2014 IEEE International Conference on Data Mining Workshop

Social Preference Ontologies for Enriching User and Item Data in Recommendation Systems

Christopher Krauss
Fraunhofer Institute FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
christopher.krauss@fokus.fraunhofer.de

Stefan Arbanowski
Fraunhofer Institute FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
stefan.arbanowski@fokus.fraunhofer.de

***Abstract*—Some of the known issues of recommendation algorithms are a result of the so called ”Cold Start Problem” that is caused by a lack of sufficient data of users, items or the content, which are essential for the calculation of context-sensitive predictions. Along with this comes the ”Sparsity Problem” which also exposes the problem of recommendation systems which are being provided with too little information of user feedback such as likes and views. As a consequent collaborative and knowledge-based filtering algorithms are unable of precise prediction which is causing a decline of the customer satisfaction. If beyond that there also is a lack of metadata, the calculation of similarities through content-based filtering algorithms is likely to fail as well.**

This paper introduces preference ontologies and how they help to reduce these issues by analyzing external data, in terms of texts from social networks and other web sources. Thereby we introduce a self-designed semantic engine, performing sentiment analysis and semantic keyword extraction. These novel ontologies represent the mined information and thus, describe the users interest in automatic analyzed topics and map them to the meta data of items in recommendation engines.

I. Introduction

In order to improve web services data is being collected from the presented services and products as well as from the users and their way of conduct. The gained information can be used among others for market analyses, for the improvement and optimization of the offer and the products as well as for the personalization and individualization of the services which are provided for the respective user.

Unfortunately, new users registering at web services usually offer, if at all, just a sparse set of personalized information. Actually a common minimum data set is just the user name and the email address. Recommendation engines, however, require at least some information about the users preferences in order to predict fitting items. At the best, this data should be domain specific, as most algorithms for a video web service cannot get productive knowledge out of the users shopping preferences. Moreover, when new products, media items or services become available, predictive data mining needs at least a basic sub set of meta data describing the items concerned. Thus, the principle, the more data, the more accurate the prediction algorithms.

There are various approaches to compensate this sparsity. Some ask the users about the users preferences [1], others offer a common set of items and request to rate them – e.g. [2]. As a third approach some add third party data such like feedback from external databases. Most of the needed data to compensate the Cold Start and Sparsity Problems [3] exists in the World Wide Web, however, it is stored unstructured and in text form of external service providers. There are a few current projects which are analyzing external sources to gain data, however, they mainly concentrate on cleaning up metadata which have been already set up and abstracted by ontologies which are feeding the User-Item-Matrix.

In this paper we want to introduce an approach that analyses texts from third party service providers, such as social networks and review websites, in order to enrich the item and user data of recommender systems and thus allow an adequate personalization. The focus of this explanation is the textual analysis engine for German texts. Beyond direct user input from social networks like Facebook where users are able to like, share or rate items, we want to analyse opinions, in terms of sentiments, and extracted keywords. The underlying approach of generating preference ontologies is introduced in section 3. Section 4 looks at semantic keyword extraction and section 5 at the sentiment analysis as well as the used data sources and frameworks. This will be followed by an explanation of how to map the user's preferences into appropriate item data. Our application prototype for a preliminary analysis is a personal EPG which will be elaborated in more detail in section 7 and finally section 8 concludes with a summary and an outlook.

II. Related Work

As the term “Recommendation Engines” describes a very complex issue with a multitude of subsets and different approaches, it can be subclassified in multiple ways. This subsection will briefly explain the main terms and the related work. Moreover it will provide an introduction on different lexical semantic approaches needed to understand the preference ontology generation first introduced in [4].

A. Recommendations

A Recommendation Engine aims at filtering the most relevant items from the set of all items. Top-N Filtering means that the resulting collection of elements does not contain the whole data set, but only the first N elements of an ordered list. Therefore, the best recommendations only will be finally offered to the user. For instance, the 10 most watched videos or the 3 products with the best average rating will be presented to an anonymous user.

978-1-4799-4274-9/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDMW.2014.76

In Content Based Filtering an element is just compared to another element by considering their content information on meta data. So an item for instance can be very similar or dissimilar to another item. Therefore each attribute has to be examined and compared to the adequate attributes of the other elements. [5]

Collaborative Filtering (CF) is the most common approach for Web 2.0 technologies. The simple comparison of elements is extended by data on consumer behaviour. Thus, a recommendation engine is able to predict items based on characteristics of other users. In CF the user is related to items – often by a value, for instance, the rating as numerical or a Boolean value for "like it" or "don't like it" or even the time a user spends watching this item. So this information can be visualized as a user-item matrix. [6] To find the best fitting items for a user, there are some concrete approaches, such as Item-based Top-N Recommendation Filtering, that filters the items by including the ratings of all other users, and User-based Top-N Recommendation Filtering, that first searches for most similar users and retrieves their best rated items. [6] In contrast to Item-based Filtering, User-based Top-N Recommendation Filtering focuses on finding similar users – called neighbours. So the objective of Neighbourhood-based Collaborative Filtering is to find the nearest neighbour. Afterwards the Top-N items of the nearest neighbours are predicted.

There are some well known problems almost every Recommendation Engine has to deal with. One of them is the Cold Start Problem. At the initial start of a recommender or even when a new item or user registers there is no meta data the engine can work with [7].

Furthermore a recommendation engine has to be capable of compensating the Sparsity Problem [8]. In most cases there is a vast quantity of items as well as many users who, however, lack any coherency. For example, a particular user of a video-on-demand-service only watches less than 1% of the offered items. Some users consume just a few items and even some items have never been watched. So the user-item matrix is really sparse. Jia Zhou and Tiejian Luo are grouping the approaches into two general classes [9]:

- Dimensionality Reduction: Less important rows and columns of the matrix are ignored. This implicates a loss of possible useful information.
- Find and Add Additional Information: By means of the similarity measure rows and columns are filled with values.

This paper focuses on the latter approach. There are some solutions to solve these problems. For instance, the recommender can offer an often used item and wait for the users reaction to rate this item. So the engine receives some feedback at the beginning which will enable the engine to suggest better items to the user. [10]

Rashid et al. [11] classify different strategies to tackle the initial sparsity in user data by offing meaningful items to a new user during the registration process. The user will be asked to rate at least ten movies which were selected for specific strategic reasons, ranging from a random selection, to a popularity-based selection, through to a selection of the most valuable items for the recommendation algorithm. Unfortunately, this process still requires to browse between 50 and 80 different movies. . As a consequence, over 13% of the users taking part in this experiment aborted the registration.

Middleton et al. [12] suggest to use ontological inference in the user profiling process in order to reduce the cold start problem. The performance of their research paper recommendation engine has significantly improved by taking previously browsed research papers and and their classifiers into account. However, this system needs at least some user feedback to perform better in future in order to reduce the cold start problem.

B. *Lexical Semantics*

In Lexical Semantics a sentence, no matter how complex, consists of structured sequences of single words [13]. The combination of these words are, for the time beginning, only understandable for humans.

Thereby, single words can have relations to other words and groups, called ontology, that in turn results in an a so called semantic network or semantic graph representing either the whole language or just domain specific representations. These relations are classified as follows [14], [15], [16]:

- Synonymy is the equality in meaning of two words.
- Polysemy characterizes a word with multiple different semantics, but a unique word origin.
- Homonymy characterizes the equality in pronunciation and spelling, but with a different semantic of two words. The origins of both words are diverse.
- Hyperonymy and Hyponymy are to describe a hierarchic relationship between a superordinate (hyperonym) and a subtopic (hyponym).
- Holonymy and Meronymy are to describe a hierarchic relationship between the whole thing (holonym) and a subset (meronym).
- Antonymy stands for the opposite in meaning of two words.
- Associations are unspecific relations of words sharing a common context. The definition of further meta data allows to concretize the type of relation.

More and more machine learning algorithms try to analyze the meaning of the sentences in order to extract the intended semantic and to make it interpretable even for computers. Natural Language Processing (NLP) describes the set of approaches for information retrieval from texts. Brants stated the following approaches to process texts[17]:

- Removal of stop words to increase the systems performance
- Stemming: to reduce a word to its word stem, the so called "lemma"
- Part of Speech (POS) Tagging: Allocate a word to its part of speech
- Identify Compounds and Statistical Phrases to treat them as units

- Compound Splitting for differing multiple semantics of a concatenated word
- Chunking and Shallow Parsing separate words, sentences or whole text corpora into sub sets representing only cohesive topics and semantics
- Word Sense Disambiguation aims at identifying the correct sense of homonyms

Moreover, Aussenac-Gilles et al. stated the Term Extraction technique for automatic recognition of important words [18].

Sentic computing is defined as "the analysis of natural language [...] based on affective ontologies and common-sense reasoning tools" [19]. Thereby Cambria et al. [20] classified opinion mining into keyword spotting, lexical affinity and statistical methods. Moreover, they introduced an engine performing opinion mining in four steps: it skims the opinion, extracts concepts from the concerned corpus, infers semantics with the concepts and then afterwards extracts the sentics.

A polarity analysis describes an approach to link terms and phrases to numerical values indicating the closeness to one out of two reference poles. These two poles must be antonyms, e.g. "good" and "bad", "passive" and "active", "week" and "strong" or "forgettable" and "unforgettable". Sentiment analysis is an approach to allocate the author's opinion of his writing to a numerical value [21]. The bi-polar polarities, called sentiments, are in most cases "positive" and "negative".

In [22] Cambria et al. identified the relevance of context- and intent-level analysis for opinion mining and sentiment analysis. That, in turn, will help to extract meaningful information from different media formats, e.g. video, audio, texts etc., out of every-day life services, such as commerce, tourism, education and health.

III. Social Preference Ontologies

We focus on specific domains and categories of texts with the highest estimated value for recommendation engines. The concerned texts are first and foremost comments of the user and his friends from social networks. The precondition is that the user has logged in with his user credentials at the respective social network and thus, has allowed the system to access his personal information. Moreover, the semantic engine analyzes review websites and other textual data to enrich the item data set.

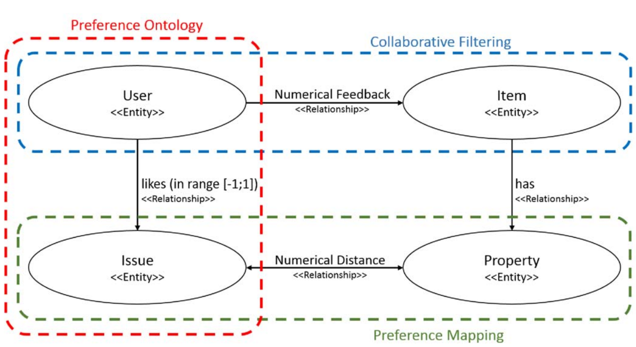

Fig. 1: Preference Ontology

Picture 1 shows preference ontologies, first introduced in [4]. Thereby the blue box indicates the well-known numerical feedback between users and items in collaborative filtering approaches. Every feedback is stored in an user-item-matrix, where n users are connected to m items in an explicit (e.g. rating, likes etc.) or an implicit way (e.g. views, consumption etc.). In case of the cold start problem this user-item-matrix is very sparse.

The red box shows the novel approach of preference ontologies representing identified topics a user is interested in and the user's opinion (as numerical value) on this issues. External data sources are analyzed to retrieve all important topics the user is writing about (shown as issue entity). These topics are connected to the user via a like relationship, representing the sentiment value in range of [-1;1] where -1 is the worst and 1 is the most positive opinion on this issue. A user can be interested in as many topics as the automatic analysis determines.

The last step (in the green box) is to map an issue with one or more items (or their properties). This mapping value is represented by a normalized numerical distance, where 0 means the preference is equal to the property – e.g. an analysis of Facebook estimates a high interest of the current user in action movies and thus, movie items with the genre action are similar. The higher the values, the more diverse the entities. Thereby it does not matter if it is a semantic distance or a distance based on common usage transactions.

The main aspect of this paper is the semantic engine introduced here. This engine is designed to extract meaningful data out of unstructured texts from existing web services which hence provides the needed data for the ontology generation. It consists of two core components. The first one tries to identify semantic keywords from texts, summarizing the topic by searching for the most meaningful terms that represent this text. The second approach tries to allocate an implicit feedback, in terms of the authors opinion to the identified topics (sentiment analysis).

IV. Keyword Detection

Our semantic keyword detection algorithm tries to identify important words that summarizes the core ideas of a text. Therefore we assume a manual chunking of the text corpora. Textual reviews of movies, for example, normally consist of a synopsis and the actual critic. While the synopsis is valuable for the keyword detection, it causes an offset in the authors sentiment as most movies are about to solve a problem. The review sections in turn are more appropriate for sentiment analysis than for a semantic keyword detection.

Important words, such as nouns, proper nouns and adjectives, are matched with a self made graph database containing all the words and relations from OpenThesaurus [16], a German word net with synonyms, hyperonyms and hyponyms, as well as with GermaNet [15], a lexical semantic net for German similar to its English equivalent WordNet [23]. The implemented processing algorithms of the graph database retrieve a set of terms representing the hyperonyms of all words by recursively analyzing the respective superordinate, its

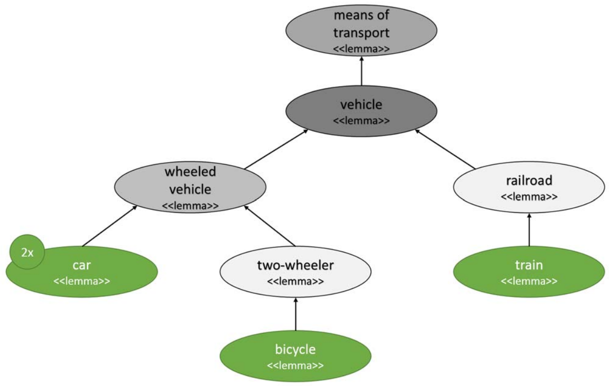

Fig. 2: Example of semantic keywords analysis

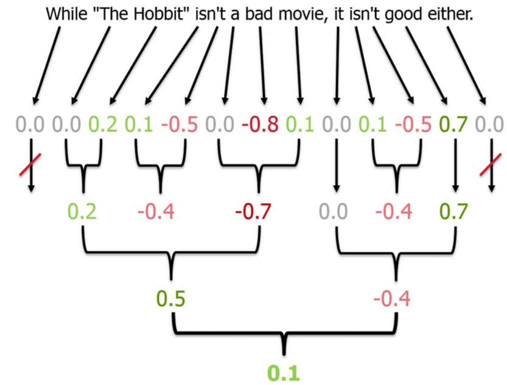

Fig. 3: Example of sentiment calculation

superordinate and so forth. Actually this approach is designed for German texts, but should also work when using word nets of other languages like English.

An example of the keyword detection process is shown in figure 2. Important words appearing in the original text are marked in green: e.g. two times the word "car", one time "bicycle" and one time "train". The terms in grey indicate hyperonyms – from specific umbrella terms at the bottom up to more general terms at the top. The color strength, in turn, portraits the relevance value of each hyperonym – the darker, the more relevant.

The calculation of the "nearest common hyperonym" of two words is shown in the following formula and represents the relevance value for a keyword in the given context:

$$value_{keyword}(h) = \sum_{n=0}^{|W_h|}(1 + \frac{1}{1 + e_{w,h}}) \tag{1}$$

h is the keyword to be determined. W_h is the set of important words w in the analyzed text having h as a hyperonym and thus $|W_h|$ is its amount. $e_{w,h}$ is the number of edges between the word w and its hyperonym h. The weight of a single edge is 1. The higher the resulting $value_{keyword}$, the more important the keyword h.

In our example, the term "vehicle" has the highest predicted $value_{keyword}$ with 5.25, followed by "means of transport" (4.95) and "wheeled vehicle" (4.33). In contrast to the original words "car" (4.00) and "bicycle" as well as "train" (each 2.00), the terms "two-wheeler" and "railroad" only retrieve a value of 1.50 and consequently, are less significant.

This approach is used to predict the keywords of a corpus automatically, that represent the topics of the given texts. The amount of relevant keywords depends on the text size and the calculated keyword value. Therefore future work is needed to identify reasonable thresholds.

V. Opinion Mapping

In order to retrieve the user's opinion in his written text and thus, implicitly on the mined keywords, we use a self-designed sentiment analysis engine. The algorithms process any text corpus and returns a numerical value in the range of $[-1, 1]$, where -1 is worst opinion and 1 is the best. This sentiment value represents the like relationship of the preference ontologies.

A. Approach

The TreeTagger [24], also used for the keyword detection, retrieves word stems and the part of speech of all given terms. The next step is to get sentiment values for all word stems. Therefore, we use SentiWS [25] (a sentiment database offering 31,000 lemmas and automatic calculated polarities) as well as the Polarity database introduced in [26] (consisting of 8,000 lemmas with manually allocated polarities).

Our sentiment algorithm consists of 4 different approaches to combine the single polarity values:

- $[Statistic]$ A statistical approach averaging the known polarity values in a sentence.
- $[StatisticAll]$ Another statistical approach averaging all polarity values. If no value could be allocated, the polarity value is set to 0.0.
- $[Klenner]$ A grammar with a set of rules for the German language to compose the sentiment of phrases designed by Klenner et al. [27]. A simple example is the combination of a negative adjective and a positive verb result in a negative combination of the sentiment.
- $[Alexis]$ A self designed grammar for recursively combining phrases with allocated polarities. It is similar to the approach by Klenner, but takes shift words (e.g. "not" turns the sentiment) as well as intensifier (e.g. "very") and reducer (e.g. "little") into account. After composing sentiment values of single words to a combined sentiment, this step is repeated with combined sentiments until there is only one sentiment left for the whole sentence.

An example for calculating sentiments is displayed in picture 3. In step one the system tries to allocate a sentiment for every single word (shown as numerical values) by looking up the databases. Step two, three and four recursively combine the single sentiments by using one of the before mentioned approaches. The steps are repeated until there is only one overall value left.

A first case study on the sentiment analysis indicates that a polarity value can only be allocated to the given lemmas in 16.6 %. Therefore, using related words in order to search for an alternative polarity value strikes to be a reasonable idea. That is why we use settings where GermaNet and OpenThesaurus retrieve synonyms and hyperonyms of these lemmas and search again for their polarity. A found polarity value is allocated to the original lemma and used for further calculations. The resulting settings for the algorithms are:

- $[NoSyn]$ Calculation without lemma substitution
- $[OpenThesaurus]$ Substitution with OpenThesaurus relations
- $[GermaNet]$ Substitution with GermaNet relations
- $[GermaThesaurus]$ Substitution with relations from GermaNet and OpenThesaurus

An alternative approach is introduced by Kamps et al. [28] when translating the whole word net of a language into a graph database. By using reference terms, such as bi-polar entities (”positive” and ”negative”), it allows to identify the numerical value of closeness to one of both terms. The Dijkstra algorithm [29] identifies the shortest distance between all three terms (the lemma concerned on the two reference terms). This formula allows to automatically calculate a sentiment for each word in a word net:

$$polarity(l) = \frac{dist(l, r_1) - dist(l, r_2)}{dist(r_1, r_2)} \tag{2}$$

where l is the lemma concerned and r_1 and r_2 are the two bi-polar reference terms. $dist$ is the number of edges on the shortest track between the two given terms. $polarity$ is the numerical sentiment in range of $[-1, 1]$ for the concerned lemma.

B. Results of the case study

We have analyzed 948 German texts taken from different sources that can be categorized in comments of social media, reviews of products, services and media items as well as news. The social texts originate from personal user accounts of our colleagues, the only requirement is: It must be a German text allowing a clear assignment to a sentiment.

- Social media comments from Facebook.de, Twitter.de
- Reviews from Amazon.de (products), Chip.de (products), Connect.de (products), Douglas.de (products), ebay.de (products), Filmstarts.de (movies), Lieferheld.de (bring service), MoviePilot.de (movies) and Testberichte.de (products)
- News from Sportbild.de, Welt.de

For the evaluation we manually labeled an expected sentiment value in range of $[-1, 1]$ for each text and compared this value to the calculated sentiment. Table 1 shows the results. As you can see, the Mean Absolute Error (MAE) is small for each algorithm and setting, but the deviation to the Root Mean Square Error RMSE indicates that results are very scattered. The mean average of all algorithms indicates a better performance then single ones, as it minimizes outliers. The optimzed weighted average, in contrast, uses only the following algorithms, as they minimize the error values the best: [StatisticAll, NoSyn], [StatisticAll, OpenThesaurus], [Klenner, GermaNet], [Alexis, OpenThesaurus].

Table 2 shows results grouped by service providers and other parameters, such as word count and labeled sentiment. As you can see, the sentiment analysis works well for some service providers: Most service providers, having a good accuracy, offer product reviews Connect.de, Chip.de and Testberichte.de, but the lowest values originate from other product sellers (Amazon.de, Douglas.de and eBay.de), too. Social Networks show the highest average value and thus, seem to allow an accurate prediction of the users interest in a specific topic.

As you can see, the system performs better when processing less words (the optimum is between 3 and 50 words), in general, however, the text size does not have a huge impact on the accuracy. In turn, the expected sentiment plays a key role, as a negative sentiments seem to be very unpredictable. All in all, accuracy based on MAE when using a hybrid algorithm is about 84.123% of right allocated sentiments.

VI. Preference Mapping

The so called Preference Filtering is a mixture of Content-based and Collaborative Filtering. [30] Its objective is to find items that fit to the users' preferences. Basically we want to find out how many of the determined criteria/attributes are matching a certain TV program.

We implemented Association Rules to identify relationships between apparently incoherent terms, such as ”sports”, ”iPhone” and ”How I Met Your Mother”. These Association Rules are aiming to disclose highly represented relations (so called transactions) between a user and the items (in this case terms). [6] [31] The resulting set of frequent items may, in addition, be scanned for some rules, so that afterwards the list of transaction can be divided in causations and consequence. For instance, when a defined number of users have watched the program ”How I Met Your Mother” and afterwards talked about ”iPhones” and ”sports” on Facebook, the resulting frequent item set could be: *The terms ”How I Met Your Mother”, ”iPhones” and ”sports” are often used together.*

Moreover, if a user has talked about a subset of these transactions, an association rule could be: *You watched ”How I Met Your Mother” and talked about ”sports”, thus you may also like ”iPhones”.* This can be formally written as {I item1, I item2} ⇒ {I item3} or more generally:

$$X \Rightarrow Y \tag{3}$$

The item set X implies the item set Y. [32]

VII. Evaluation

These preliminary results serve only to indicate the possible strength and weakness of such a system and to explain the main ideas to interested persons. A large scale user test as well as accuracy calculations and comparisons will follow in future work.

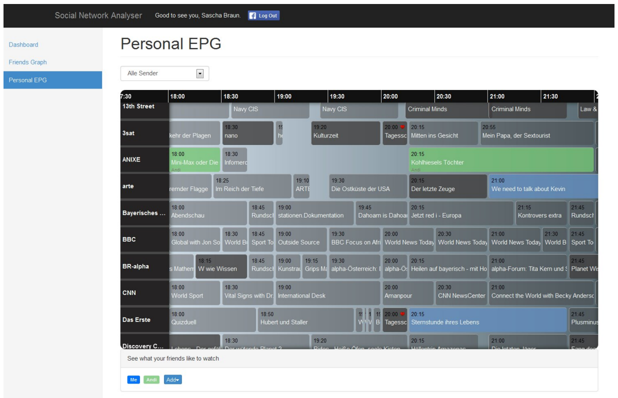

Fig. 4: The personal EPG

A. Prototype

We implemented an Electronic Program Guide (EPG) that recommends TV programs by extracting the users preferences in Facebook. Thereby we analyze all of the users' posts with our semantic engine and try to find topics which could match genres or movie descriptions. Moreover, we also determine the friends graph, find clusters and identify the most important friends, based on the number of common friends and number of interactions (comments, likes, messages) in order to analyze their posts.

The TV data is aggregated and pre-selected by the TV Predictor recommendation engine introduced in [2]. On the one hand it delivers all necessary meta data like title, channel, air time, genre and description. On the other hand it contains ratings for each program item. Those ratings are being calculated by collaborative filtering and represent an average value which could match anyone. To make it easy for the user to see what is relevant we defined a color scheme. By default all programs are being light gray. Ratings from the TV Predictor are being highlighted with a darker gray depending on the rating (actually 1 to 5, but we only highlight from 4 to 5 as these are the ones being a real recommendation). The recommendations based on the users Facebook profile are highlighted in various gradations of blue. Additionally it is possible to see what the user's friends could like. This might be useful if the user has a rather empty profile with only few comments or likes. This type of recommendations is being highlighted in green. Below the EPG is a dropdown list with all of the user's friends from which he can chose as many as he likes to. To make the recommendations more comprehensible, the names of the friends on who the recommendation is based on will be displayed.

In order to ensure a high level of privacy no user data is sent to our servers. The filtering is done offline in the user's browser. The colors represent a filter, in terms of CSS classes applied to the items concerned on demand. This will result in higher acceptance and trust as well as a better performance.

B. Results

Our evaluation indicates that users will receive a sufficient amount of recommendations even with a low degree of social network activities. Recommendations with only few matches are considered to be weak according to our color scheme. The more active a user is in terms of posting and liking, the better recommendations can become. Additionally, the quantity of the user's likes will decrease the necessity of relying on friend's information in order to provide the user with suitable recommendations.

As a test case we used about 2500 programs of 70 different German channels. The user with the fewest amount of interactivity received only 0.8% (from the 2500 programs) of weak, 0.5% of medium and 0.5% strong recommendations. The user with the highest activity level (i.e. many likes, posts, comments etc.) received 1.1% of strong, 8% of medium and 6% of weak recommendations. This first case study indicates that using textual analysis of Social Networks could compensate the cold start and sparsity problems predicting automatically the interest in otherwise unknown items by 1.8% up to 15.1%. As a result users can start with a personalized service without giving explicit feedback and by only logging into a Social Network. Since these values also differ depending on the amount of friends a user has and their respective activity levels, further evaluations need to be conducted.

VIII. Outlook and Future Work

Our paper presented a novel approach to reduce the data sparsity and to compensate the cold start problem in recommendation systems by using social preference ontologies. Therefore, texts of the user in social networks have been analyzed in order to detect important keywords and allocate the users opinion on these keywords, in terms of a sentiment value in a range of [-1,1]. A first evaluation with a social Electronic Programm Guide indicates the possible advantages when taking social topics into account. This system allows a personalization of 1.8 % up to 15.1% of the available items – depending on the social activity level of the current user. A larger case study will follow in order to produce more representative results that can also be compared to alternative solutions for compensating the cold start problem.

Moreover, there are a lot of extensions planned to increase the performance and accuracy of the system. First and foremost we need to identify domain specific corpora. An analysis of the Speech Acts [33] could help to select, chunk and pre-process adequate text sections. Another interesting research topic for analyzing comments in social networks is the identification of different languages and even colloquial language. Thus, we could trace back all given words to the lemmas concerned and increase the performance for a wide application area.

Acknowledgment

The research leading to these results has received funding from the European Unions Seventh Framework Program (FP7-ICT-2013-10) under grant agreement number 611001 (UCN project). We thank all our project partners within the UCN project towards the realization of this work. Moreover, we would like to thank the FOKUS colleagues who contributed to this project – especially Sascha Braun, Oemer Duranoglu, Rahel Freist-Held and Natalie Nik-Nafs – as well as all participants of the case studies.

REFERENCES

[1] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl, "Getting to know you: Learning new user preferences in recommender systems," ser. IUI '02. New York, NY, USA: ACM, 2002, pp. 127–134. [Online]. Available: http://doi.acm.org/10.1145/502716.502737

[2] C. Krauss, L. George, and S. Arbanowski, "Tv predictor: personalized program recommendations to be displayed on smarttvs," in *BigMine'13*. ACM, 2013, pp. 63–70.

[3] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, "Addressing cold-start problem in recommendation systems," in *ICUIMC-2008*. ACM, 2008, pp. 208–211.

[4] C. Krauss, S. Braun, and S. Arbanowski, "Preference ontologies based on social media for compensating the cold start problem," in *SNAKDD'14*. ACM, 2014.

[5] Z. Liu, H. Wang, W. Qu, and W. Liu, "Sparse matrix prediction filling in collaborative filtering," *IEEE International Conference on Scalable Computing and Communications; The Eighth International Conference on Embedded Computing;*, pp. 304–307, 2009.

[6] G. Linden, B. Smith, and J. York, "Amazon.com recommendations - item-to-item collaborative filtering," *IEEE Internet Computing*, pp. 76–80, 2003.

[7] O. H. Embarak, "A method for solving the cold start problem in recommendation systems," *IEEE - International Conference on Innovations in Information Technology*, pp. 238–243, 2011.

[8] J. Zhou and T. Luo, "Towards an introduction to collaborative filtering," *International Conference on Computational Science and Engineering*, pp. 576–581, 2009.

[9] ——, "A novel approach to solve the sparsity problem in collaborative filtering," *IEEE*, pp. 165–170, 2010.

[10] H. Kim, J. Kim, and J. Herlocker, "Feature-based prediction of unknown preferences for nearest-neighbor collaborative filtering," *Proceddings of the Fourth IEEE International Conference on Data Mining*, vol. IV, 2004.

[11] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl, "Getting to know you: learning new user preferences in recommender systems," in *Proceedings of the 7th international conference on Intelligent user interfaces*. ACM, 2002, pp. 127–134.

[12] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure, "Ontological user profiling in recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 54–88, 2004.

[13] T. E. Zimmermann, *A Gentle Introduction to Formal Semantics – Einfuehrung in die Semantik*. Wissenschaftliche Buchgesellschaft, Muenchen, BY, Germany, 2014.

[14] C. Mangold, "A survey and classification of semantic search approaches," *International Journal of Metadata, Semantics and Ontologies*, vol. 2, no. 1, pp. 23–34, 2007.

[15] B. Hamp and H. Feldweg, "Germanet-a lexical-semantic net for german," in *ACL/EACL 97*. Citeseer, 1997, pp. 9–15.

[16] D. Naber, "Openthesaurus: ein offenes deutsches wortnetz," *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen: Beiträge zur GLDV-Tagung, Bonn, Germany*, pp. 422–433, 2005.

[17] T. Brants, "Natural language processing in information retrieval." in *CLIN*, 2003.

[18] N. Aussenac-Gilles and D. Sörgel, "Text analysis for ontology and terminology engineering," *Applied Ontology*, vol. 1, no. 1, pp. 35–46, 2005.

[19] E. Cambria and A. Hussain, *Sentic computing – Techniques, Tools, and Applications*. Springer, 2012, vol. 2.

[20] E. Cambria, H. Wang, and B. White, "Guest editorial introduction: Big social data analysis," *Knowledge-Based Systems 69*, pp. 1–2, 2014.

[21] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[22] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent Systems*, vol. 28, pp. 15–21, 2013.

[23] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[24] H. Schmid, "Treetagger— a language independent part-of-speech tagger," *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*, vol. 43, 1995.

[25] R. Remus, U. Quasthoff, and G. Heyer, "Sentiws-a publicly available german-language resource for sentiment analysis." in *LREC*, 2010.

[26] S. Clematide and M. Klenner, "Evaluation and extension of a polarity lexicon for german," in *ACL - WASSA'10*, 2010, p. 7.

[27] M. Klenner, A. Fahrni, and S. Petrakis, "Polart: A robust tool for sentiment analysis," in *NoDaLiDa'09*, vol. 4, 2009, pp. 235–238.

[28] J. Kamps, M. Marx, R. J. Mokken, and M. De Rijke, "Using wordnet to measure semantic orientations of adjectives," 2004.

[29] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[30] N. Tintarev and J. Masthoff, "A survey of explanations in recommender systems," *Data Engineering Workshop, 2007 IEEE*, pp. 801 – 810, 2007.

[31] G. Liu, H. Jiang, R. Geng, and H. Li, "Application of multidimensional association rules in personal financial services," *IEEE - International Conference On Computer Design And Appliations*, pp. 500–503, 2010.

[32] J. Han, "Learning fuzzy association rules and associative classification rules," *IEEE - International Conference on Fuzzy Systems*, pp. 1454–1459, 2006.

[33] J. R. Searle, *Speech acts: An essay in the philosophy of language*. Cambridge university press, 1969, vol. 626.

TABLE I: Blank, averaged and weighted results

Algorithm	Average Sentiment	MAE All	Accuracy (MAE) in%	RMSE All	Accuracy (RMSE) in%
Labeled Sentiment	0.196				
$[Statistic, NoSyn]$	0.732	0.587	70.645	0.766	61.689
$[Statistic, OpenThesaurus]$	0.751	0.605	69.701	0.778	61.077
$[Statistic, GermaNet]$	0.755	0.751	62.451	0.866	56.670
$[StatisticAll, NoSyn]$	0.092	0.353	82.342	0.594	70.287
$[StatisticAll, OpenThesaurus]$	0.104	0.354	82.284	0.595	70.237
$[StatisticAll, GermaNet]$	0.113	0.357	82.139	0.597	70.115
$[Klenner, NoSyn]$	0.063	0.551	72.442	0.742	62.880
$[Klenner, OpenThesaurus]$	0.137	0.520	74.016	0.720	63.955
$[Klenner, GermaNet]$	0.026	0.517	74.167	0.718	64.060
$[Alexis, NoSyn]$	-0.513	0.890	55.503	0.943	52.831
$[Alexis, OpenThesaurus]$	0.024	0.336	83.189	0.579	71.007
$[Alexis, GermaNet]$	0.016	0.347	82.649	0.589	70.545
$[Statistic, Average]$	0.746	0.596	70.200	0.772	61.399
$[StatisticAll, Average]$	0.103	0.633	68.375	0.795	60.234
$[Klenner, Average]$	0.075	0.503	74.846	0.709	64.536
$[Alexis, Average]$	-0.157	0.520	73.980	0.721	63.930
$[Average, NoSyn]$	0.228	0.323	83.842	0.568	71.576
$[Average, GermaNet]$	0.254	0.325	83.733	0.570	71.480
$[Average, OpenThesaurus]$	0.228	0.325	83.748	0.570	71.493
Mean Average	0.191	0.358	82.082	0.598	70.068
Optimized weighted Average	0.237	0.318	84.123	0.564	71.825

TABLE II: Categorized results

Optimized Average for	Labeled Sentiment	MAE All	Accuracy (MAE) in%	RMSE All	Accuracy (RMSE) in%
Amazon.de	0.328	0.445	77.723	0.667	66.625
Chip.de	0.210	0.200	89.958	0.448	77.593
Connect.de	0.212	0.200	90.000	0.447	77.639
Douglas.de	0.198	0.532	73.370	0.653	67.333
eBay.de	0.055	0.472	76.368	0.687	65.625
Facebook.de	0.040	0.268	86.563	0.518	74.080
Filmstarts.de	0.287	0.341	82.903	0.584	70.763
Lieferheld.de	0.120	0.377	81.125	0.614	69.279
MoviePilot.de	0.290	0.315	84.225	0.561	71.915
Sportbild.de	0.063	0.349	82.545	0.590	70.458
Testberichte.de	0.252	0.247	87.602	0.497	75.102
Twitter.de	0.064	0.264	86.795	0.513	74.304
Welt.de	0.109	0.078	96.058	0.280	85.961
3 to 20 words	0.100	0.293	85.311	0.542	72.899
21 to 50 words	0.149	0.288	85.555	0.537	73.125
51 to 100 words	0.219	0.326	83.666	0.571	71.422
101 to 760 words	0.340	0.376	81.199	0.613	69.340
Negative $[-1, -0.333]$	-0.614	0.673	66.302	0.820	58.952
Neutral $[-0.334, 0.333]$	0.044	0.230	88.455	0.480	75.974
Positive $[0.334, 1]$	0.668	0.371	81.437	0.609	69.534