Selecting Keywords for Content Based Recommendation

Christian Wartena
Novay
Brouwerijstraat 1
7523 XC Enschede
The Netherlands
Christian.Wartena@novay.nl

Wout Slakhorst
Novay
Brouwerijstraat 1
7523 XC Enschede
The Netherlands
Wout.Slakhorst@novay.nl

Martin Wibbels
Novay
Brouwerijstraat 1
7523 XC Enschede
The Netherlands
Martin.Wibbels@novay.nl

ABSTRACT

The continued growth of online content makes personalized recommendation an increasingly important tool for media consumption. While collaborative filtering techniques have shown to be very successful in stable collections, content based approaches are necessary for recommending new items. Content based recommendation uses the similarity between new items and consumed items to predict whether a new item is interesting for the user. The similarity is computed by comparing the content or the meta-data of the items. In this paper we consider recommendation of TV-broadcasts for which meta-data and synopses are available. We thereby concentrate on the new item problem. We investigate the value of different types of meta-data provided by the broadcaster or extracted from synopsis. We show that extracted keywords are better suited for recommendation than manually assigned keywords. Furthermore we show that the number of keywords used is of great importance. Using a rather small number of keywords to present an item yields the best results for recommendation.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: [Indexing Methods]; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

General Terms

Algorithms, Experimentation

Keywords

Keyword Extraction, Recommendation

1. INTRODUCTION

The continued growth of online content makes personalized recommendation an important tool for media consumption. While collaborative filtering techniques have been shown to be very successful in stable collections, content based approaches are necessary for recommending new items.

In this paper we will focus on the top-n item prediction. Especially, we are interested in recommending new items. Therefore we split our datasets in training and test set such that the test set contains only items that are not in the training set. Content based item prediction uses the similarity between a new item and consumed items to generate a list of the most interesting items for each user. The similarity is computed by comparing the different meta-data of the different items. In this paper we consider the recommendation of TV-broadcasts and movies for which meta-data and synopses or plots are available.

The main contribution of this paper is the comparison of different types of meta-data, especially keywords, for item representation in the context of content based recommendation. We do not try to find a set of keywords that gives the best result, like in classical feature selection approaches. Rather we compare different types of keywords, like manually assigned or automatically extracted keywords. We show that extracted keywords from our dataset are better suited for recommendation than the manually assigned ones. Furthermore we show that representing an item by more words does not necessarily increase the quality of recommendation. Using a rather small number of keywords to present an item yields the best results for recommendation.

The remainder of this paper is organized as follows. In Section 2 we discuss related work. In section 3 and 4 we present the algorithms which have been used for recommendation and keyword extraction respectively. In section 4.2 we discuss the use of topic modeling as alternative to keyword extraction. In the final section 5 we study the effect of using different types of meta-data and different numbers of keywords on content based recommendation.

2. RELATED WORK

In most work on content based recommendation the available meta-data are taken as granted. In cases in which there are textual descriptions of the items, terms from the text are usually weighted using tf.idf weights or information gain ([11]). Words with low weights are usually removed, but still a relatively large number of words (100 or more [11]) is used for representation of the text. We are not aware of systematic studies comparing different weighting schemes in the context of recommendation, like they exist e.g. for text classification ([6]).

Fleischman and Govy [5] compare different similarities for IMDB movies using genres and word vectors from the plot.

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0099-5/10/10 ...$10.00.

They use human judgment for evaluation. Debnath e.a. [4] address the issue of optimal combination of different features. However, they do not compare alternative variants of the same type of information.

3. CONTENT BASED RECOMMENDATION

The main focus of this paper is to investigate the influence of selecting meta-data for content based recommendation. We will show how the selection of meta-data for the computation of distances between items influences the quality of recommendation for two recommendation strategies on two different datasets.

The first recommendation strategy we use is a straightforward nearest neighbor approach for recommendation ([12]). Content based nearest neighbor approaches are similar to classical nearest neighbor or collaborative filtering algorithms, but the similarity measure between items is based on the content of the items and not on the ratings. To be more precise, let I be a set of n items, U a set of m users and $R \in \mathbb{R}^{n \times m}$ the ratings assigned by the users to each item. For each item $i \in I$ let v_i be the vector representing i. Now we define the distance of an item $i \in I$ to a user $u \in U$ as

$$D(u,i) = \frac{\Sigma_{j \in I} R_{uj} d(v_j, v_i)}{\Sigma_{j \in I} R_{uj}}. \tag{1}$$

In many cases we only know whether a user has seen an item or not. In this case each rating is always 0 or 1. We divide by the sum of the ratings to get an average distance, which in fact is not necessary if we rank results for one user. If we would use similarities rather than distances and divide by the sum of the similarities instead, we would predict a rating that is largely determined by the *nearest neighbors* that have the largest weights.

In the nearest neighbor approach we compute the average distance of an item to the consumed items. Alternatively, we can compute the distance of an item to the "average" consumed item. For each item i we have a distribution over keywords (or other meta data), p_i. For each user u we define as well a distribution over keywords p_u as the weighted average of the distributions of the items he has rated. The ratings are used as weights. The distance between a user and an item is now defined as:

$$D(u,i) = d(p_i, p_u) \tag{2}$$

where we use Pearson's coefficient to compute $d(p_i, p_u)$. We will call the distribution p_u the profile of u and the recommendation strategy the *profile based* recommendation.

4. KEYWORD EXTRACTION

For all items in our datasets a short textual description is available. We extract words from these texts to represent the text as a vector in a word space. We can either use all words (after removing stop words) or only a small selection.

For keyword extraction we compare two different extraction methods. Both methods are based on ranking words and selecting the top n ranked words. The first method uses standard tf.idf ranking. The tf.idf value of a term t in a document d is defined as

$$tf.idf(t,d) = n(d,t) \frac{1}{\log df(t)} \tag{3}$$

where $n(d,t)$ is the number of occurrences of w in d, and df is the number of documents d' for which $n(d',t) > 0$.

The second method tries to determine how characteristic a word is for a given text. We have motivated and presented this method in detail in [13]. The basic idea is that we represent each term t by a distribution of terms that is typical for the documents in which t occurs. This distribution is called the *co-occurrence distribution* of t. A term is considered to be a good keyword for a document if its co-occurrence distribution is similar to the distribution of terms in the document. However, instead of using the term distribution of the document we use a smoothed variant of this distribution.

4.1 Distribution of Co-occurring Terms

We simplify a document to a bag of words. Consider a set of n term occurrences $\mathcal{W}$ each being an instance of a term t in $\mathcal{T} = \{t_1, \dots t_m\}$, and each occurring in a source document d in a collection $\mathcal{C} = \{d_1, \dots d_M\}$. Let $n(d,t)$ be the number of occurrences of term t in d, $n(t) = \sum_d n(d,t)$ be the number of occurrences of term t, $N(d) = \sum_t n(d,t)$ the number of term occurrences in d and n the total number of term occurrences in the entire collection.

We define three (conditional) probability distributions

$$q(t) = n(t)/n \qquad \text{on } \mathcal{T} \tag{4}$$

$$Q(d|t) = n(d,t)/n(t) \qquad \text{on } \mathcal{C} \tag{5}$$

$$q(t|d) = n(d,t)/N(d) \qquad \text{on } \mathcal{T} \tag{6}$$

Probability distributions on $\mathcal{C}$ and $\mathcal{T}$ will be denoted by P, p with various sub and superscripts .

Consider a Markov chain on $\mathcal{T} \cup \mathcal{C}$ having transitions $\mathcal{T} \to \mathcal{C}$ with transition probabilities $Q(d|t)$ and transitions $\mathcal{C} \to \mathcal{T}$ with transition probabilities $q(t|d)$ only. Given a term distribution $p(t)$ we compute the one step Markov chain evolution. This gives us a document distribution $P_p(d)$:

$$P_p(d) = \sum_t Q(d|t)p(t). \tag{7}$$

Likewise given a document distribution $P(d)$, the one step Markov chain evolution is the term distribution

$$p_P(t) = \sum_d q(t|d)P(d) \tag{8}$$

Since $P(d)$ gives the probability to find a term occurrence in document d, p_P is the weighted average of the term distributions in the documents. Combining these, i.e. running the Markov chain twice, every term distribution gives rise to a new term distribution

$$\bar{p}(t) = p_{P_p}(t) = \sum_{t',d} q(t|d)Q(d|t')p(t') \tag{9}$$

For some term z, starting from the degenerate term distribution $p_z(t) = p(t|z) = \delta_{tz}$ (1 if $t = z$ and 0 otherwise), we get the *distribution of co-occurring terms* or *co-occurrence distribution* $\bar{p}_z$

$$\bar{p}_z(t) = \sum_{d,t'} q(t|d)Q(d|t')p_z(t') = \sum_d q(t|d)Q(d|z). \tag{10}$$

This distribution is the weighted average of the term distributions of documents containing z where the weight is the probability $Q(d|z)$ that an instance of term z has source d. Likewise, we can run the Markov chain twice on the document distribution $q_d(t)$, which by linearity results in the weighted sum of the co-occurrence distributions:

$$\bar{q}_d(t) = \sum_{d',t'} q(t|d')Q(d'|t')q(t'|d) = \sum_z q(z|d)\bar{p}_z(t). \quad (11)$$

The distribution $\bar{q}_d$ can be seen a smoothed version of the document distribution q_d.

4.1.1 *Co-occurrence based keyword extraction*

The similarity between the co-occurrence distribution of a word and the document distribution is a good indication of how characteristic a word is for the document. There are various options to compute the similarity between two distributions. In [13] we have shown that the following correlation coefficient gives the best results:

$$r(z,d) = \frac{\sum_t (\bar{q}_d(t) - q(t))(\bar{p}_z(t) - q(t))}{\sqrt{\sum_t (\bar{q}_d(t) - q(t))^2}\sqrt{\sum_t (\bar{p}_z(t) - q(t))^2}}. \quad (12)$$

This coefficient captures the idea that two distributions are similar if they diverge in the same way from the background distribution q.

4.2 Topic modeling

Another approach in using item descriptions is to extract topics from the synopses. A recent technique for topic detection with promising results is Latent Dirichlet Allocation (LDA; [3]). We have used an open source implementation for LDA from the MALLET toolkit ([9]).

Before applying LDA we removed all closed class words (articles, prepositions etc.) and a number of stop words. We computed 40 clusters. The top n clusters for an item where added as generated topics for that item. These topics were treated as keyword by the recommendation algorithms.

5. EXPERIMENTAL RESULTS

The different keyword extraction strategies are implemented in a UIMA ([1]) text analysis pipeline. All words in the text are stemmed using the tagger/lemmatizer from [7] and tagged by the Stanford part of speech tagger ([2]). In order to compute co-occurrence distributions all open class words are taken into account

5.1 Datasets

We have used two different datasets for evaluation. As a first dataset we have used BBC audience research data collected in May 2008. The dataset contains 1408 programs and has information about 2166 users. Each user has rated 20 programs on average. Each item has manually assigned keywords, genre labels and a synopsis. The length of the synopses varies from a few words to several sentences. Only the synopses are used for keyword extraction.

The dataset was split into two by choosing date and time such that 75% of the items was broadcast before and 25% after that moment, this gives us our training and test set, respectively. There is no overlap between the items of the test and the training set. Recommendation performance therefore depends completely on the ability of the algorithm to deal with the new item problem. The training set contains 18 428 and the test set 5418 ratings.

The second dataset we have used is derived form the 10 Million rating dataset from MovieLens ([10]). We have extended this dataset with the plot descriptions of the movies from IMDB ([8]). For a lot of movies the available plots are very short and uninformative. Thus we restricted the dataset to the movies having plots of at least 200 words. This resulted in a set of 704 movies and plots and 4805 users. The plots are used to extract keywords.

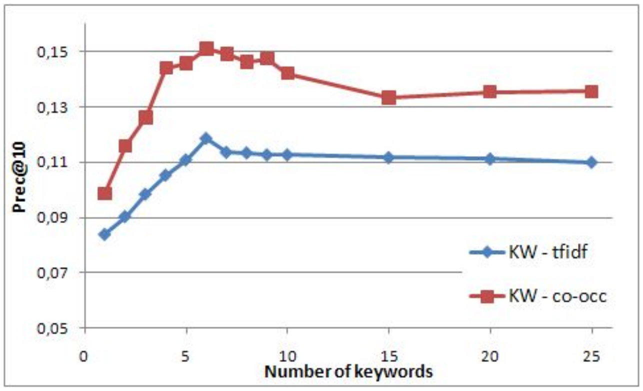

Figure 1: Influence of number of keywords per item on prec@10 of recommendation for the BBC data using the item based nearest neighbor algorithm

The set was split arbitrarily in a training and a test set, such that the training set contains 75% of the items and 25% of the items are in the test set. Again there is no overlap between items in both sets. The training set now contains 295 575 and the test set 66 386 ratings.

5.2 Evaluation

Our focus is on top-n recommendation. Two obvious evaluation measures are the precision at a given level and the area under the ROC curve (AUC). We will use the precision for top 10 recommendation (prec@10).

5.2.1 *Optimal number of keywords*

The first parameter we investigate is the number of extracted keywords. We vary the number of extracted keywords from 1 to 25 which means that for many texts we select almost all semantically interesting words from the synopses. Note that for a very large number of selected keywords the results of all selection methods will converge. Instead of simply selecting the keywords, we could also use the weights assigned by equation (3) or (12). In our experiments the use of weights did hardly influence the recommendation results.

In Figure 1 the effect of varying the number of keywords on content based recommendation is shown. The more advanced co-occurrence based keyword extraction algorithm gives significantly better results than the tf.idf based extraction. The second interesting observation is that in all cases the precision has a maximum for a relatively small number of keywords. The IMDB dataset gives rise to a similar picture. This means that for representing a text for recommendation it suffices to use a few words. Eventually, selecting the right keywords even can lead to an abstraction from irrelevant details, improving the recommendation quality.

It is also interesting to see how many different keywords are actually used when the optimal number of keywords is assigned to each document. For the BBC dataset a total number of 5949 different keywords is assigned by the tf.idf based keyword extraction and 2648 by the co-occurrence method. There are 67 different manually assigned keywords.

Table 1: AUC and precision of recommendation for the BBC dataset using different types of meta-data.

	Profile based		Nearest neighbors	
Data used	AUC	Prec@10	AUC	Prec@10
KW - man.	0,62	0,11	0,63	0,12
KW - tfidf	0,58	0,10	0,59	0,11
KW - co-occ.	0,67	**0,13**	**0,69**	**0,15**
genres	**0,68**	0,11	**0,69**	0,11
LDA	0,59	0,071	0,60	0,075

Table 2: AUC and precision of recommendation for the MovieLens/IMDB dataset using different types of meta-data.

	Profile based		Nearest neighbors	
Data used	AUC	Prec@10	AUC	Prec@10
KW - tfidf	0,59	0,18	0,60	0,18
KW - co-occ.	0,60	0,**21**	0,61	**0,21**
genres	**0,70**	0,20	**0,70**	0,19
LDA	0,61	0,14	0,61	0,14

For the IMDB plots the tf.idf keyword extraction comes up with 6651 keywords, whereas the co-occurrence based extraction assigns 4827 different keywords. We do not have manually assigned keywords, but there are also social tags assigned by users in the MovieLens dataset. For our subset there are 14 179 different tags.

5.2.2 *Comparison of meta-data*

Next we have compared the suitability of different type of meta-data for recommendation. We extracted a maximum of 8 keywords for each item for the BBC synopses and 20 keywords for the IMBD plots.

If we compare different types of keywords, the co-occurrence based keywords clearly give better recommendation results than the tf.idf based keywords and the manually assigned ones. Especially the latter fact is interesting, since manually assigned keywords are used as gold standard in most research on keyword extraction. Furthermore, for both datasets the genres are very effective for recommendation. In contrary, the generated topics from LDA fall short of expectations.

If we compare the two datasets, we see that the gap between the genre based recommendation and the best keyword based method is much larger in the MovieLens/IMDB dataset. The most likely reason for this difference is that for selecting movies, there are many aspects that are much more important than the topic of the plot. While this holds for a part of the BBC data as well, for a lot of items the topic of the broadcast might be more relevant.

For the BBC dataset we also tried out a number of combinations, like keywords and genres etc. However, none of the combinations yielded better results than the recommendations based on one type of meta-data.

6. CONCLUSION

Quality of recommendation using meta-data is extremely dependent on the quality of the provided meta-data. Also the quality of the extracted keywords depends on the quality and the length of the available synopses. Thus we should be very cautious to generalize the results. Nevertheless two important observations could be made that are relevant for the design of recommender systems: The number of words that is used to represent a text does not need to be very large and the method to rank words is an important factor influencing the quality of content based recommendation.

Another interesting aspect is that we in fact have used recommendation as a tool for the evaluation of keyword extraction. In most work on keyword extraction, manually assigned keywords are used as gold standard. This is always problematic, since manually assigned keywords are usually not the only possible good ones. Here we have an alternative evaluation method that even allows the automatically assigned keywords to be better than the manually assigned ones. In the dataset for which manually assigned keywords were available our proposed method indeed outperforms the provided keywords.

Acknowledgments

The research presented here was done within the MyMedia project funded by the European Community's Seventh Framework Program (FP7/2007-2011) under grant agreement N$^{\underline{o}}$ 215006. We thank the BBC for providing the dataset.

7. REFERENCES

[1] http://incubator.apache.org/uima/.

[2] http://nlp.stanford.edu/software/tagger.shtml.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] S. Debnath, N. Ganguly, and P. Mitra. Feature weighting in content based recommendation system using social network analysis. In J. Huai,et al., editors, *WWW*, pages 1041–1042. ACM, 2008.

[5] M. Fleischman and E. Hovy. Recommendations without user preferences: a natural language processing approach. In *Proceedings of the 8th int. conf. on Intelligent user interfaces*, p. 244. ACM, 2003.

[6] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[7] M. Hepple. Independence and commitment: Assumptions for rapid training and execution of rule-based pos taggers. In *ACL*, 2000.

[8] http://www.imdb.com.

[9] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[10] http://www.grouplens.org/system/files/-README_10M100K.html.

[11] M. J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev.*, 13(5-6):393–408, 1999.

[12] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The Adaptive Web: Methods and strategies of web personalization. Volume 4321 oF Lecture Notes in Computer Science*, pages 325–341. Springer-Verlag, 2007.

[13] C. Wartena and R. Brussee. Keyword extraction using word co-occurrence. In *DEXA Workshops*, Bilbao, Spain, 2010. Forthcoming.