., He, Y., Lambert, M., Livingston, B., et al. (2010). The youtube video recommendation system. In *4th ACM conference on Recommender systems*, pages 293–296. ACM.

Delli Bovi, L., Telesca, L., and Navigli, R. (2015). Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *TACL*, 3:529–543.

Dooms, S., De Pessemier, T., and Martens, L. (2014). Mining cross-domain rating datasets from structured data on twitter. In *23rd International Conference on World Wide Web*, pages 621–624. ACM.

Dror, G., Koenigstein, N., Koren, Y., and Weimer, M. (2012). The yahoo! music dataset and kdd-cup?11. In *KDD Cup 2011*, pages 3–18.

Faralli, S., Stilo, G., and Velardi, P. (2015). Large scale homophily analysis in twitter using a twixonomy. In *24th, IJCAI, Buenos Aires, July 25-31, 2015*, pages 2334–2340.

Faralli, S., Stilo, G., and Velardi, P. (2017). Automatic acquisition of a taxonomy of microblogs users' interests. *Web Semantics: Science, Services and Agents on the World Wide Web*, 45:23–40.

Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., Reiterer, S., and Stettinger, M. (2014). Basic approaches in recommendation systems. In *Recommendation Systems in Software Engineering*, pages 15–37. Springer.

Fouss, F. and Saerens, M. (2008). Evaluating performance of recommender systems: An experimental comparison. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 735–738. IEEE Computer Society.

Gesualdo, F., Stilo, G., Gonfiantini, M. V., Pandolfi, E., Velardi, P., Tozzi, A. E., et al. (2013). Influenza-like illness surveillance on twitter through automated learning of naïve language. *PLoS One*, 8(12):e82489.

Gunawardana, A. and Shani, G. (2009). A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, 10(Dec):2935–2962.

Harper, F. M. and Konstan, J. A. (2016). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19.

Kamishima, T. and Akaho, S. (2010). Nantonac collaborative filtering: A model-based approach. In *4th ACM conference on Recommender systems*, pages 273–276. ACM.

Linden, G., Smith, B., and York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80.

Mavalankar, A. A., Gupta, A., Gandotra, C., and Misra, R. (2017). Hotel recommendation system. *Internal Report*.

McFee, B., Bertin-Mahieux, T., Ellis, D. P., and Lanckriet, G. R. (2012). The million song dataset challenge. In *21st International Conference on World Wide Web*, pages 909–916. ACM.

Myers, S. A. and Leskovec, J. (2014). The bursty dynamics of the twitter information network. In *23rd international conference on World wide web*, pages 913–924. ACM.

Navigli, R. and Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Nichols, J., Mahmud, J., and Drews, C. (2012). Summarizing sporting events using twitter. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pages 189–198. ACM.

Pazzani, M. J. and Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer.

Pichl, M., Zangerle, E., and Specht, G. (2014). Combining spotify and twitter data for generating a recent and public dataset for music recommendation. In *Grundlagen von Datenbanken*, pages 35–40.

Pichl, M., Zangerle, E., and Specht, G. (2015). # nowplaying on# spotify: Leveraging spotify information on twitter for artist recommendations. In *International Conference on Web Engineering*, pages 163–174. Springer.

Sawant, S. and Pai, G. (2013). Yelp food recommendation system.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.

Schinas, E., Papadopoulos, S., Diplaris, S., Kompatsiaris, Y., Mass, Y., Herzig, J., and Boudakidis, L. (2013). Eventsense: Capturing the pulse of large-scale events by mining social media streams. In *17th Panhellenic Conference on Informatics*, pages 17–24. ACM.

Shepitsen, A., Gemmell, J., Mobasher, B., and Burke, R. (2008). Personalized recommendation in social tagging systems using hierarchical clustering. In *2008 ACM RecSys*, pages 259–266. ACM.

Trewin, S. (2000). Knowledge-based recommender systems. *Encyclopedia of library and information science*, 69(Supplement 32):180.

Wang, H., Lu, Y., and Zhai, C. (2010). Latent aspect rating analysis on review text data: a rating regression approach. In *16th ACM SIGKDD i*, pages 783–792. ACM.

Yan, M., Sang, J., and Xu, C. (2014). Mining cross-network association for youtube video promotion. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 557–566. ACM.