Big Data and Information Analytics
©American Institute of Mathematical Sciences
Volume **1**, Number **2&3**, April & July **2016**

doi:10.3934/bdia.2016008
pp. **261–274**

TIME AWARE TOPIC BASED RECOMMENDER SYSTEM

Elnaz Delpisheh*, Aijun An
Heidar Davoudi and Emad Gohari Boroujerdi

Department of Electrical Engineering and Computer Science
York University, 4700 Keele Street, Toronto, Ontario M3J 1P3, Canada

(Communicated by Xiaomin Zhu)

Abstract. News recommender systems efficiently handle the overwhelming number of news articles, simplify navigations, and retrieve relevant information. Many conventional news recommender systems use collaborative filtering to make recommendations based on the behavior of users in the system. In this approach, the introduction of new users or new items can cause the cold start problem, as there will be insufficient data on these new entries for the collaborative filtering to draw any inferences for new users or items. Content-based news recommender systems emerged to address the cold start problem. However, many content-based news recommender systems consider documents as a bag-of-words neglecting the hidden themes of the news articles. In this paper, we propose a news recommender system leveraging topic models and time spent on each article. We build an automated recommender system that is able to filter news articles and make recommendations based on users' preferences. We use topic models to identify the thematic structure of the corpus. These themes are incorporated into a content-based recommender system to filter news articles that contain themes that are of less interest to users and to recommend articles that are thematically similar to users' preferences. Our experimental studies show that utilizing topic modeling and spent time on a single article can outperform the state of the arts recommendation techniques. The resulting recommender system based on the proposed method is currently operational at *The Globe and Mail* (http://www.theglobeandmail.com/).

1. Introduction.

People have always been confronting with a growing amount of data, which in turn demands more on their abilities to filter the content according to their preferences. Among the increasingly overwhelming amounts of webpages, documents, pictures, or videos, it is no longer intuitive to find what we really need. Furthermore, duplicate or several information sources are found covering the same topics. The users are sensitive to the recentness of information and their interests are also changing over time along with the content of the Web [23].

During the past two decades, the concepts of recommender systems have emerged to remedy the situation. The essence of recommender systems are highly associated with the extensive work in cognitive science, approximation theory, information retrieval, forecasting theories, and management science [1]. Recommender systems

2010 *Mathematics Subject Classification.* Primary: 58F15, 58F17; Secondary: 53C35.

Key words and phrases. Recommender system, topic model, LDA, cosine similarity, word embedding.

* Corresponding author: Elnaz Delpisheh.

have many applications, such as product recommendations at Amazon.com [18], movies recommendations by MovieLens [22], and news recommendations [1].

The increasing amount of electronic news articles requires better tools for searching, exploring, and organizing news article collections. Previously, news article were collected and stored in large text repositories and retrieved by a set of keywords. News article were seldom analyzed using their themes, because there were very few technologies to extract their thematic structures. Moreover, newspaper companies typically do not require users to subscribe and create their user profiles and users read news articles anonymously. Therefore, news recommender systems have to make recommendations without clear user profiles. In addition, many recommendation techniques face the cold start problem. This problem occurs when there is insufficient data to draw any inferences for new users or items [4].

To remedy the situation, in this paper, we design a news recommender system that eases reading and navigation through online newspapers. In essence, the recommender system acts as filters, delivering only news articles that can be considered relevant to a user. The resulting recommender system based on the proposed method is currently operational at *The Globe and Mail.* The Globe and Mail offers most authoritative news in Canada, featuring national and international news.

The major contributions of this paper are as follows:

- Inferring users' profiles and predicting users' preferences by analyzing contents of large collections of news articles.
- Designing a news recommender system based on both the content of news articles and users' time spent on each article.
- Experimental studies on a news corpus and outperforming baseline recommendation approaches in terms of precision, accuracy, and recall.

The structure of this paper is as follows. In Section 2, the related literature is reviewed. Section 3 describes main objectives of a news recommender system. Section 4 presents our proposed content-based news recommender system. In Section 5, we demonstrate the effectiveness of our approach through experiments. Section 6 concludes the paper and discusses future work.

2. Related work.

All of the known recommender techniques have strengths and weaknesses. In this section, we briefly survey the different recommender techniques, the data that they support, and the algorithms they employ [5, 6].

On this basis, the following three recommender techniques are distinguished: *Collaborative filtering-based*, *Content-based*, and *Hybrid-based.*

Collaborative filtering-based recommender systems make recommendations based on the behavior of other users in the system. Intuitively, these systems assume that if users agree about the quality of some items, then, they will likely agree about other items [9]. For example, if a group of users have similar tastes to Mary, then, Mary is likely to like the things the group likes which she hasn't seen yet. However, in this approach the introduction of new users or new items can cause the *cold start* problem, as there will be insufficient data on these new entries for the collaborative filtering to draw any inferences for new users or items. The system requires a substantial number of users to show interest to a new item before that item can be recommended [4, 6]. Addressing the cold start problem can be important for a new user's engagement and is therefore of critical significance in trade applications.

Content-based recommender systems recommend items similar to items a user preferred in the past [1]. For example, a content-based news recommender system observes the collection of news articles a user prefers and reads frequently. Then, only the news articles that have a high degree of similarity to the user's read articles are recommended. The greatest strength of this approach is that it only considers the properties of an item, i.e. the content of news articles, and accordingly makes recommendations. Therefore, in this approach, once a new user is introduced to the system, as soon as they read their first article, the content-based recommender system starts by recommending articles similar to the read article. Thus, this approach does not cause the cold start problem mentioned in collaborative recommender systems. The weakness of this approach is that users are limited to being recommended news articles that are similar to their read history.

Hybrid recommender systems generate recommendations by combining the above two recommendation techniques, thus, maximizing the benefits and minimizing the disadvantages of them [1]. For example, a hybrid recommendation system that combines content-based and collaborative recommendation systems considers both the content of news articles and a user's demographic information to issue recommendations. Given the fact that this approach contains collaborative recommender systems, it contains the disadvantages of such systems. Therefore, this approach also suffers from the cold start problem.

Due to the textual nature of our news application domain and avoiding the cold start problem, we focus on content-based recommender systems. Most existing content-based news recommender systems are based on keywords that is they represent the content of news articles using a set of keywords neglecting the thematic structure of the articles. We apply topic models to discover hidden themes of the news articles, and we incorporate these themes into a content-based recommender system. The reasons we employ topic models in news recommender systems are as follows. Firstly, topic models yield great insight about different themes of a newspaper article. Secondly, topic models capture probabilities of assigning different themes to newspaper articles. Thirdly, topic models provide a generative probabilistic model for the themes. As a consequence, topic models accurately assign probabilities to an unseen document. Our experimental studies show that the proposed recommender system yields more accurate results than other counterparts.

3. **Problem statement.**

News recommender systems arise to efficiently handle the overwhelming number of news articles, simplify navigations, and retrieve relevant information. Formally, the recommendation problem can be formulated as follows: Let $\mathcal{U}$ be the collection of $|\mathcal{U}|$ users, represented by $\mathcal{U} = \{u_1, u_2, \cdots, u_{|\mathcal{U}|}\}$, and let $\mathcal{C} = \mathcal{D} \cup \mathcal{Q}$ represent all the news articles, where $\mathcal{D}$, denoted by $\mathcal{D} = \{d_1, d_2, \cdots, d_M\}$, is the collection of *read articles* that is all news articles that have been read by at least one user, and $\mathcal{Q}$, denoted by $\mathcal{Q} = \{q_1, q_2, \cdots, q_N\}$, is the collection of *non-read articles* that is all the latest articles published daily that have not yet been read and are to be recommended. Note that our news recommender system is capable of personalizing the collection of non-read articles ($\mathcal{Q}$) for each user.

Let f be a utility function that measures the usefulness of a news article $c \in \mathcal{C}$ to a user $u_l \in \mathcal{U}$, i.e., $f : \mathcal{U} \times \mathcal{C} \rightarrow R$, where R is a totally ordered set (e.g., non-negative integers or real numbers within a certain range). Then, for each user $u_l \in \mathcal{U}$, we want to choose such news article $c^{'} \in \mathcal{C}$ that maximizes the user's utility. More formally:

$$\forall u_l \in \mathcal{U}, c^{'}_{u_l} = argmax_{c \in \mathcal{C}} f(u_l, c). \tag{1}$$

In recommender systems, the sets $\mathcal{U}$ and $\mathcal{C}$ are usually defined by several characteristics [1]. Similarly, in our work, each user $u_l \in \mathcal{U}$ is defined by a unique identifier, such as user ID. Each article in the collection $\mathcal{C}$ is defined by a unique article identifier and article content. In addition, we represent the utility of a news article by the amount of time a user spends on the article, which indicates the interestingness of the news article to the user. For example, user u_0 spent two minutes (out of five minutes[1]) on the news article *"d_0: SpaceX launches fifth official mission"*.

In our recommender system, the amount of time spent on the collection of non-read articles ($\mathcal{Q}$) is not available. Thus, the fundamental issue of our recommender system is that the utility function f is not defined on the whole $\mathcal{U} \times \mathcal{C}$ space, but only on $\mathcal{U} \times \mathcal{D}$ space. This means f needs to be extrapolated to the space $\mathcal{U} \times \mathcal{Q}$. Therefore, the goal of our news recommender system is to estimate the time each user would spend on the non-read news articles and issue appropriate recommendations based on these estimates.

4. The time aware topic based recommender system.

In this section, we propose a time aware content-based news recommender system by employing Latent Dirichlet Allocation (LDA). LDA-based topic modeling approaches to measure the similarity between read news articles and non-read news articles. LDA-based approaches elicit a topic model from the collection of news articles. The topic model represents news articles as a multinomial distribution over topics, where each topic is a multinomial distribution over words. Then, given the time a user has spent on read news articles, and the topic models of the collection of news articles, a user's time spent toward non-read news articles is estimated.

4.1. LDA-based topic models.

Latent Dirichlet Allocation (LDA), proposed by Blei *et al.* [3], is a generative probabilistic model for collections of discrete data such as text corpora. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. LDA also assumes that a corpus is a collection of D documents. Let $\mathcal{D} = \{w_1, w_2, \cdots, w_N\}$ represent a corpus of length N, resulting from the concatenation of the D documents which contains N words in total, where each word w_i belongs to a set of unique vocabulary words of size V [2]. LDA assumes that each word $w_i \in \mathcal{D}$ is associated with a latent topic variable z_i where $i \in \{1, 2, \cdots, N\}$. Each of these topics $t = 1 \cdots K$ is associated with a multinomial $\vec{\Phi}_t$ over V vocabulary words, such that $p(w_i|z_i = t) = \Phi_{z_i,w_i}$. Each $\vec{\Phi}_t$ is generated from a Dirichlet distribution with prior $\vec{\beta}$. Also, each document d is associated with a multinomial distribution $\vec{\Theta}_d$ over K topics, such that $p(z_i = t|d) = \Theta_{d,z_i}$, generated from a Dirichlet distribution with prior $\vec{\alpha}$. To discover the set of topics used in the corpus $\mathcal{D}$, the objective is (1) to obtain an estimate of $\underline{\Phi}$, where $\underline{\Phi} = \{\vec{\Phi}_t\}_{t=1}^{K}$, that is the term distribution for each topic, and (2) to obtain an estimate of $\underline{\Theta}$, where $\underline{\Theta} = \{\vec{\Theta}_d\}_{d=1}^{D}$, that is the topic distribution for each document. LDA is one such model.

In LDA, each document d is generated by first drawing a distribution over K topics with parameters $\vec{\Theta}_d$, generated from a Dirichlet distribution with prior $\vec{\alpha}$. The words in the document are then generated by drawing a topic $z_i = t$ from this

[1] In order to avoid idle time spent on a news article, we normalize the time by scaling between zero and five.

[2] This set of vocabulary words can be the set of unique words contained in the corpus with removal of stop words.

distribution and then drawing a word w_i from that topic according to a multinomial distribution with parameters $\vec{\Phi}_t$ generated from a Dirichlet distribution with prior $\vec{\beta}$ [3].

This procedure is a joint probability distribution over the random variables $(\mathcal{D}, \vec{z}, \underline{\Phi}, \underline{\Theta})$ given by [2]

Note that words are the only observed variables. The hyperparameters $\vec{\alpha}$ and $\vec{\beta}$ are input from the user. The latent topic assignments $\vec{z}$, document distributions over topics $\underline{\Theta}$, and topic distributions over words $\underline{\Phi}$ are all unobserved. Estimation of $\underline{\Theta}$ and $\underline{\Phi}$ requires computing the latent topic assignments $\vec{z}$, $p(\vec{z}|\mathcal{D}, \vec{\alpha}, \vec{\beta})$. Unfortunately, this posterior distribution is intractable due to the coupling between $\underline{\Phi}$ and $\underline{\Theta}$ [3]. However, Griffiths *et al.* [10, 11] proposed to use Gibbs sampling to obtain approximate estimates for the latent variables as well as the posterior distributions. In this method, the parameter sets $\underline{\Theta}$ and $\underline{\Phi}$ can be integrated out because they can be interpreted as statistics of the associations between the observed w_i and the corresponding z_i [10, 13].

With a set of samples from the posterior distributions $\underline{\Phi}$ and $\underline{\Theta}$ can be computed by integrating across the full set of samples. For any single sample we can estimate $\Theta_{d,t}$ by

$$\Theta_{d,t} = \frac{n_t^{(d)} + \alpha}{n_\cdot^{(d)} + K\alpha}, \tag{2}$$

where $n_t^{(d)}$ is the total number of words from document d assigned to topic t and $n_\cdot^{(d)}$ is the total number of words in document d.

Similarly, Φ_{t,w_i} is estimated by

$$\Phi_{t,w_i} = \frac{n_t^{(w_i)} + \beta}{n_t^{(\cdot)} + V\beta}, \tag{3}$$

where $n_t^{(w_i)}$ is the total number of times word w_i is assigned to topic t and $n_t^{(\cdot)}$ is the total number of words assigned to topic t.

4.2. **The proposed algorithm.** Our content-based recommender system employs probabilistic topic models to uncover the thematic similarity between news articles and a user's preferences. Then, news articles that have a high degree of thematic similarity to the user's preferences are recommended.

We assume a collection of users is represented by $\mathcal{U} = \{u_0, u_1, \cdots, u_{|\mathcal{U}|}\}$. Let the corpus of news articles be $\mathcal{C} = \mathcal{D} \cup \mathcal{Q}$, where $\mathcal{D} = \{d_1, d_2, \cdots, d_M\}$ is the collection of read articles, and $\mathcal{Q} = \{q_1, q_2, \cdots, q_N\}$ is the collection of non-read articles. We define a read article $d_i \in \mathcal{D}$ as a tuple of textual content and a subset of readers. That is $d_i =< t_i, U_i >$, where t_i is the textual content, represented by a sequence of terms of the article and $U_i \subset \mathcal{U}$ is a subset of users associated with the article. Similarly, a non-read article $q_j \in \mathcal{Q}$ is defined by $q_j =< t_j, \emptyset >$, where the set of readers is empty.

Our task is to appropriately recommend non-read articles to users or alternatively to assign users to non-read articles. In other words, for each non-read article $q_j =< t_j, \emptyset >$, we plan to predict the most appropriate subset of users and replace it with the empty set $(\emptyset)$.

The proposed content-based news recommender system consists of the following three steps.

4.2.1. *Building a topic model.* In this step, we use LDA-based topic models to best reflect the thematic structure of news articles. We build a topic model from the collection of read articles ($\mathcal{D}$). Our topic model assumes that each news article $d_i \in \mathcal{D}$ has a multinomial distribution over K topics with parameters $\vec{\Theta}_{d_i}$. As a result of this step, we obtain $\underline{\Theta_{\mathcal{D}}}$ that is an $M \times K$ array of topic probabilities given read articles, where M is the total number of read articles and K is the total number of topics.

4.2.2. *Inference and learning.* We use the topic model, built in Section 4.2.1, to infer the multinomial distribution of each non-read article ($q_j \in \mathcal{Q}$) over K topics with parameters $\vec{\Theta}_{q_j}$. As a result of this step, we obtain $\underline{\Theta_{\mathcal{Q}}}$ that is an $N \times K$ array of topic probabilities given non-read articles, where N is the total number of non-read articles and K is the total number of topics.

4.2.3. *Making recommendations.* For each user $u_l \in \mathcal{U}$, we obtain their collection of read articles $D_{u_l} \subset \mathcal{D}$ and their respective topic vectors $\underline{\Theta_{D_{u_l}}}$. Given a collection of non-read articles $\mathcal{Q}$, and their topic vectors $\underline{\Theta_{\mathcal{Q}}}$, our proposed method outputs a ranked list $Q_y^{u_l} = \{q_0, q_1, \cdots, q_y\}$, where $q_r \in \mathcal{Q}$, of y non-read articles interesting to a user u_l.

The probability of article q_r being interesting to user u_l is computed for each $q_r \in \mathcal{Q}$ as

$$p(q_r|u_l, \mathcal{Q}, D_{u_l}) = \frac{InterestingnessScore(q_r, u_l, D_{u_l})}{\sum_{q_j \in \mathcal{Q}} InterestingnessScore(q_j, u_l, D_{u_l})}, \tag{4}$$

$$InterestingnessScore(q_r, u_l, D_{u_l}) = \sum_{d_i \in D_{u_l}} DocSim(q_r, d_i, D_{u_l}) \cdot timeSpent[u_l, d_i]. \tag{5}$$

$InterestingnessScore(q_r, u_l, D_{u_l})$ calculates how interesting article q_r is to user u_l. This score can be any real non-negative number. $DocSim(q_r, d_i, D_{u_l})$ measures the similarity between two articles, i.e. q_r and d_i, given a collection of read articles by user u_l (D_{u_l}) and returns a similarity measure ranging between $[0, 1]$, and $timeSpent[u_l, d_i]$ is the amount of time user u_l spends on article d_i.

We apply LDA-based topic model to compute the article similarity. We utilize two arrays $\vec{\Theta}_{q_r}$ and $\vec{\Theta}_{d_i}$, obtained from Sections 4.2.1 and 4.2.2, to determine the similarity between q_r and d_i. Arrays $\vec{\Theta_{q_r}}$ and $\vec{\Theta_{d_i}}$ represent the latent topic distribution of articles q_r and d_i. Thus, inspired from Chang *et al.* [8], we view each article as a topic-based vector and use *cosine-based* similarity measure to compute the similarity between a read and a non-read article. Note that our experimental studies show similar results for other similarity measure approaches, such as Manhattan distance. A comprehensive survey on similarity measures between vectors can be found at [7].

Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The more similar hence the more co-oriented the vectors, thus the cosine of the angle between them is closer to one. Cosine similarity measure is often used to compare documents for text mining, classification, and clustering purposes [7]. Equation 6 is used to calculate the similarity.

$$cosine - similarity(\vec{\Theta}_{q_r}, \vec{\Theta}_{d_i}) = \frac{\vec{\Theta}_{q_r} \cdot \vec{\Theta}_{d_i}}{|\vec{\Theta}_{q_r}| \times |\vec{\Theta}_{d_i}|}, \tag{6}$$

where "·" denotes the inner product of two vectors, and $|\vec{x}|$ represents the size of the vector.

Finally, we return top y articles ranked by the $p(q_r|u_l, \mathcal{Q}, D_{u_l})$ probability.

5. **Experiments.** We conducted experiments on *The Globe and Mail* news article corpus. The Globe and Mail collection appeared on The Globe and Mail newswire during the period between January 2013 and March 2014. The articles were assembled and indexed with article IDs by personnel from The Globe and Mail. The Globe and Mail corpus contains $73,909$ news articles. Moreover, the collection contains $10,150$ subscribed users that have spent some time, i.e. any real non-negative number between one and five minutes, on each article. In order to avoid idle time spent on a news article, we normalize the time by scaling between zero and five. The news articles are divided into $73,000$ read articles that are read by at least one reader and 909 non-read articles that are recently published.

We compare the performance of our proposed content-based recommender system against the following baseline recommendation systems:

5.1. **A tfidf-based content-based recommendation system.** In this recommendation system solely bag-of-words tfidf representation of news articles is used. Term frequency-inverse document frequency (tfidf) [15] is a statistical measure that increases proportionally to the frequency of a term in a document but lessens by the frequency of the term among documents in the corpus. The $tfidf$ score of a term t in document d, represented by *tfidf(t,d)*, is defined as

$$tfidf(t, d) = tf_{t,d} \times \log \frac{M}{df_t}, \tag{7}$$

where $tf_{t,d}$ measures the ratio of the number of times term t appears in document d to the total number of terms in document d, and M is the total number of documents in a corpus, and df_t is the number of documents containing term t.

5.2. **Item popularity.** In this method items are ranked based on the spent time on each article as a popularity measure. In fact, the results based on popularity are not personalized but are used in many research [12] to show the effectiveness of methods.

5.3. **Item-to-Item collaborative filtering (ItemKNN).** Item-to-Item collaborative filtering method has been commercially used by Aamzon [18]. Each article is represented by a vector of users on which they have spent time, then cosine measure is utilized to assess the similarity among articles. We tested the method with different number of neighbors and found 80 is the best. ItemKNN using Binary feature ignores the amount of time spent on each article. Instead, this method represents each article by a vector of users who have or have not read the article.

5.4. **User-to-user collaborative filtering (UserKNN).** User-to-User collaborative filtering is a classical collaborating filtering method [24]. This method is similar to ItemKNN where each user is represented by a vector of articles she has read and similarities are computed among the users (rather than items). In this method, we used the same setting as the ItemKNN. UserKNN using Binary feature, similar to ItemKNN using Binary feature, ignores the amount of time spent on each article.

5.5. **Non-negative matrix factorization (NMF).** The method is based on Non-negative Matrix factorization [17] where the user-article matrix is factorized into two matrices with the property that all matrices have no negative value. Compared to traditional matrix factorization, the result of this method is interpretable and are more proper for the ranking tasks in recommendation. For this method, we set number of factors to 30 as increasing it had no significant effect on the result.

5.6. **Content-based recommender system using document embedding.** Document Embedding learns a vector-space representation of the terms of a document by exploiting a two-layer neural network [21, 16]. The architecture of the model that is used for training document embedding is the distributed memory phrase vector. When the model is trained on a dataset of documents, it tunes the word vectors and document vectors according to stochastic gradient descent optimization. In our experiment, the model is trained on a corpus of $73,000$ news articles. The learned document vectors are extracted from the model and used to determine the similarity of two articles regarding the cosine similarity of the vectors.

5.7. **LDA-based recommendation system.** LDA-based recommendation system is explained in Section 4.2. The topic models were trained with 1000 iterations of Gibbs sampling [10, 11] used in the MALLET [20]. Initial values for the hyper-parameters α and β applied to all our experiments are $\alpha = 50.0/K$ and $\beta = 0.01$. Note that these parameters are default parameters of most LDA-based topic models, expected to result in a fine-grained decomposition of the corpus into topics [11].

The optimum number of topics is expected to result in a fine-grained decomposition of the corpus into topics [11], where topic distributions over words are of minimum similarity. Furthermore, the optimum number of topics leads to a low cross-entropy between the term distribution learned by the topic model and the distribution of terms in an unseen test article. Thus, the optimum number of topics results in a lower perplexity score indicating that the model is better in predicting distribution of the test article [3].

In our experiments, we learn topics for different values of K and choose the value which minimizes the perplexity score. The experiments are conducted using different topic models for different number of topics K, where $K = 20 \cdots K = 300$. Figure 1 illustrates the average perplexity as a function of number of K. In this figure, the values of $K \in [180 \cdots 190]$ achieve the best performance in terms of perplexity.

As mentioned earlier, a topic model generates K topics, where each topic is a distribution over V words, denoted by $\vec{\Phi}_k = \{w_1, w_2, \cdots, w_V\}$. Similarity between topics is the similarity of topic distributions over words across different topics. We calculate the normalized average sum of similarity scores between every pair of K topics ($K \in [180 \cdots 190]$), generated from The Globe and Mail corpus. As illustrated

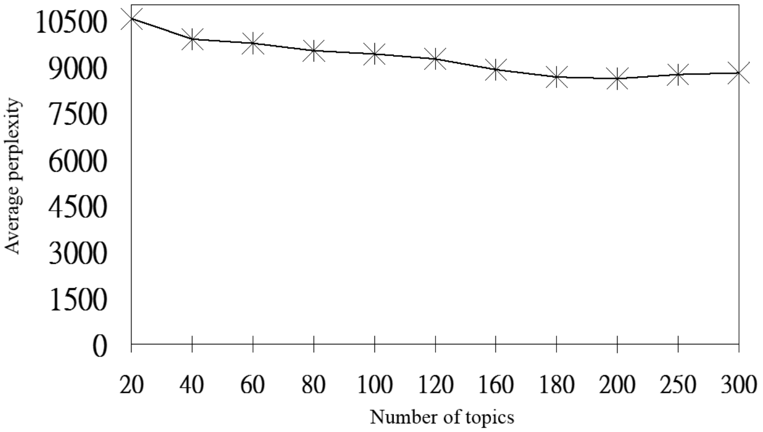

FIGURE 1. Average perplexity as a function of number of topics, using the LDA-based topic model on The Globe and Mail corpus.

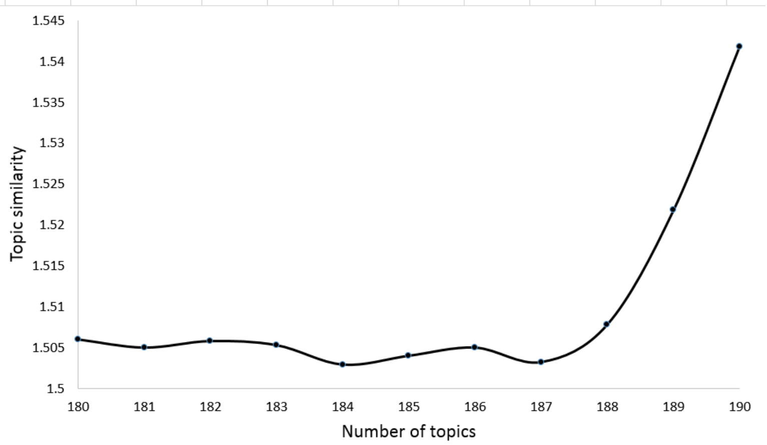

FIGURE 2. Similarity of topic distributions over words, as a function of number of topics, using LDA on The Globe and Mail corpus.

in Figure 2, $K = 187$ results in the most fine-grained decomposition of the corpus into topics with the minimum similarity between topic-word distributions.

5.8. **Evaluation of the recommender system.** In this section, we evaluate the performance of our proposed content-based news recommender system using the following metrics: *precision*, *recall*, and *F-measure*.

Precision, recall, and F-measure are well-known evaluation metrics in information retrieval literature [19]. For each user, we use the original set of read articles as the ground truth T_g. Assume that the set of recommended news articles are T_r, so that the correctly recommended articles are $T_g \cap T_r$. Precision, recall, and F-measure are defined as follows:

$$precision = \frac{|T_g \cap T_r|}{|T_r|}, \tag{8}$$

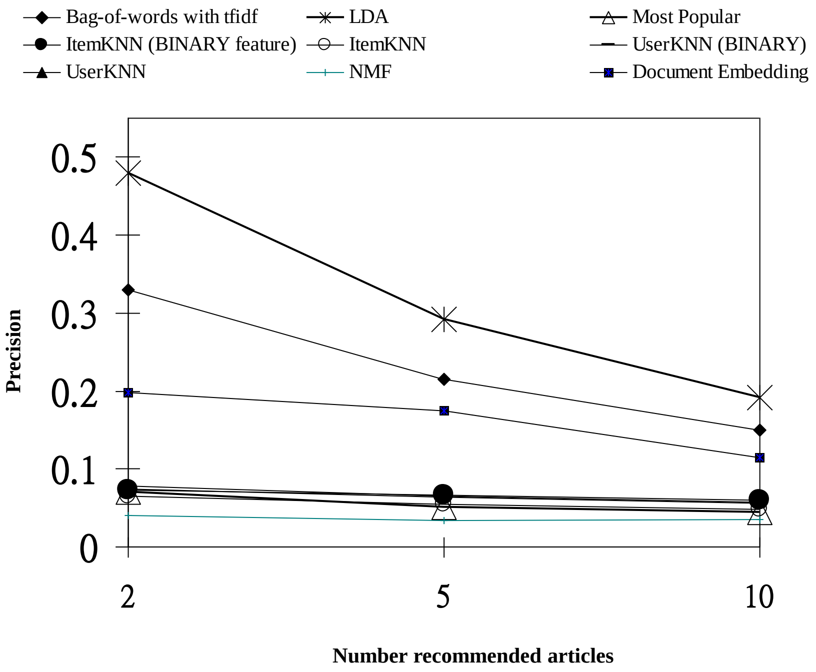

Figure 3. Precision of the proposed recommender system as a function of number of recommended articles, using the following recommendation systems: bag-of-words with tfidf, Item popularity, Item-to-Item collaborative filtering (ItemKNN), User-to-user collaborative filtering (UserKNN), Non-negative matrix factorization (NMF), Content-based recommender system using document embedding, and LDA on The Globe and Mail corpus.

$$recall = \frac{|T_g \cap T_r|}{|T_g|}, \tag{9}$$

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}. \tag{10}$$

In our experiments, the number of recommended articles ranges from 1 to 30. Figures 3, 4, and 5 illustrate the precision, recall, and F-measure of the proposed recommender system as a function of number of recommended articles.

Empirical comparisons show that using topic models to represent articles improves the precision, recall, and F-measure. Since the only difference between the comparisons is the article similarity function $DocSim(q_r, d_i, D_{u_l})$, which compares the similarity between a new non-read article q_r and a read article d_i, analyzing the differences between the two article similarity measures provides explanation about the performance difference.

The bag-of-words with tfidf approach represents two articles by tfidf vectors. Then, the cosine similarity between these vectors are computed and used in the recommendation system. Generally speaking, the tfidf article similarity measures the quantity of term overlap, where each term has a different weight, in the two

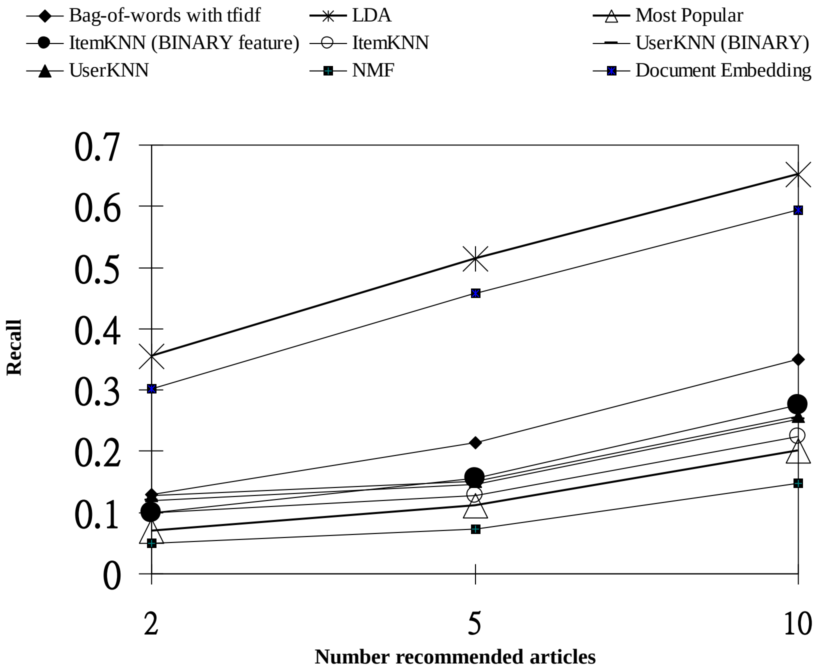

Figure 4. Recall of the proposed recommender system as a function of number of recommended articles, using the following recommendation systems: bag-of-words with tfidf, Item popularity, Item-to-Item collaborative filtering (ItemKNN), User-to-user collaborative filtering (UserKNN), Non-negative matrix factorization (NMF), Content-based recommender system using document embedding, and LDA on The Globe and Mail corpus.

articles [25]. This approach ignores the thematic structures of articles to perform the similarity measure.

The LDA-based approaches first generate a set of topic vectors for the articles, each of which is represented by a distribution over terms. Terms in each topic are semantically coherent. Then, LDA-based recommender systems measure the cosine similarity between the topic vectors. Generally speaking, using LDA-based topic vectors quantifies the topic similarity between the two articles. These vectors yield a higher precision, recall, and F-measure than when using tfidf or document embedding vectors. Key to this improvement is incorporating thematic structure of news articles into the recommendation system. This leads to better estimates for topic similarity between two articles.

Hence we recommend using topic models to represent articles for time aware content-based news recommender systems.

6. **Conclusions.** This paper presents a time aware topic based recommender system for The Globe and Mail, a company that offers most authoritative news in Canada, featuring national and international news. One of the important problems of The Globe and Mail newswire is the growing amount of articles, which in

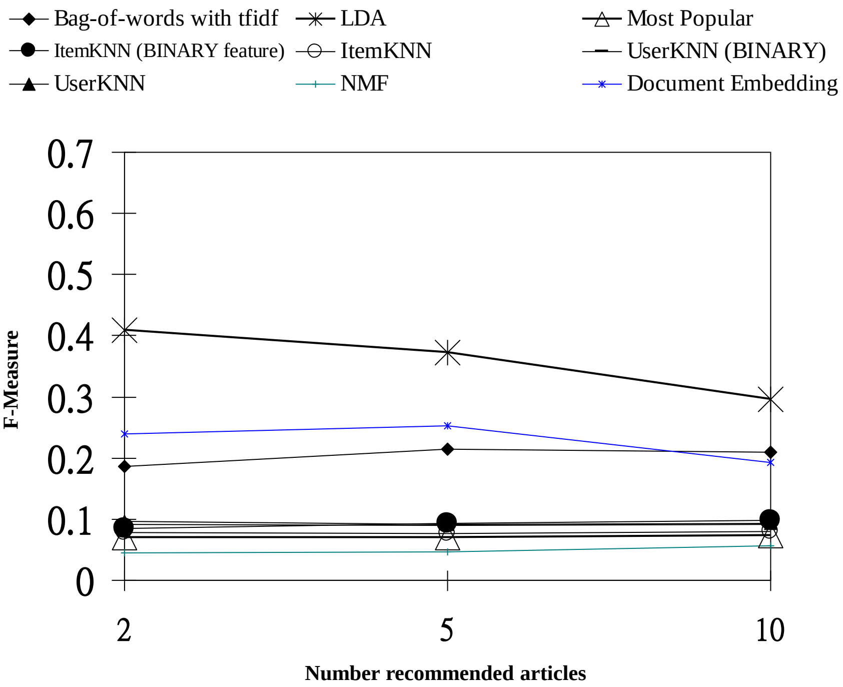

Figure 5. F-measure of the proposed recommender system as a function of number of recommended articles, using the following recommendation systems: bag-of-words with tfidf, Item popularity, Item-to-Item collaborative filtering (ItemKNN), User-to-user collaborative filtering (UserKNN), Non-negative matrix factorization (NMF), Content-based recommender system using document embedding, and LDA on The Globe and Mail corpus.

turn demands a system to automatically filter and deliver the content according to readers' preferences. Furthermore, in the collaborative-filtering-based recommender system at The Globe and Mail, the introduction of new news articles can cause the cold start problem, as there will be insufficient data on these new entries for the collaborative filtering to work accurately.

We propose to utilize the latent Dirichlet allocation (LDA) model to discover hidden themes of the news articles. We incorporate these themes into a content-based recommender system. Our experimental studies show that the proposed recommendation system yields better results than solely bag-of-words with tfidf presentation. Moreover, given the fact that our recommender system considers the content of news articles to make recommendations, introducing a new news article does not cause the cold start problem.

Applying topic models in a content-based recommender system yields more accurate results than other recommender systems. However, our content-based recommender system must effectively evolve with its content. In our current system, the topic model needs to be generated offline. For instance, once non-read news articles enter the collection of read articles, the topic model needs to be updated to reflect the themes of new articles. This offline generation of a topic model is a drawback, as it hinders the system's ability to evolve quickly. We could develop a real-time content-based recommender system, that leverages a stream of news articles and is capable of handling online LDA [14] a

Acknowledgments. We would like to thank the data science group at **The Globe and Mail**, in particular, **Michael O'Neill, Gordon Edall, and Shengqing Wu**, for providing us with the data set used in this research, and insight and expertise that greatly assisted this research.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Transaction on Knowledge and Data Engineering*, **17** (2005), 734–749.

[2] D. M. Andrzejewski, *Incorporating Domain Knowledge in Latent Topic Models*, PhD thesis, University of Wisconsin-Madison, USA, 2010.

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

[4] J. Bobadilla, F. Ortega, A. Hernando and J. Bernal, A collaborative filtering approach to mitigate the new user cold start problem, *Knowledge-Based System*, **26** (2012), 225–238.

[5] H. Borges and A. Lorena, A survey on recommender systems for news data, in *Smart Information and Knowledge Management* (eds. E. Szczerbicki and N. Nguyen), vol. 260 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 2010, 129–151.

[6] R. Burke, Hybrid recommender systems: Survey and experiments, *User Modeling and User-Adapted Interaction*, **12** (2002), 331–370.

[7] S.-H. Cha, Comprehensive survey on distance/similarity measures between probability density functions, *International Journal of Mathematical Models and Methods in Applied Sciences*, **1** (2007), 300–307.

[8] T.-M. Chang and W.-F. Hsiao, Lda-based personalized document recommendation, Proceedings of the PACIS, 2013.

[9] M. D. Ekstrand, J. T. Riedl and J. A. Konstan, Collaborative filtering recommender systems, *Journal of Foundations and Trends in Human-Computer Interaction*, **4** (2011), 81–173.

[10] T. Griffiths, Gibbs sampling in the generative model of latent dirichlet allocation, *Standford University*, **518** (2002), 1–3.

[11] T. L. Griffiths and M. Steyvers, Finding scientific topics, *Proceeding of the National Academy of Sciences of the United States of America*, **101** (2004), 5228–5235.

[12] X. He, T. Chen, M.-Y. Kan and X. Chen, Trirank: Review-aware explainable recommendation by modeling aspects, in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, ACM, New York, NY, USA, 2015, 1661–1670.

[13] G. Heinrich, Parameter estimation for text analysis, `http://www.arbylon.net/publications/text-est.pdf`.

[14] M. D. Hoffman, D. M. Blei and F. R. Bach, Online learning for latent dirichlet allocation., in *NIPS* (eds. J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel and A. Culotta), Curran Associates, Inc., 2010, 856–864.

[15] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st edition, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.

[16] Q. V. Le and T. Mikolov, Distributed representations of sentences and documents, *CoRR*, abs/1405.4053, URL `http://arxiv.org/abs/1405.4053`.

[17] D. D. Lee and H. S. Seung, Algorithms for non-negative matrix factorization, in *Advances in Neural Information Processing Systems 13* (eds. T. K. Leen, T. G. Dietterich and V. Tresp), MIT Press, 2001, 556–562, URL `http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf`.

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

[19] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, Massachusetts, 1999.

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

[21] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, *CoRR*, abs/1310.4546, URL `http://arxiv.org/abs/1310.4546`.

[22] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan and J. Riedl, Movielens unplugged: Experiences with an occasionally connected recommender system, in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, ACM, New York, NY, USA, 2003, 263–266.

[23] D. Z. Mária Bieliková Michal Kompan, Effective hierarchical vector-based news representation for personalized recommendation, *Computer Science and Information Systems*, 303–322, URL `http://eudml.org/doc/252774`.

[24] F. Ricci, L. Rokach and B. Shapira, *Recommender Systems Handbook*, chapter Introduction to Recommender Systems Handbook, Springer US, Boston, MA, 2011.

[25] S. Tuarob, L. C. Pouchard and C. L. Giles, Automatic tag recommendation for metadata annotation using probabilistic topic modeling, in *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '13, ACM, New York, NY, USA, 2013, 239–248.

Received August 2016; revised October 2016.

E-mail address: `elnaz@cse.yorku.ca`
E-mail address: `aan@cse.yorku.ca`
E-mail address: `davoudi@cse.yorku.ca`
E-mail address: `gohari@cse.yorku.ca`