Personalized News Recommendation based On Implicit Feedback

Ilija Ilievski*
University of "Ss. Cyril and Methodius"
Skopje, Macedonia
ilija139@gmail.com

Sujoy Roy,
Institute for Infocomm Research
A*STAR, Singapore
sujoy@i2r.a-star.edu.sg

ABSTRACT

This paper presents a personalized news recommendation system that combines effective ways of understanding new articles with novel ways of modelling evolving user interest profiles to deliver relevant news articles to a user. A news article is represented as a taxonomy of hierarchical abstractions that capture different semantic facets of the news story. A users interest profile is modelled as an evolving interest over these facets. Users interest in individual articles is determined using a novel **SWL** (*select-watch-leave*) interest modelling framework that leverages on a detailed analysis of his usage history. Initial performance comparisons with state-of-the art personalized ranking approaches[2] are promising.

General Terms

Recommendation, Rating Prediction

1. INTRODUCTION

Despite significant advancements made in the field of recommender systems, news recommendation is still the Holy grail for recommendation. The underlying algorithm driving most of the commercially popular recommender systems has been collaborative filtering[6].

While collaborative filtering works exceptionally well when the number of items and users' are fixed, it starts to fail when they are not. Especially, in the news domain where the life time of a news story is in general ephemeral and the number of stories and their content gets dynamically updated. This makes the problem of recommending relevant news articles extremely challenging. Moreover, a news recommender systems needs to cater to factors like freshness and dynamic popularity of the articles. Added to the above concerns is the reality check that the news needs to be personalized which requires understanding the users temporal consumption behavior and several other localized factors.

*On internship at the Institute for Infocomm Research

NRS 2013 October 13 2013, Kowloon, Hong Kong

http://dx.doi.org/10.1145/2516641.2516644.

Naturally, existing popular news recommender systems have taken a simplistic approach of aggregating/clustering category or publisher specific news articles and allowed users to manually adjust ones personal feeds to control how news articles are recommended. Of-late several works have looked to leverage on social media footprints to personalize news delivery. We believe more can be done; both in terms of understanding the user and the news articles in finding a good mapping between them.

In this paper we present a personalized news recommendation system that presents novel ways of understanding news articles and the user in delivering personalized news items. Our main contributions are as follows. (1) We represent each news article by a taxonomy of hierarchical semantic abstractions that capture different semantic facets of a news story. A collection of linguistic and statistical tools are used to represent an article as such a taxonomy. Note that while the taxonomy helps to describe a content, a users interest in news articles can be parameterized by the evolving weights a users watching behavior induces on each abstraction of the taxonomy. This means, we don't just know that the user was interested in an article, we also know what was he interested in and how much (hopefully!). (2) We propose a novel framework for estimating a users interest rating/preference for an article, named the **SWL** (*Select-Watch-Leave*) framework. Unlike collaborative ways of estimating interest that requires knowledge of the users interest in a large number of other articles or other users interests, our proposed **SWL** framework leverage's on effective knowledge mining of the users usage patterns and unsupervised feature learning. As the name suggests, the **SWL** framework investigates the temporal patterns that tell us (a) why a user *Selected* the article, what did he do while *Watching* the article and how and where did he *Leave* this current article to move on. In essence, a method formalizing the way to combine several contextual usage observations is presented. (3) Lastly a method for updating user profile and ranking of new articles in presented that shows promising performance based on state-of-the art performance measures.

In the next section we present an overview of the proposed news recommendation framework. Section 4 describes in detail key components of the framework that form the contribution of our work. Section 5 presents a descriptive analysis of the NRS challenge dataset, preliminary experimental results, analysis and performance comparisons with a state-of-the art personalized recommendation approach[2].

2. RELATED WORKS

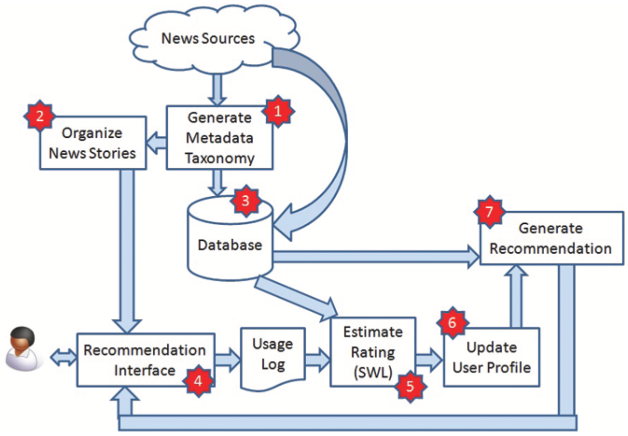


Figure 1: News Recommendation Framework

Works in recommender systems have been grouped into systems that perform either rating prediction or item recommendation given explicit feedback or based on implicit feedback. Rating prediction, given an explicit user-item rating matrix has been extensively studied in the literature[6]. Herein rating for an article by a user is estimated based on knowledge of his and other users ratings for items. However predicting a rating for an item based on implicit feedback is not so well studied. Hu et. al. [5] propose a collaborative filtering approach for rating prediction from implicit feedback in the TV shows recommender domain. They propose the need for transferring raw observations, specifically a measure of the users involvement in the item, into interpretations of preference and confidence level and then estimate the users rating/preference for the item. Their actual formulation points to the fact that if a users involvement in an item is high, his rating/preference for the item is high, whereas if the involvement is low, rating/preference is also low. Since preference and confidence are function of the users involvement in the item it does not really consider the users contextual indications in the estimation process. We believe that if we consider the contextual information, we hope to see clear deviations from their assumption. For example, in the context of news, if a user consumes different articles on the same topic it is a clear indication of the users interest in the topic although his involvement in each article need not be very high. In fact, in the news domain, the users actual involvement in the item is also not very clear. Contextual and content information can help in a realistic implicit feedback scenario.

Rendle et. al. [10] propose an item recommendation approach based on implicit feedback that optimizes an items ranking criterion. Their work does not require estimating a users preference rating for an individual item that the user has seen before. Since item consumption depends on how items are recommended this work is very relevant. But this requires making some not so reasonable assumptions regarding the users preference model and it does not consider the contextual information. It is based on the binary knowledge that an item was consumed by the user.

Several news recommender systems[3, 7, 8] have also been proposed in the literature that recognize the limitations of applying existing collaborative filtering based recommender systems to the news domain. Social statistics based on user click information[9], popularity metrics have been widely used to compensate for the lack of availability of explicit feedback. Garcin et. all [3] propose a news recommender system based on context trees built based on the users temporal browsing behavior and content information. Li et. al. [7] model the news recommendation problem as a contextual bandit problem and use contextual information about articles and users to maximize user clicks. We wish to highlight that most of these complex algorithms need to be evaluated in a LIVE environment where limited resources and time efficiency are of the essence, especially in the domain of news recommendation.

Our proposed approach also highlights the need for using contextual information and formalizes an intuitive way of combining contextual information from implicit feedback to model the users profile.

3. FRAMEWORK OVERVIEW

An illustrative workflow of our proposed News Recommender Framework is depicted in Figure 1. The framework works on the basis of better understanding of the user and articles. The workflow follows the numbering on the blocks and is described below. For the purpose of clarity in this section we only go over the workflow overview. The details of some of individual components are covered in section 4.

(1) Generate Metadata Taxonomy. News articles are aggregated from multiple news sources and stored in our database (**3**). The nature of these articles is free text and does not contain any structured information about the story. Given a free text news article we generate structured metadata about the news story in the form of a taxonomy of hierarchical abstractions that capture different semantic facets of the news story. Note that the basic taxonomy is hand crafted based on expert domain knowledge and a collection of linguistic and statistical knowledge extraction tools is used to extract structured information from it. The structured metadata about the articles is also stored in the database (**3**).

(2) Organize News Stories. The news articles are next organized based on the different abstractions of the taxonomy to form clusters of related articles.

(3) Storage. The raw news articles and their metadata are stored in a database in an organized way to facilitate fast search and delivery.

(4) Recommender Interface. Contents are recommended to the user over the recommendation interface. We note that how we recommend content is as important as what we recommend. Hence the interface which forms the portal for news consumption plays a very important role in the recommendation performance. In this work our focus is not on how we design the interface but rather on what we recommend. Hence in our present implementation the interface presents news articles that have been either organized/clustered as per their story facets or have been deemed to be recommendable by our recommendation system. For first time users, the former mode of presentation applies.

(5) Estimate Article Rating. The recommender interface collects the users usage footprint in logs to understand the users interest profile. The usage log is first filtered to identify the articles the user clicked on. Next, the context of the users selection of an article is extracted as a feature vector based on implicit evidences in the log trail. This feature vector is used to learn the users interest rating for a content using a novel **SWL** approach. Details of the **SWL** approach is described in section 4.

(6) Update User Profile. The knowledge of the interest rating for an article and the contextual feature descriptor is used to estimate the influence of the article selection on the different abstractions of the articles taxonomy. For the sequence of watched contents in the usage log, the user profile is updated. In this way the users interest profile is generated and updated progressively following the users selection trail. Details of the user interest profile update process is described in section 4.

(7) Generate Recommendation. Given an updated user profile represented as a taxonomy of abstractions describing the user and his interest in news articles, we predict the interest on a future article which has also been represented by the same taxonomy. The details of described in Section 4. The ranked list of recommendable articles are presented and delivered to the Recommender Interface.

Since the recommendation process is clearly dynamic in nature and not designed for a fixed set of articles we believe that the recommendations improve as we understand more about the user.

4. METHOD DETAILS

In this section we present the mathematical formulations and details algorithms driving the proposed news recommendation workflow.

4.1 Metadata Taxonomy

A hand crafted hierarchical taxonomy $\mathcal{U}$ of m abstractions $\mathcal{U} = \{U_1, \ldots, U_m\}$ that describes an article and a user is designed. A subset of these abstractions are article specific whereas a subset of these abstractions are relevant for the user. The values for the user specific abstractions are determined from the log whereas the values for the article specific abstractions are generated by analyzing the article. Each abstraction is described by a set attributes $U_i = \{U_{ij} : \forall j\}$.

The taxonomy abstractions are {Genre: {Politics, Finance, Entertainment, Sports, Weather, Technology, Lifestyle, Medicine, General, Social}, Location: {local (German, Austrian, Swiss), international}, Entities: *variable length set*, Keywords: *variable length set*, Popularity: *categorical*, Publisher:*fixed entities*, Total Time to Read: *categorical*, Gender: {Male, Female}, Age: *categorical*, Income: *categorical*, User Activity Index: *categorical*, Publisher preference: *categorical*, Freshness: *categorical* }.

We use state-of-the art statistical and linguistic tools namely, entity recognition, topic detection, word representation based clustering and classification to identify relevant attributes and their relevance scores for an article [1].

[1]Due lack of time and to take advantage of state-of-the art word embedding results [1] we also translate the articles from German to English using Bing translator for clustering/classification. Translation was performed on the word vocabulary instead of the full articles themselves due to quota limitations on free translation. We note that this can lead to loss of contextual information in the word representation

Table 1: Semantic Contextual Features: Encodes the reasons that contribute towards estimating the probability for Select-Watch-Leave

No.	Features	Remarks
1.	weekday(S/L)	Day of the week as categorical information
2.	hour(S/L)	hour of the day as 3 categories (morning, day, night)
3.	keywords(S/L)	encodes the topic of the article
4.	itemAge(S/L)	0:<day, 1:<week, 2:<month, 3:>month
5.	explainVisitProb(S)	0: No. of user-pub impressions, 1:reading some article, 2:reading his favourite genre, 3: first visit, 4: No. deviceType-user-pub impressions, 5: item is click not impression
5.	explainClickProb(S)	0: best score compared to others, 1: natural click flow, 2: mistake
6.	timeSpent(W)	Time Spent in reading article. -1 if it is > 4 hrs
7.	wordCount(W)	size of the article
8.	whereNext(L)	-1: unknown, 0: clicked on recommended article, 1: same publisher and same genre, 2: same publisher different genre, 3: different publisher same genre, 4: different publisher different genre, 5: same item, 6: stopped reading
9.	timeSpentWithPub(W)	Percentage over all publishers
10.	repeat(S/L)	read count of article

4.2 Estimating Rating (SWL)

The values for the user specific abstractions are obtained by mining the usage logs. Usage logs carry a footprint of the users usage trail which includes information like what article the user clicked/read, how long he read, what was the source of the article and several other indicators that can be seen as implicit feedback for estimating the users interest in the articles he reads. Most real usage logs do not contain explicit ratings from users indicating how much they like articles. However there is an abundance of implicit feedback.

We propose a novel (SWL) framework for leveraging on the contextual implicit feedback left by the user to estimate his interest in an article. Note that there is no explicit ratings to learn a model for estimating users interest in a supervised way hence we can only estimate a binary preference measure.

Corresponding to every article the user has clicked on, first we generate a d dimensional contextual feature vector that encodes all the discriminative information regarding how and why the user *selected* an article for reading, what did he do while reading/*watching* and why the user *left* the article and moved on; all of which can give some indication of the users interest in the article. The contextual feature vector is based on the reason clues given in Table 1. Since d is very large (as we intend to encode as much discriminative information as possible) the feature vector is clearly sparse in nature. Each dimension encodes the possible reasons for select, watch and leave. The select and leave reasons are encoded in a complimentary way. For example, if the probability of selecting a sports article is 0.3 estimated based on the contextual window of the users log, the probability of leaving is 0.7.

We propose a novel way of encoding semantic features from implicit feedback called Select-Watch-Leave framework. The intuition behind it is explained by the following. If the user selects to view a sports article and moves on to view another sports article on the same topic that should indicate that the user is interested in the topic and the news genre. On the other hand, if the user moves on to another article from a different genre we cannot really say that the user is very interested in the topic or the genre. Similarly if the user spent exceptionally more time on the article relative to the size of the article (measured by word count) or repeatedly read the article, it could be an indication of interest.

First a distributive low dimensional embedding of the implicit feedback feature vector that tells the probability of a user selecting, watching and leaving an article is generated. A constrained auto-encoder is used to map the d-dimensional feature vector to a 3-dimensional vector. The auto-encoder[4] learns the weights for mapping a feature vector to a 3-dimensional select-watch-leave vector. Note that the actions are correlated and hence a distributive representation is learnt. For example, if the user moves on from a sports article to another sports article and that too on the same topic, the network should learn the select probability to be high and the leave probability to be low. Figure 2(a) depicts an impersonation of our auto-encoder network.

Once we obtain a semantic mapping, we estimate the interest of a user in the article by learning a probabilistic graphical interest prediction model of select-watch-leave and interest nodes (refer Figure 2(b) given by the conditional probability $P(I|S,W,L)$. Given a new article that a user has selected, we can extract the contextual feature vector, map it to a select-watch-leave feature representation and make an inference on the interest model to estimate the users interest rating r.

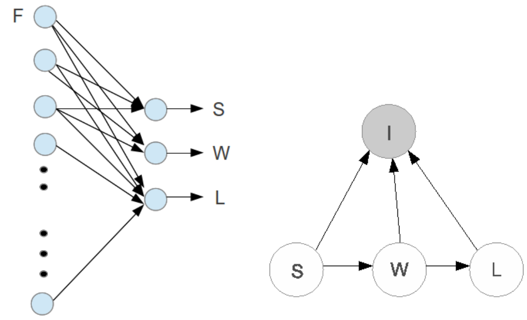
Note that by learning user specific contextual features, the rating prediction model can encode local personalized information. We can also include social contextual features to encode global statistics in the model.

4.3 Updating User Profile

The users estimated interest rating for an article r and the contextual feature representation is used to estimate the users preference over abstractions in the user profile taxonomy ($\mathcal{U}$). The update of preference weight p_i for an abstraction U_i (e.g. Genre, Popularity etc) is given by

$$\hat{p}_i = p_i * (1 + r/D_1(U_i^{(t+1)}, U_i^{(t)})), \tag{1}$$

where p_i is the preference weight for abstraction U_i in



(a) (b)

Figure 2: Rating Prediction: (a) SWL Embedding using auto-encoder (b) SWLI Model for estimating rating based on $P(I|S,W,L)$

the user model U at instance t and $\hat{p}_i$ is the preference at instance $t+1$. The function

$$D_1(U_i^{(t+1)}, U_i^{(t)}) = J(U_i^{(t+1)}, U_i^{(t)}) + 1/m, \tag{2}$$

where $J(U_i^{(t+1)}, U_i^{(t)})$ is the Jaccard index between abstractions U_i at time instance t and $t+1$ and m is the number of abstractions (to avoid division by zero), captures the degree of preference for the abstractions in the user profile. Note that $D_1(\cdot)$ is a similarity measure hence small $D_1(U_i^{(t+1)}, U_i^{(t)})$ leads to increased preference for the abstraction U_i. This gives an estimate of why the user was interested in the article.

Once a preference weight on the abstractions is determined, the weight on the attributes under an abstraction $p_{U_{ij}}$ are updated by the following update rule,

$$\hat{p}_{ij} = \begin{cases} p_{ij} + (\hat{p}_i/|U_i|)\exp^{|U_{ij}|} & \forall U_{ij} \in U_i^t \\ \hat{p}_i/|U_i^{(n)}| & \{U_i^{(n)} : U_{ij} \in U_i^{t+1} - U_i^t\} \end{cases} \tag{3}$$

where $D_2(U_{ij}^{(t+1)}, U_{ij}^{(t)})$ is the absolute difference in weight between two attributes, $|U_i^{(n)}|$ refers to the number of new attributes, $|U_{ij}|$ is the count of the attribute in the user profile and $|U_i|$ is the number of attributes in abstraction U_i. In this way for every article read by the user his profile is updated.

4.4 Generating Recommendations

Given a user model U, for a new article (say modeled as N) a score $\mathcal{S}$ is estimated by the following

$$\mathcal{S} = \sum_{\forall U_i \in N} \hat{p}_i \, (\frac{1}{|U_i|} \sum_{\forall U_{ij} \in N} \hat{p}_{ij}), \tag{4}$$

where U_{ij} refers to the attributes that are common to both the user profile and the article. The interest scores are ranked and the top-N articles are recommended to the user. We would like to point out that this scoring approach would not recommend articles based on attributes the user has never seen before and hence it may be argued that the recommendation will not be diverse enough. Note that our framework includes attributes that are keywords that are matched based on word representations. This helps to build up related attributes as part of the user profile. We also include popularity statistics with the hope of adding informed diversity in our recommendations. Hence the recommendation incorporates both global and local user specific information while being efficient in building the ranked list of recommendations. We tested several other ranking function like KL-divergence (matching the probability distributions over attributes) etc., but the above function gives best results and is efficient as well.

5. EXPERIMENTS

5.1 Data Set

The whole dataset (72 GB) consists of news articles and user logs collected between 1 to 30 June from 15 different news article publishers. The publishers are based in Germany, Austria and Switzerland and all articles are in German. Three types of data are provided.

News article items. Data contains create and update events on items. Total count of unique items is 758,929 and total number of events is 5,051,543. For each event some of the following properties are provided: (1) timestamps for created_at, published_at, updated_at, (2) urls to article's text and image and (3) short text and title.

Read events. These are events generated when a user reads an article. Total number of such events is 84,187,577 and number of unique users is 1M.

Click events. These are generated when a user clicks on a recommended article. Total number of such events is 1,053,709 and number of unique users is 569,054.

For each event type the following properties are available about the user: income, gender and age given as probabilities over different groups. Browser, ISP, OS, User location, language and device type are given as IDs for which we weren't provided with their mappings. But we did group them found distinct values, namely Browser (9), ISP (15), OS (5), User Location (17), Language (2), Device type (5). Not all properties are present for each event.

Some of the events didn't contain the ID that corresponds to the actual news article item. Some did contain ID but to an article item that was created before 1st June, so we didn't have more information about the item, only its ID. Many events contained IDs to items that weren't news articles, but forum posts from user's asking some advice or similar website articles. Hence the usage data was in no way clean. We cleaned the data to some extent and estimated some of the categorical information described before to generate our article taxonomy and user profile.

For performance evaluation our system was trained for 16 users, 2093 items, 2725 events and tested on 14 users, 985 items and 1145 events. Due to limitations of time and resources we only choose top 18 active users in our evaluation. Evaluation on more number of users is a subject of future work.

5.2 Experimental Analysis

Herein we present results and some analysis of experiments we conducted towards (1) evaluating performance of estimated preference for articles based on implicit feedback observations and (2) comparing our recommendation results with a state-of-the art personalized ranking approach[2].

We evaluate our preference prediction results based on performance measures proposed in [5]. Our system generates an ordered list of shows from the one predicted to be most preferred till the least preferred one. We use the same recall oriented ranking measure $rank_{ui}$ that Hu et. al.[5] uses to evaluate rating performance. $rank_{ui}$ gives a percentile-ranking of article i within the ordered list of all articles prepared for user u. That is, $rank_{ui} = 0\%$ would mean that article i is predicted to be the most desirable for user u, thus preceding all other programs in the list. On the other hand, $rank_{ui} = 100\%$ indicates that article i is predicted to be the least preferred for user u, thus placed at the end of the list. Hence, for an observation, the basic quality measure is the expected percentile ranking of a watching unit in the test period, which is:

$$\overline{rank} = \frac{\sum_{u,i} o^t_{u,i}\ rank_{ui}}{\sum_{u,i} o^t_{u,i}} \tag{5}$$

where o^t_{ui} is an observation made about the usage article i from the test set, by user u. Note that in our proposed SWL framework this observation can refer to any observation made in the context of the usage of article i. Lower values of $\overline{rank}$ are more desirable, as they indicate ranking actually watched shows closer to the top of the recommendation lists. For random predictions, the expected value of $rank_{ui}$ is 50% (placing i in the middle of the sorted list). Thus, $\overline{rank} \geq 50\%$ indicates an algorithm no better than random.

Table 2 presents values of $\overline{rank}$ under different contextual usage observations vs. ranking schemes. Note that unlike the TV program recommendation scenario as analyzed in [5], in the news recommendation scenario the users actual engagement on the article is not known, as there is no known fixed duration to read the article. Although we do know the amount of time the user has spent in reading the article, different users will spend different amount of time on the same article. Hence it is important to leverage from the knowledge of other contextual usage observations. Row 1 in Table 2 analyzes the effectiveness of these observations individually and independently compared with our proposed approach that tries to combine the information from them. The ranking of articles in this case is based on our systems output. The superior performance of our SWL framework in combining several contextual usage observations is demonstrated by the lowest $\overline{rank}$ value in the first row of Table 2.

We also analyzed the effect of ranking articles simply based on the contextual usage observations. Note that this is of academic interest only as in a LIVE recommendation situation for new articles these usage observations are not available. What we would have is a user model that needs to be somehow used to rank the new articles. Nevertheless, we make some interesting and surprising observations that are marked in bold in Table 2. The usage observations considered are explained in Table 3. While we use these observations as a weighting factor the articles are also ranked based on these observations. For ranking, SWL stands for the ranking estimated by our proposed framework and SWL observation refers to the estimated score. Note that ranking the articles based on their observations gives the best value

Table 2: Illustration of $\overline{rank}$ values under different contextual usage observations and ranking schemes.

Ranking \ Usage Obs	SWL score	timeSpent	genreCount	sameItemCount	pubCount	itemSize	itemAgeCat	itemAge
SWL	**20.22**	46.33	46.95	53.23	47.10	48.47	48.50	41.26
timeSpent	46.79	**13.44**	50.57	40.39	50.39	48.32	51.21	41.26
genreCount	40.73	52.14	**31.19**	43.75	32.98	49.77	45.80	47.19
itemSize	45.57	46.67	49.66	54.81	47.50	**35.96**	43.30	49.57

Table 3: Explanation of Usage Observations used to evaluate $\overline{rank}$

Observation	Explanation
SWL	Score generated by the user profile for the article
timeSpent	how much time the user has spent reading the article, not scaled
genreCount	how many times the user read the articles from this genre, not scaled
sameItenCount	how many times the user has read this same article
publisherCount	how many times the user read articles from this publisher
itemSize	how many words are there in the article
itemAgeCat	Age of the article grouped in 4 categories, value from 1 to 4
itemAge	actual time difference between the article and the event

Table 4: Comparison of MAP scores with [2] for top 5 users (columns). kNN attributes use the same attributes as our proposed framework

[2]-kNNAttrib	1	0.62	0.25	0.13	0.08
[2]-BPRMF	0	0	0.2306	0.0311	0
proposed	0.71	0.7	0.26	0.2	0.2

of $\overline{rank}$ when the weighting value is the same observation. While it may seem obvious it is not really so. We note that the value of $\overline{rank}$ when the weighting value is the actual rank of the article is clearly random (value = 66.67). The value of $\overline{rank}$ clearly depends on the actual values of the usage observations. Hence, which contextual usage observation is used and how they are used affects the recommendation performance. This justifies the need for investigating ways for combining the input given by several observations under an implicit feedback scenario into the rating prediction process.

We also compare MAP scores for recommendation performance with the algorithm presented in [2] for the same implicit attributes (refer Table 4). The scores presented are for five most active users, when test case is consider as correct if only one of the four recommend articles is relevant. We note that a specific attribute based approach gives better performance than a method agnostic to attribute which also is consistent with our approach that models the user based on his evolving interest over content attributes. Hence it is evident that specific content understanding is important for news recommendation.

6. CONCLUSION

News recommendation is an extremely challenging area particularly when given with the task of understanding actual noisy usage logs with no explicit feedback from the users. The dynamic nature of the domain makes the process more difficult to evaluate. We believe that the real merit of our proposed framework will have to be tested on live evaluation which is yet to be seen.

7. REFERENCES

[1] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[2] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. MyMediaLite: A free recommender system library. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011)*, 2011.

[3] F. Garcin, C. Dimitrakakis, and B. Faltings. Personalized news recommendation with context trees. *CoRR*, abs/1303.0665, 2013.

[4] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[5] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 263–272, Washington, DC, USA, 2008. IEEE Computer Society.

[6] Y. Koren and R. M. Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 145–186. 2011.

[7] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 661–670, New York, NY, USA, 2010. ACM.

[8] L. Li and T. Li. News recommendation via hypergraph learning: encapsulation of user behavior and news content. In *Proceedings of the sixth ACM international conference on Web search and data mining*, WSDM '13, pages 305–314, New York, NY, USA, 2013. ACM.

[9] J. Liu, P. Dolan, and E. R. Pedersen. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*, IUI '10, pages 31–40, New York, NY, USA, 2010. ACM.

[10] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the*

Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.