Combining Content with User Preferences for TED Lecture Recommendation

Nikolaos Pappas, Andrei Popescu-Belis
Idiap Research Institute, Rue Marconi 19, 1920 Martigny, Switzerland
{nikolaos.pappas, andrei.popescu-belis}@idiap.ch

***Abstract*—This paper introduces a new dataset and compares several methods for the recommendation of non-fiction audiovisual material, namely lectures from the TED website. The TED dataset contains 1,149 talks and 69,023 profiles of users, who have made more than 100,000 ratings and 200,000 comments. This data set, which we make public, can be used for training and testing of generic and personalized recommendation tasks. We define content-based, collaborative, and combined recommendation methods for TED lectures and use cross-validation to select the best parameters of keyword-based (TFIDF) and semantic vector space-based methods (LSI, LDA, RP, and ESA). We compare these methods on a personalized recommendation task in two settings, a cold-start and a non-cold-start one. In the former, semantic-based vector spaces perform better than keyword-based ones. In the latter, where collaborative information can be exploited, content-based methods are outperformed by collaborative filtering ones, but the proposed combined method shows acceptable performances, and can be used in both settings.**

I. Introduction

The recommendation of multimedia content to users can leverage either the content descriptors (*content-based methods, CB*) or information from the preferences of similar users (*collaborative filtering, CF*) or both (*hybrid systems*). While in some domains, such as movie recommendation, content descriptors and user ratings can be available on a large scale, such as the Movielens data with millions of ratings (from www.grouplens.org), in other domains these can be scarce.

In this paper, we compare recommendation techniques for scientific lectures or courses, that is, non-fiction audiovisual material with informative purposes, the content of which plays a significant part in deciding what to recommend. We compare the merits of CB and CF methods and propose a new method for combining semantic features (Section V) based on distances in semantic vector spaces (Section IV), with user preferences, when they are available. Following appropriate training to identify the best performing features (Section VI), we show that CB recommendation using Explicit Semantic Analysis [1] is the best performing method in a cold-start setting, when no user preferences are known, including the case of anonymous viewers (Section VII-A). In a non-cold-start setting (Section VII-B), pure CF methods perform best, but only slightly above the combined CB and CF method with keyword-based distance, showing the importance of using content features in both settings.

The methods are tested on a new public dataset acquired from the TED web-based repository of lectures on social and scientific topics (www.ted.com, see Section II). We show how this dataset can be used for the evaluation of lecture recommendations (Section III), given its rich content and metadata (to be used as features) along with explicit feedback from users (to be used as ground truth for training and testing). Our results thus constitute the first benchmark scores on this promising data set.

II. The TED Collection: A Dataset for Recommendation Evaluation

The TED website is the online repository of audiovisual recordings of the popular TED lectures given by prominent speakers. The recordings and the metadata accompanying them are made available under a Creative Commons non-commercial license (see www.ted.com). The website provides extended metadata as well as user-contributed material such as discussion threads related to the talks. The TED speakers are scientists, writers, journalists, artists, and businesspeople from all over the world who are generally given a maximum of 18 minutes to present their ideas. The talks are given in English and are usually transcribed and then translated into several other languages by volunteer users. The quality of the talks has made TED one of the most popular online lecture repositories, as each talk was viewed on average almost 500,000 times, as we found based on the collected metadata – with large variations across talks, of course. An important characteristic of TED is that the metadata for the audiovisual content is human-made.

A. Metadata Structure and Statistics

We crawled the TED dataset in April 2012 and consists of two main entry types: talks and users. The talks have the following data fields: identifier, title, description, speaker name, TED event at which they were given, transcript, publication date, filming date, number of views. Each talk has a variable number of user comments, organized in threads. In addition, three fields were assigned by TED editorial staff: related tags, related themes, and pointers to related talks (generally three per talk). For 95% of the talks, a high-quality manual transcript is available. Table I provides the main statistics of the dataset, which includes 1,149 talks from 960 speakers.

Users have an identifier and a list of talks marked as favorites. There are 69,023 registered users, but only 10,962 of them (i.e. 14%) have explicitly indicated one or more favorite talks, and we will refer to them as *active users*. Only the profiles of active users can be used for testing a recommender system, by comparing its output with known favorites (not shown to the system). In this paper, we will only use the subset of 2,427 users who have made 12 or more ratings each – a

	Total	Per Talk		Per User		Per Active User	
Attribute	Count	Average	Std	Average	Std	Average	Std
Talks	1,149	-	-	-	-	-	-
Speakers	961	-	-	-	-	-	-
Users	69,023	-	-	-	-	-	-
Active Users	10,962	-	-	-	-	-	-
Tags	300	5.83	2.11	-	-	-	-
Themes	48	2.88	1.06	-	-	-	-
Related Videos	3,002	2.62	0.74	-	-	-	-
Transcripts	1,102	0.95	0.19	-	-	-	-
Favorites	108,476	94.82	114.54	1.57	8.94	9.89	20.52
Comments	201,934	176.36	383.87	2.92	16.06	4.87	23.42

TABLE I. STATISTICS FOR THE TED DATASET: WE REPORT TOTAL COUNTS AND AVERAGES WITH STANDARD DEVIATIONS (‘STD’) PER TALK AND PER USER, FOR EACH OF THE ATTRIBUTES. THE USERS THAT HAVE INDICATED AT LEAST ONE FAVORITE TALK ARE CALLED ‘ACTIVE USERS’.

balance between having enough ratings per user and enough users in the subset. Overall, the users have expressed more than 100,000 indications of favorite talks, and made more than 200,000 comments. We make available this dataset[1] under the same Creative Commons non-commercial license as the TED talks.

B. Ground Truth

The explicit user preferences in a given dataset constitute the ground truth which can be used for training and evaluating recommendation algorithms for *personalized recommendations*. A common form of such preferences are numeric ratings (e.g. from 1 to 5) that are given by users to items. In the TED dataset, the fact that a user has listed a talk among her favorite talks will count as the explicit preference. This corresponds to a binary numeric rating, ‘1’ for a favorite talk, and ‘0’ for a talk not included in the list of favorites, which can mean two things: either the talk was not seen, or it was not liked. (The data set does not provide viewing information.) Such problems are called one-class collaborative filtering problems [2], and are particularly challenging due to the fundamental uncertainty of the ‘0’ class. User comments constitute a more implicit form of expressing preferences, which we exploited in another study to augment rating information [3].

The three related talks (or “what to see next”) recommended by TED editors for each talk can be used as ground truth for a *user-independent recommendation task*, the goal of which is to determine for each talk the “most similar” ones, i.e. the ones most likely to be of interest to an anonymous user who has just viewed the talk. Similarly to the long-tail distribution of rated items in many previous datasets [4], here most of the ratings are also condensed over a small fraction of the most popular items: for instance, 23% of the ratings cover the top 5% of the items.

C. Comparison with other Collections

The aforementioned properties of the TED data cannot be easily found in other alternative lecture repositories such as Khan Academy[2], VideoLectures.NET[3], YouTube EDU[4], or Dailymotion[5]. Khan Academy is an online learning community that contains more than 3,200 videos on scholarly topics. It shares some properties with TED in terms of providing transcripts and commenting capabilities, but it lacks descriptive fields, tag annotation and explicit feedback. Similarly, VideoLectures.NET, Youtube EDU or Dailymotion do not provide transcripts in text form and do not provide all the TED metadata fields. The dataset provided for the VideoLectures.NET recommender system challenge [5] includes the viewing history of the lectures as a ground truth for predicting future views of each lecture, along with content-related features, author and event information. However, information that is particularly useful for recommendation tasks such as explicit user feedback and detailed content information such as lecture transcripts is not made available.

TED thus appears as particularly valuable since it provides ground truth from explicit user preferences along with human-made recommendations, which are critical for evaluating, respectively, personalized and generic recommendation tasks.

III. DEFINITION OF RECOMMENDATION TASKS

In this section, we specify the two recommendation tasks that can be evaluated using the TED dataset, namely the personalized and the generic one. In this paper, we will then focus on the first one.

A. Personalized Recommendations

Given a set of binary ratings as a ground truth, the goal of the personalized recommendation task is to predict whether items will be interesting or not for the users [6], or more simply to predict the N most interesting ones (top-N recommendation task [7]). In such a scenario of offline prediction, the recommendation models are classically trained on fragments of user’s histories, and evaluated by hiding some of the preferred user items and then trying to predict them. The performance is evaluated using classification accuracy metrics.

For the TED dataset, we suggest that for each user u in a set U, her ratings (favorites) are randomly split into training and test sets, namely M and T, typically 80% vs. 20%. A recommendation model is trained (possibly with cross validation) on M, and then tested on the held-out set T by comparing its output with the recommended items R for each user u. The set of users U can be selected based on various constraints, such as a minimal number of ratings available.

B. Generic Recommendations

The generic or user-independent recommendation task corresponds to scenarios in which the users’ history of ratings is

[1] https://www.idiap.ch/dataset/ted/

[2] http://www.khanacademy.org/

[3] http://www.videolectures.net/

[4] http://www.youtube.com/education/

[5] http://www.dailymotion.com/

absent, e.g. for anonymous users. The goal of this task is to predict the most similar items to a given one, which can also be seen as a non-personalized top-N recommendation task. Given a set of human-made, user-independent recommendations for each item in a dataset (e.g. "related videos" in TED), a model can be trained and evaluated using this information as ground truth. The evaluation can be done by splitting the set of items I into a training M and a testing T set.

C. Evaluation Metrics

For the top-N personalized recommendation task, error metrics such as RMSE are not the most appropriate ones, since a top-N recommender is not necessarily able to infer the rating of a user $u \in U$ for an item $i \in I$ [7]. Instead, this task can be evaluated more informatively by using the classification accuracy metrics of precision, recall and f-measure (see [6]). For instance, precision at N is given by:

$$P(N) = \frac{1}{|U|} \sum_{u \in U} \frac{|T_u \cap R_{u@N}|}{N} \tag{1}$$

where N is the bound of top recommendations, $|U|$ is the total number of users in U, T_u is the set of items in user's u history and $R_{u@N}$ are the top-N recommendations of the model for the user u. Recall is computed by dividing by the number of items in user's u history, $|T_u|$, instead of N. The harmonic mean $F(N)$ of $P(N)$ and $R(N)$ can also be computed. Similarly, applying Eq. 1 directly to items I in a test set T, we obtain the accuracy metrics for generic recommendations.

IV. Semantic Vector Space Models

Content-based recommender systems use similarities between items that rely on their content descriptors. Here, we investigate semantic vector space models (VSM) to define such similarities, and later in Section 4 we compare their merits for cold-start recommendation over the TED dataset. This benchmarking is a contribution to ongoing debates on semantic-based recommendation approaches [8]. Semantic VSMs are considered to be able to reduce the effect of the curse of dimensionality, data sparseness and other problems of standard VSMs (such as those based on TF-IDF weighting [9]). The proximity of two vectors in the semantic space (usually computed with cosine similarity) can be interpreted as a semantic relatedness between the objects that are represented by those vectors, which can be used to model user preferences in recommendations tasks.

When using a VSM, each document d_j is represented as a feature vector $d_j = (w_1, w_2, ..., w_{ij})$, where each position i corresponds to a word of the vocabulary. The weights w_{ij} can be computed using various models: Boolean values, counts, term frequencies, inverse document frequencies, or TF-IDF coefficients. TED talks can thus be represented by creating vectors of words from their text attributes, which can be pre-processed to remove stop words or to apply stemming.

There are several techniques in the literature for creating semantic representations in VSMs. In our experiments, we consider a baseline VSM with TF-IDF as the state-of-the-art weighing model [10] and four representative semantic VSMs from the three main existing categories, as follows: (1) as dimensionality reduction methods we use Latent Semantic Indexing (LSI) [11] and Random Projections (RP) [12]; (2) as a topic modeling approach we use Latent Dirichlet Allocation (LDA) [13]; and (3) as a concept space based on external knowledge we use Explicit Semantic Analysis (ESA) [1]. These techniques have generalization capabilities, as they project the data from the original vector space to a topic or concept space with a reduced number of dimensions – apart from ESA which actually augments the dimensionality to the number of Wikipedia concepts. In terms of free parameters, LSI, RP and LDA rely on the number of topics t (latent factors). Moreover, LDA relies on two parameters traditionally noted α and β for the Dirichlet prior of topic and word distributions.

For the implementation of LSI, RP and LDA, we used the Python Gensim library [14], while for ESA we used the Wikipre-ESA Python implementation of the method described in [1], over a 2005 snapshot of Wikipedia.

V. Recommendation Algorithms

We benchmark on the TED data two main types of recommendation methods, namely content-based and collaborative filtering ones, focusing on item-based similarities [15]. For the first type of methods, we pre-compute an item similarity matrix for each of the VSMs above, noted respectively S_{TF-IDF}, S_{LSI}, S_{RP}, S_{LDA} and S_{ESA}. Each S is an $m \times m$ matrix, m being the number of talks, and the value of each element s_{ij} is the cosine similarity of the vectors representing items i and j in the given model. For the second type, we pre-compute the item similarity matrices based on the common ratings between pairs of items in the user-item matrix (built from the training set) by using two common metrics, namely Pearson correlation S_{PC} (e.g. [16]) and cosine similarity S_{COS} (e.g. [7]).

A. Content-based Algorithms

We define a ranker based on content similarities, noted as CB. Given a similarity function that outputs a score for two items (talks), CB recommends to a user u a list of ranked items based on the k most similar items to those already known to be her favorites, i.e. to the training data M_u. Therefore, CB recommends items based on their estimated relevance $\hat{r}_{ui}$:

$$\hat{r}_{ui} = \sum_{j \in D^k(u;i)} s_{ij} \tag{2}$$

where $D^k(u;i)$ are the k most similar items to the ones in the training set of the user M_u and s_{ij} is the similarity between items i and j according to one of the five matrices S.

B. Collaborative Filtering Algorithms

Neighborhood models as in Eq. 3 are commonly used for collaborative filtering. The prediction function $\hat{r}_{ui}$ estimates the rating of a user u for an unseen item i, based on the bias estimate b_{ui} of u for item i, given in Eq. 4, and on a score that is calculated from the the k most similar items to i (according to either S_{PC} or S_{COS}) which the user u has already rated, i.e. the neighborhood $D^k(u;i)$. The denominator guarantees that the predicted ratings will fall in the same range of values as the known ones.

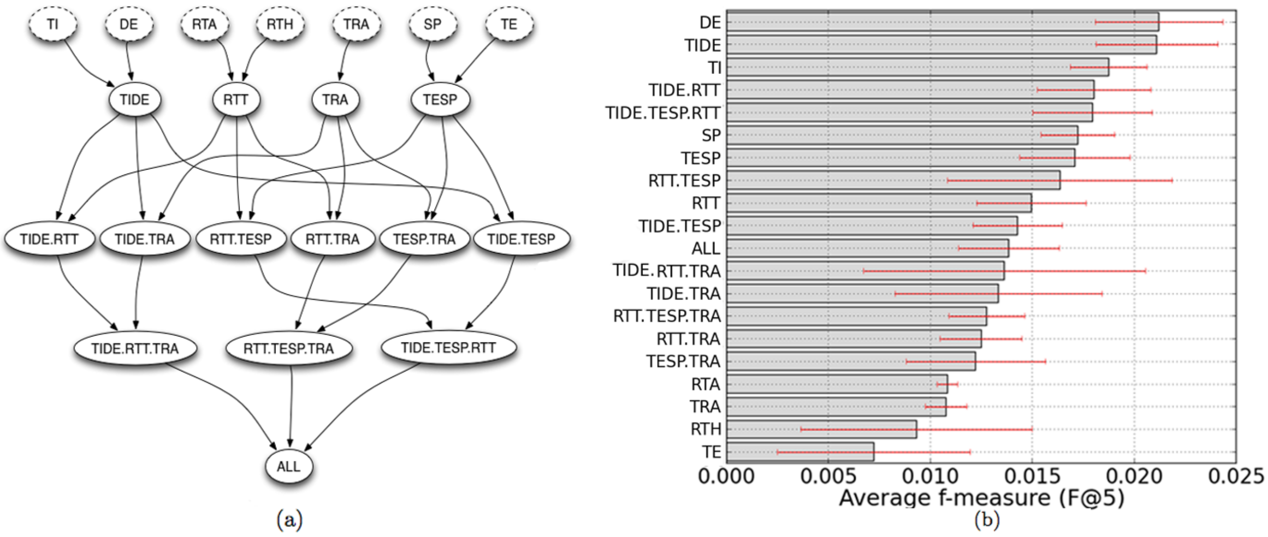

Fig. 1. (a) Combinations of features for comparison, and (b) ranking of individual and combined features based on the decreasing average of f-measure over all five methods. Atomic features are title (TI), description (DE), related tags (RTA), related themes (RTH), transcript (TRA), speaker (SP) and TED event (TE).

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in D^k(u;i)} d_{ij}(r_{uj} - b_{uj})}{\sum_{j \in D^k(u;i)} d_{ij}} \tag{3}$$

The bias estimate b_{ui} is the sum of the average ratings μ of items in the dataset, the average rating b_u of a user u and the average rating b_i for a given item i. The term r_{uj} is the rating value of a user u for a given item j. The coefficient d_{ij} expresses the similarity between item i and item j (see Eq. 4) by using the similarity s_{ij} between items i and j multiplied by a factor varying from 1 (when the number of common raters n_{ij} is considerably larger than λ) to 0 (when $n_{ij} \ll \lambda$). Typically, $\lambda \approx 100$.

$$d_{ij} = s_{ij}\frac{n_{ij}}{n_{ij} + \lambda};\ b_{ui} = \mu + b_u + b_i \tag{4}$$

We use two representative variants of this model. First, we use a normalized neighborhood model (as in Eq. 3) with Pearson Correlation for vector similarity, which is noted as *CF(PC)*. Second, we use a non-normalized one (noted with a preceding 'u' for 'unnormalized'), removing the denominator in Eq. 3, with a distance based on cosine similarity (noted as 'COS'); hence this is referred to as *uCF(COS)*. In previous studies [7], non-normalized models were found to perform better for the top-N recommendation task than the normalized ones.

C. Combining Collaborative Filtering with Content Similarity

We incorporate in the neighborhood model presented above information about content-based similarity, by replacing the d_{ij} similarity with a content-based one in Eq. 3. This new model allows us to exploit at the same time the semantic-based similarities (Eq. 2) and the bias estimate, i.e. to combine the two types of information, content and collaborative. This is especially useful when collaborative information is sparse, and the similarity computed using it is less reliable than the content-based one.

We will consider only the non-normalized versions of Eq. 3 (noted again with 'u'), and indicate the type of content-based similarity that is used in combination to the CF neighborhood model. Hence, these new models will be referred to as *uCF(TFIDF)*, *uCF(LSI)*, *uCF(LDA)*, *uCF(RP)* and *uCF(ESA)*.

For comparison purposes, we consider a user-independent recommender noted *TopPopular*, which always recommends the items with the highest popularity (based on the number of total ratings) regardless of a user's preferences.

VI. Parameter and Feature Selection

We determine the optimal parameters and features of the content-based methods using 5-fold cross-validation over the training set M, which includes 80% of the ratings for each of the 2,427 TED users that have made 12 or more ratings. The remaining ratings form the test set T used in the next section.

The CB methods use one or more lexical features (words) extracted from the fields of each TED talk, represented schematically in Figure 1, and several meta-parameters for each of the semantic representations (TF-IDF, LSI, RP, LDA, and ESA) as described in Section IV. However, exploring all possible combinations of features is not tractable. Therefore, we grouped individual features into four groups: title plus description (TIDE), related tags and themes (RTT), transcript (TRA), and speaker plus TED event (TESP). Along with all individual features, we tested these sets, and all their combinations, organized as in Fig. 1 (a).

For LSI and RP we varied the values of the parameter t (number of topics) from 10 to 5,000 and for LDA from 10 to 200 only, for tractability reasons. Additionally, for LDA, we varied the α and β parameters from 0 to 1, and the optimal ones were found to be $\alpha = 1$ and $\beta = 0.002$. We fix the value of neighborhood at $k = 3$, which is a trade-off between computational cost and expected prediction accuracy [17].

Figure 1 (b) displays the ranking of features and their combinations, ordered by the average f-measure (F@5) over *all* the tested methods. These results thus indicate which features perform well over *all* methods. Alternatively, the optimal features for *each* method are indicated in Table II.

The results show that the human-made description of talks (DE), the title (TI), and their combinations with other features

Method	Optimal Features	Performance (%)		
		P@5	R@5	F@5
LDA (t=200)	Title, desc., TED event, speaker (TIDE.TESP)	1.63	1.96	1.78
TF-IDF	Title (TI)	1.70	2.00	1.83
RP (t=5000)	Description (DE)	**1.83**	**2.25**	**2.01**
LSI (t=3000)	Title (TI)	**1.86**	**2.27**	**2.04**
ESA	Title, description (TIDE)	**2.79**	**3.46**	**3.08**

TABLE II. OPTIMAL FEATURES FOR CONTENT-BASED METHODS FOUND USING 5-FOLD C.-V. ON THE TRAINING SET. SCORES IN BOLD ARE SIGNIFICANTLY HIGHER THAN TF-IDF ONES (T-TEST, $p < 0.05$).

(TIDE, TIDE.RTT, and TIDE.TESP.RTT) are the most useful features for content-based personalized recommendations. In addition, knowledge of the speaker (SP) is useful too (ranked sixth). However, these metadata fields come to a cost because they must be entered by the editors of the lecture repository.

The lowest performing features were the name of the TED event (TE) and the related themes (keywords) assigned by TED experts (RTH), which presumably lack specificity for recommendation. Somewhat surprisingly, the transcript (TRA) decreases the performance of all methods and most of the combinations that include it are in the middle of the ranking. One explanation is that the huge size of the transcript's vocabulary introduces a lot of noise.

Table II shows the optimal features and parameters for each semantic representation used with *CB*, together with the scores (precision, recall and f-measure at 5) that they enable the recommender system to reach (5-fold cross-validation on the development data). All the semantic-based methods except LDA outperform significantly the TF-IDF baseline (t-statistic, $p < 0.05$): 11% improvement for LSI, 7.6% for RP and up to 64% by ESA, which reaches the best score. The good performance of ESA shows that the external-knowledge-based representation of the items is significantly more useful to our task than the domain knowledge captured intrinsically by the other methods.

VII. Personalized Lecture Recommendation

We compare recommendation performance of CB, CF and combined methods on the held-out test set T, considering two different settings: (i) a cold-start setting where the collaborative rating information for the items is not available and (ii) a non-cold-start setting where it is. Note that when testing, we only hide the rating information for the user currently tested.

A. Cold-start Recommendations (CB Methods Only)

The cold-start setting is characterized by sparse user ratings, with many items not having been rated at all, which makes it impossible for CF methods to recommend these items (e.g. new TED lectures). In this situations, only content-based methods can help making recommendations. In Figure 2, we show the performance of our CB methods in terms of precision and recall over the held-out set T. Most of the semantic-based representations perform significantly better (t-statistic, p < 0.05) than TF-IDF, with +62% for ESA, +7% for LSI and +8% RP. LDA does not improve over TF-IDF except at the top 1 to 4 recommendations (as also seen in Table II) and it was also the most difficult method to tune. The scores obtained appear to be small, however they are in line with previous works (e.g. [2], [7]).

The improvement brought by ESA appears to be again much greater than that of LSI and RP, allowing us to conclude that similarity based on concept spaces from external knowledge captures more effectively the content similarity (and, accordingly) the user preferences than the other semantic spaces and the baseline TF-IDF. Semantic-based approaches are thus more effective than keyword-based ones for cold-start personalized recommendations.

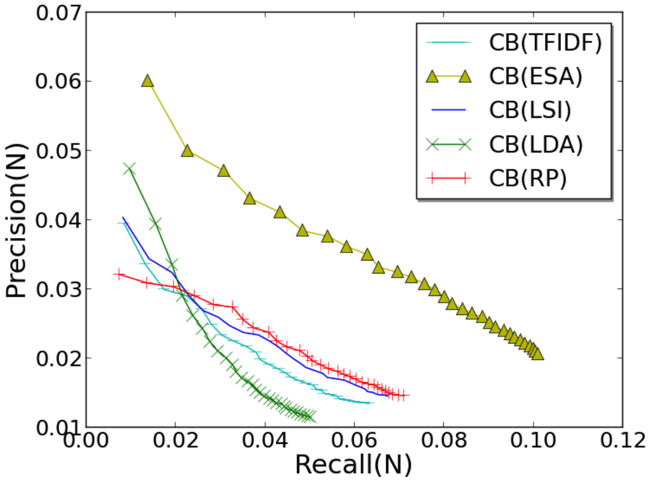

Fig. 2. Scores of content-based methods in a cold-start setting, in terms of precision and recall at N ($1 \leq N \leq 30$) on the held-out set T. The ESA-based distance outperforms by far all the others.

B. Non-Cold-Start Recommendations (All Methods)

In a non-cold-start setting, where the items have been rated by many users, the collaborative filtering (CF) information and the bias introduced by the popularity of items can be specifically exploited. As the CB methods do not have such information, their performance was found to be lower than CF methods, and will not be reported here. However, the combinations of CB and CF proposed in Section V-C (noted *uCF(·)* with · indicating the similarity method) allow content-based similarity to take into account the bias estimate, and their results are only slightly below pure CF methods in the non-cold-start scenario, while being operational in cold-start situations as well.

Fig. 3 displays the performance of two neighborhood models used for collaborative filtering: the normalized one using Pearson Correlation (*CF(PC)*) and the unnormalized one using cosine similarity (*uCF(COS)*). We also represent the two best performing combined methods, unnormalized, using TFIDF and LSI distances (*uCF(TFIDF)* and *uCF(LSI)*), as well as the *TopPopular* baseline. The best performance is achieved by the non-normalized neighborhood model with cosine similarity (+34%), *uCF(COS)*. The *CF(PC)* model is slightly below it, but is significantly better (+15%) than *TopPopular*.

The combined models, *uCF(TFIDF)* and *uCF(LSI)*, perform similarly to *CF(PC)* and are also significantly better (t-statistic, $p < 0.05$) than *TopPopular*, respectively +10.5% and +13% above it. The other content-based similarities (RP, LDA, ESA) performed slightly below TF-IDF, but the difference is

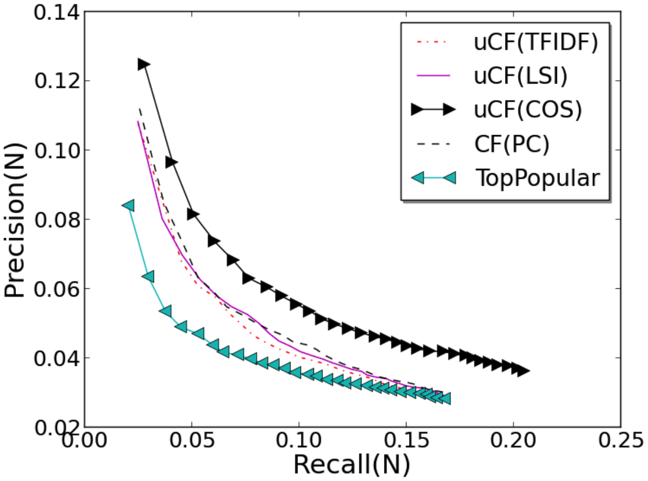

Fig. 3. Lecture recommendation scores for two CF (*CF(PC)* and *uCF(COS)*) and two combined methods (neighborhood with TFIDF and LSI distances) in a non-cold-start setting. Precision and recall at $1 \leq N \leq 30$ are computed on the held-out test set T. Collaborative filtering using cosine similarity in a neighborhood model scores highest, but the combined model using neighborhoods and TFIDF is not far behind.

not statistically significant. Using the bias introduced by the item popularity thus decreases the difference in performance between the content-based similarity models.

VIII. Discussion and Future Work

In this paper, we introduced a new dataset, the TED lectures, and defined lecture recommendation tasks utilizing the available ground truth. The feature selection experiments over 80% of the most active TED users indicated that the most informative data fields for CB methods were the description and the title of each lecture. Using cross-validation, CB using Explicit Semantic Analysis was found to outperform all other CB methods.

We then compared content-based, collaborative-filtering, and combined recommendation methods over the test set in two different settings: a cold-start one and a non-cold-start one. We showed that the semantic-based methods (ESA, RP and LSI) were able to make more relevant recommendations than keyword-based ones (TFIDF) in a cold-start setting, making them particularly applicable to multimedia datasets into which new items are inserted frequently. However, the CB methods were outperformed by CF ones in a non-cold-start setting, although a combined method using a neighborhood model, user/item biases and TF-IDF similarity achieved reasonable performance compared to pure CF by utilizing only the popularity bias. This method can be used when newly-added and older items are both present, as it does not rely entirely on collaborative rating similarities.

We will further explore algorithms inspired from such tasks, in particular hybrid ones, especially given that the TED dataset has rich content information to be exploited. We will also use semantic spaces with other learning models, such as matrix factorization, and improve the fusion of CB and CF information. Lastly, we will assess recommendation performance when automatically-assigned values are available for metadata fields, for instance through automatic speech recognition (for TRA), speaker detection (for SP), or automatic summarization (for DE).

IX. Acknowledgments

The work described in this article was supported by the European Union through the inEvent project FP7-ICT n. 287872 (see http://www.inevent-project.eu). We would like to thank the managers of the TED website for their support in accessing and distributing the TED metadata.

References

[1] E. Gabrilovich and S. Markovitch, “Computing semantic relatedness using Wikipedia-based explicit semantic analysis,” in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, ser. IJCAI’07, Hyderabad, India, 2007.

[2] R. Pan, Y. Zhou, B. Cao, N. Liu, R. Lukose, M. Scholz, and Q. Yang, “One-class collaborative filtering,” in *8th Int. Conf. on Data Mining*, Pisa, Italy, 2008, pp. 502–511.

[3] N. Pappas and A. Popescu-Belis, “Sentiment analysis of user comments for one-class collaborative filtering over TED talks,” in *Proceedings of the 36th ACM SIGIR Conference on Research and Development in Information Retrieval, Short Papers*, Dublin, Ireland, 2013.

[4] M. A. Hart, “The long tail: Why the future of business is selling less of more,” in *Journal of Product Innovation Management*. Blackwell Publishing Inc, 2007.

[5] N. Antulov-Fantulin, M. Bošnjak, M. Žnidaršič, M. Grčar, M. Morzy, and T. Šmuc, “ECML/PKDD 2011 discovery challenge overview,” in *Proceedings of ECML/PKDD 2011 Discovery Challenge Workshop*, Athens, Greece, 2011.

[6] G. Shani and A. Gunawardana, “Evaluating recommendation systems,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, 2011.

[7] P. Cremonesi, Y. Koren, and R. Turrin, “Performance of recommender algorithms on top-n recommendation tasks,” in *Proceedings of the fourth ACM conference on Recommender Systems*, ser. RecSys ’10, Barcelona, Spain, 2010.

[8] P. Lops, M. Gemmis, and G. Semeraro, “Content-based recommender systems: State of the art and trends,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, 2011.

[9] M. Sahlgren, “The wordspace model,” in *PhD Thesis*. Stockholm University, 2006.

[10] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” in *Information Processing and Management*. Pergamon Press, 1988.

[11] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and Development in Information Retrieval*, ser. SIGIR ’99, Berkeley, California, 1999.

[12] M. Sahlgren, “An introduction to random indexing,” in *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering*, Copenhagen, Denmark, 2005.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” in *Journal of Machine Learning Research*. JMLR.org, 2003.

[14] R. Řehůřek and P. Sojka, “Software framework for topic modeling with large corpora,” in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, Valletta, Malta, 2010, pp. 45–50.

[15] M. Papagelis and D. Plexousakis, “Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents,” in *Engineering Applications of Artificial Intelligence*. Pergamon Press, Inc., 2005.

[16] T. Mahmood and F. Ricci, “Improving recommender systems with adaptive conversational strategies,” in *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, ser. HT ’09, Torino, Italy, 2009.

[17] Y. Koren and R. Bell, “Advances in collaborative filtering,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer US, 2011, pp. 145–186.