Enriching Topic Coherence on Reviews for Cross-Domain Recommendation

Article *in* The Computer Journal · May 2020

DOI: 10.1093/comjnl/bxaa008

Mala Saraswat
ABES Engineering College
27 PUBLICATIONS **84** CITATIONS

doi: 10.1093/comjnl/bxaa008

Enriching Topic Coherence on Reviews for Cross-Domain Recommendation

Mala Saraswat[1,2,*] and Shampa Chakraverty[1]

[1]*Department of Computer Engineering, Netaji Subhas University of Technology (NSUT), Dwarka Sector-3, Dwarka, Delhi, 110078, India*
[2]*Department of Computer Science and Engineering, ABES Engineering College, Campus -1, 19th KM Stone, NH 24, Ghaziabad, Uttar Pradesh 201009*
**Corresponding author: malasaraswat@gmail.com*

With the advent of e-commerce sites and social media, users express their preferences and tastes freely through user-generated content such as reviews and comments. In order to promote cross-selling, e-commerce sites such as eBay and Amazon regularly use such inputs from multiple domains and suggest items with which users may be interested. In this paper, we propose a topic coherence-based cross-domain recommender model. The core concept is to use topic modeling to extract topics from user-generated content such as reviews and combine them with reliable semantic coherence techniques to link different domains, using Wikipedia as a reference corpus. We experiment with different topic coherence methods such as pointwise mutual information (PMI) and explicit semantic analysis (ESA). Experimental results presented demonstrate that our approach, using PMI as topic coherence, yields 22.6% and using ESA yields 54.4% higher precision as compared with cross-domain recommender system based on semantic clustering.

Keywords: recommender systems; topic modeling; pointwise mutual information; explicit semantic analysis; topic coherence

Received 2 April 2019; Revised 30 November 2019; Editorial Decision 11 January 2020; Accepted 11 January 2020
Handling editor: Fionn Murtagh

1. INTRODUCTION

Topic models extract the latent topics in a set of documents by conducting a statistical analysis of word occurrences in them. Each such generated latent topic consists of topic words. Topic coherence is a method for evaluating the quality of topic models [1, 2]. A good topic model generates cohesive topics with high topic coherence scores equal to that generated by humans.

Various topic models and topic coherence methods have been proposed for different applications such as text classification [3] and Twitter analysis [4]. Recently, topic modeling has been used to improve the performance of recommender systems. This application is based on the concept that a system would recommend articles to a user with a similar topic structure as that of the articles liked in the past [5].

From the literature survey, we found that recent research works have applied topic modeling on unstructured text data to perform multi-domain collaboration [6, 7]. Tang *et al.* [6] proposed a Cross-Domain Topic Learning method that performs cross-domain collaborations between various research domains such as Data Mining, Medical Informatics, Visualization, Theoretical Computer Science and Databases. Low *et al.* [7] proposed a hierarchical Bayesian model for user personalization across several internet portals. The model integrates recommendation systems over different domains such as Yahoo FrontPage, Yahoo News and My Yahoo to provide a personalized experience [7].

The main drawback of the above works is that their performance degrades rapidly when the vocabularies of the domains differ widely, even though they may be semantically related. For example, Medical Informatics vocabulary will differ from the vocabulary used in databases, yet there may be several semantic linkages between the two. To address this issue, it becomes necessary to find meaningful correlations between the vocabularies of different domains. Kumar *et al.* [8] tackled this problem by deriving semantic relationships between words of vocabularies of participating domains, thereby building a common semantic space. Their approach, named semantic clustering-based cross-domain (SCD), form clusters of words from different domains based on their semantic relatedness as gathered from WordNet. Next, latent Dirichlet allocation (LDA)-based topic modeling is performed on the clustered semantic space instead of individual words. Finally, the similarity between the probability distribution of topics between the different domains generates cohesive recommendations. The authors demonstrate that there is a significant improvement in the precision of cross domain recommender system (CDR) using semantic clustering as compared when non-clustered vocabularies were used.

Aletras and Stevenson [9] experimented with a dataset consisting of news articles from the New York Times. The authors demonstrated that when the similarity between topics is derived based on co-occurrence semantics such as pointwise mutual information (PMI) and knowledge-based metrics such as explicit semantic analysis (ESA) rather than cosine distances between word probability distributions as was used in [10, 11], it provides more reliable estimates of topic similarity that are comparable with human estimates [9]. Newman *et al.* [2] evaluated a wide range of topic coherence methods over News and Books datasets, drawing upon WordNet, Wikipedia and the Google search engine as information resource. Experiments demonstrate that the best performing method was PMI using Wikipedia as a resource. The authors concluded that Wikipedia's encyclopedic nature resulted in good coverage of vocabularies over both domains and was thus more robust.

Taking a cue from the topic modeling approach proposed in [8] and combining it with PMI- and ESA-driven approaches proposed in [1] and [9], we build our topic coherence-based cross-domain recommender (TC-CDR) system. In this work, we use PMI and ESA as measures of topic coherence. Before describing our system, let us revisit the concepts underlying these measures.

Semantic analysis on textual corpus of reviews allows us to capture the context of the reviews. Unlike traditional approaches, our TC-CDR approach uses external knowledge resource such as Wikipedia to exploit contextual information from reviews for linking different domains. Recurrent neural network (RNN), a deep learning approach, models the effect of user historical sequence behavior on the user current behavior and predicts current users behavior based. But for real-world problems, RNN has difficulty in modeling contextual information that is very important for behavior modeling [12].

The remaining paper is organized in the following manner. Section 4.1 discusses the different approaches for obtaining topic coherence between the participating domains in a multi-domain application. We discuss the details of our proposed TC-CDR approach in Section 4.2. We demonstrate experimental results and present their analysis in Section 4.3. In Section 4.4, we conclude our findings.

2. MEASURES FOR TOPIC COHERENCE

Our TC-CDR-based approach uses the following measures of topic coherence for providing CDR in various domains.

2.1. Pointwise mutual information

PMI captures the semantic similarity of pairs of words, by empirically estimating occurrence probabilities from knowledge sources such as Wikipedia, WordNet and Google [11]. PMI is defined as an average of pairwise word similarities formed by the top n words. These top n words are selected by their probability distribution for a given topic. Some researchers use median instead of average pairwise word similarities for computing coherence [2].

A topic $\mathbf{W}$ is represented by a set of n topic words. Thus $\mathbf{W} = \{w_1, w_2, \ldots w_N\}$.

The PMI between two words, w_1 and w_2, is given by

$$PMI(w_i, w_j) = \log \frac{p(w_i, w_j)}{p\,(w_i) \times p(w_j)}, \qquad (1)$$

where $p(w_i)$represents the probability of seeing a topic word w_i, $p(w_j)$ represents the probability of seeing topic word w_j and $p(w_i, w_j)$ represents the probability of seeing both w_i and $w_j$co-occurring in a random document.

For computing topic coherence over topic $\mathbf{W}$, the semantic similarity of all pairs of words associated with that topic is averaged:

$$Coherence\ (W) = \frac{1}{\sum_{m=1}^{n-1} m} \sum_{i=1}^{n} \sum_{j=i+1}^{n} PMI(w_i, w_j). \qquad (2)$$

2.2. Explicit semantic analysis

Gabrilovich and Markovitch [13] proposed ESA for computing semantic relatedness of natural language texts using very large scale repositories such as Wikipedia and Open Directory Project (ODP). Wikipedia and ODP are well known knowledge repositories incorporating millions of human-defined concepts and provide a plethora of information about each concept. The term *explicit* signifies that semantic analysis in ESA uses human-defined concepts explicitly based on human cognition rather than latent topics generated from statistical co-occurrence information as is done in latent semantic analysis.

'ESA uses machine learning techniques to explicitly represent the meaning of any text as a weighted vector of Wikipedia-based concepts ordered by their relevance to the input' [14]. Wikipedia provides a large and diverse knowledge base. Here, each article is considered a distinct concept. All vectors of words occurring in the article weighted by their term frequency–inverse document frequency (TF-IDF) scores represent these concepts.

Let $d_1, \ldots, d_N$ be a set of documents and let $C_1, \ldots, C_N$ be a set of associated concepts. ESA assigns semantic interpretation to words using association-based method. For example, the word *cat* relates rigidly to the concepts *feline* and *pet* and less rigidly to the concepts *mouse* and *Tom & Jerry* [14]. For building a semantic interpreter, a table $\mathbf{T}$ is formed where each of its $\mathbf{N}$ columns corresponds to a specific document and its

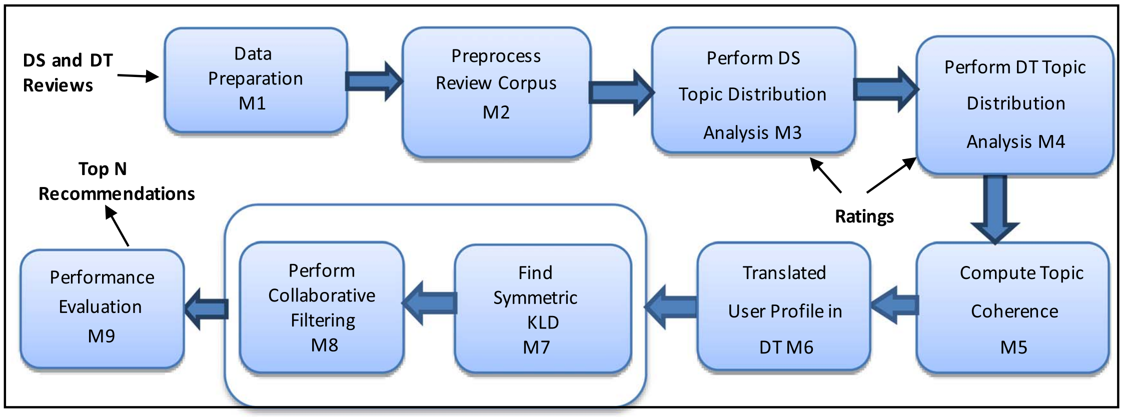
DS and DT Reviews
Data Preparation M1
Preprocess Review Corpus M2
Perform DS Topic Distribution Analysis M3
Perform DT Topic Distribution Analysis M4
Ratings
Compute Topic Coherence M5
Translated User Profile in DT M6
Find Symmetric KLD M7
Perform Collaborative Filtering M8
Performance Evaluation M9
Top N Recommendations

FIGURE 1. Block diagram of the proposed TC-CDR system.

associated concept and each of its **M** rows corresponds to a word that is occurring in $\bigcup_{i=1}^{N} d_i$. An entry T[i, j] in the table represents the TF-IDF value of the word or phrase t_i in document d_j. Then cosine normalization is applied to each row

$$T[i,j] = \frac{T[i,j]}{\sqrt{\sum_{i=1}^{r} T[i,j]^2}}, \tag{3}$$

where r is the number of terms in row i. This results in a sparse table **T**, which acts as an inverted index, mapping each word into a list of related concepts.

Thus for a word t_i, row i of the table **T** represents its semantic interpretation. T is given by a vector of concepts weighted by their respective $T[i,j]$ scores. This value reflects the relevance of the concept C_j to that word t_i. For example, the term ‘Mars’ generates the concepts planet, solar system and Jupiter with their corresponding scores as follows:

Mars: planet—0.90, solar system—0.85, Jupiter—0.30.

For a text fragment consisting of a set of words, $< t_1 \ldots t_k >$, the semantic interpretation is the centroid of the weighted vectors representing the individual words/phrases that comprise the fragment.

ESA thus represents text as a systematically derived interpretation vector in a high-dimensional space of concepts. To compute the semantic relatedness of two texts using this representation, we need to compare their interrelation vectors using cosine similarity.

3. FRAMEWORK FOR TOPIC COHERENCE BASED CDR APPROACH

Figure 1 illustrates the framework of the proposed TC-CDR system. There are two domains of interest to a user u: (i) the source domain (DS) in which the user u is active, i.e. he/she purchases and rates its items regularly and (ii) the target domain (DT) which consists of items that may be of interest to the user but are hitherto unexplored. The aim of TC-CDR is to recommend items from the DT to the user based on (i) the user’s rating patterns in DS and (ii) topic coherence between the topics of both domains. Our approach works well even when there is no user overlap between the two domains, i.e. there is no common user who has given his ratings in both DS and DT.

The first module M1, *data preparation*, extracts user-item rating matrices and user reviews from the source and DTs. It prepares a consolidated textual corpus for each item included in the two domains. These raw corpora are pre-processed in module M2. They are then processed further for topic distribution analysis in modules M3 for DS and M4 for DT. The topic distributions of the corpora in turn help generate the item and user profiles in both domains, which serve as links between the two domains to perform CDR.

Modules M5–M8 process the profiles thus generated to recommend items from DT to a user in DS. Module M5 first computes topic coherence between all pairs of topics from the two domains. M6 uses topic coherences to translate the user’s DS profile into a DT profile. M7 evaluates the distances between the translated user’s profiles and its equivalent DT user profiles using symmetric Kullback–Leibler divergences (KLD) [15]. Module M8 then performs collaborative filtering to generate Top N recommendations. Module M9 is the performance evaluation module that evaluates the performance of the TC-CDR system.

To illustrate the proposed TC-CDR approach, we use movies and books as representative entertainment domains, using the Movielens dataset [16] and Internet Movie Database, i.e. imdb.com to represent the movie domain and the Bookcrossing dataset [17] and amazon.com to represent the book domain. We now focus on each module of the TC-CDR system shown in Fig. 1 in greater detail.

3.1. Data preparation

The data preparation module M1 first extracts the user-item rating matrices for the source and DTs. Users express their experience, sentiments and opinions on their chosen items through reviews on e-commerce websites. The data preparation step builds a textual corpus for each and every item in the source and DTs by extracting and concatenating all the reviews related to the item.

- *DS ratings*: We use the publicly available Movielens dataset [16] to obtain user-item ratings in the movie domain. The Movielens dataset contains ratings on 1581 movies given by 943 users. The dataset contains only those movies that have been rated by at least 20 active users who have rated at least 20 items. The user-item rating matrix has a sparsity level of 5.23%. Let **M** be the set of selected movies in the movie domain and $\mathbf{U_M}$ be the set of users active in this domain.
- *DS item corpus*: Reviews for each of the movies included in the Movielens dataset are extracted from the imdb.com website. All the reviews pertaining to a given movie are concatenated to form a document that serves as textual review corpus for that movie.
- *DT ratings*: We used the Bookcrossing dataset to obtain user-item ratings for the book domain [17]. The complete Bookcrossing dataset consists of around 1.1 million ratings on 271 379 books provided by 278 858 users on a rating scale of 1–10. All ratings are normalized to fit into a range between 1 and 5. From this dataset, we selected 3156 active users who gave at least 20 ratings on 7472 books. Further, each book is rated by at least 20 users. This filtered dataset has a sparsity level of 0.86%. Let **B** be the set of books in the book domain and $\mathbf{U_B}$ be the set of users active in this domain.
- *DT item corpus*: This step obtains all the reviews for each of the books that are included in the user-item matrix filtered from Bookcrossing dataset, from the amazon.com website. The separate reviews for a given book are combined together to form its textual corpus.

3.2. Preprocessing

An informal style of writing that results in spelling errors and noise is used usually while writing reviews. Textual corpora from these errors and irrelevant content are filtered in module M2. The following steps are incorporated in preprocessing module:

(i) Removal of stop words: Words such as ‘that’, ‘the’, ‘is’, ‘at’, ‘which’ ‘no’, etc. that are very common in use and do not provide any key significance are removed from the textual corpus as they do not enrich the corpus semantically.

(ii) Numeric characters removal and only alphabets are retained: Numeric characters such as ‘1’, ‘23’ and ’@’ are removed as they do not provide any useful information.

(iii) Compression of words: Often, while expressing on social media, people have a proclivity to elongate words that are expressive such as ‘hiiiiiii’ instead of ‘hi’ or ‘happy’ to ‘happpy’. This unit of the segment compresses such words.

(iv) Stemming: Semantic duplicates are transformed to a single originating word so that it is easier to comprehend the reviews clearly. For instance, the words sadness, sadly and sadder all map to the origin word sad.

(iv) Input for topic modeling is prepared after preprocessing the corpus by cleaning noise.

3.3. Topic analysis

A topic model is defined as a statistical model that discovers the hidden semantic structure occurring in a collection of documents. In their paper, Blei *et al.* [18] described LDA as a generative probabilistic model for collections of discrete data. LDA is a three-level hierarchical Bayesian model [5]. In LDA, each item of a corpus is modeled as a finite mixture over an underlying set of topics. Each topic is modeled as an infinite mixture over an underlying set of topic probabilities. Topic probabilities provide an explicit representation of a document [18]. Topic models have a wide range of applications in an area of natural language processing such as multi-document summarization [19], word sense disambiguation [20] and generation of comparable corpora [21].

In our work, the purpose of topic analysis is to measure the preponderance of various topics that users convey through their reviews. The topic analysis process builds the topic distribution profiles for items and users. We represent the variable $\text{Ø}(k,l)$ as the probability of topic distribution of the topic k for any item or user l. For movie domain, $\text{Ø}^M(k,l)$ denotes the topic distribution of k^{th} topic for movie l. Similarly, $\text{Ø}^B(k,l)$ denotes the topic distribution for k^{th} topic number for book l.

A topic W is represented via its N and top-N topic words, ordered by $p(w_i/W)$ where $p(w_i/W)$ denotes the probability of word w_i in topic W. Here, we use the top 10 words based on their probability distribution in a topic. This is so because top 10 words can give sufficient detail about the topic to distinguish one topic from another. After topic modeling, we get:

(i) The distribution of each topic in a movie as shown in Table 1 and

(ii) the top 10 words with their probability in a topic as shown in Table 2.

3.4. Profiling the movie domain

In order to analyze the movie domain, the DS topic analysis module M3 first builds the topic distribution-based profiles of all movies included in the Movielens dataset. Next, it uses these

TABLE 1. Snapshot of topics showing topic distribution in different movies.

Movie title	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
The Usual Suspects (1995)	0.059	0.112	0.184	.0631	0.0329	0.127
The Thin Man (1934)	0.172	0.135	0.046	0.050	0.042	0.069
The Wild Bunch (1969)	0.025	0.016	0.016	0.045	0.152	0.052
Touch of Evil (1958)	0.227	0.015	0.039	0.032	0.015	0.016
Under Siege (1992)	0.049	0.037	0.137	0.021	0.074	0.071

TABLE 2. Snapshot showing top 5 words out of 10 for each topic.

Topic	Word 1	Word 2	Word 3	Word 4	Word 5
Topic 1	Murder: 0.026	Suspect: 0.017	Crime: 0.014	Case: 0.014	Killer: 0.011
Topic 2	War: 024	Group: 0.020	Power: 0.017	Involve: 0.014	Fight: 0.014
Topic 14	Team: 0.063	Dog: 0.0232	Earth: 0.022	Investigate: 0.018	Mission: 0.018
Topic 17	Friend: 0.0304	Leave: 0.021	Relationship: 0.016	People: 0.015	Well: 0.015

item profiles and combines them with the users' ratings to build the topic distribution-based profiles of users.

3.4.1. *Topic distribution of movie items*

For profiling the topic distribution related to a movie l, the textual corpus $R^M(l)$ is input to the LDA-based topic model. This module outputs the movie's topic distribution profile:

$$\O^M(l) = \left\{\O^M(1,l), \O^M(2,l), \O^M(3,l), \ldots \O^M(20,l)\right\}, \tag{4}$$

where $\O^M(k,l)$ is the k^{th} topic distribution of movie item l.

3.4.2. *Topic distribution of movie users*

This step computes the user-based topic distribution profile, using item-based topic distributions as the weighted sum of topic distributions of items rated by the user. We assume that the users' ratings indicate their overall assessment of the latent semantic structure of items.

Consider a user j who is active in the movie domain. Let $r(j,l)$ denote the rating given by user j to movie l. The rating-weighted topic distribution $\O r^{MU}(k,j)$ for topic k by user j is given by

$$\varnothing r^M(k,j) = \frac{\sum_l \varnothing^{\mathrm{M}}(\mathrm{k,l}) \times \mathrm{r}(\mathrm{j,l})}{\sum_l \mathrm{r}(\mathrm{j,l})}. \tag{5}$$

Thus, topic distribution profile for user j in the movie domain is given by

$$\O r^M(j) = \left\{\O r^M(1,j), \O r^M(2,j), \ldots, \O r^M(20,j)\right\}. \tag{6}$$

3.5. Profiling the book domain

The DT topic distribution analysis module M4 develops topic distribution profiles for the items and users of the book domain.

3.5.1. *Topic distribution of book items*

For profiling the topics related to a book l, the textual corpus book $R^B(l)$ is input to the topic modeling step in M3. This module outputs the book's topic distribution-based profile $\O^B(l)$,:

$$\O^B(l) = \left\{\O^B(1,l), \O^B(2,l), \ldots \O^B(20,l)\right\} \tag{7}$$

where $\O^B(k,l)$ is the k^{th} topic distribution of book item l.

3.5.2. *Topic distribution of book users*

Similar to movie users, the topicwise distribution profile of an active user is computed by aggregating the rating-weighted topic profiles of all the items rated by them. Let $r(j,l)$ denote the rating by user j to book l. The rating-weighted topic based $\O r^B(k,j)$ for topic k is given by

$$\varnothing r^B(k,j) = \frac{\sum_l \varnothing^{\mathrm{B}}(\mathrm{k,l}) \times \mathrm{r}(\mathrm{j,l})\}}{\sum_l \mathrm{r}(\mathrm{j,l})}. \tag{8}$$

The DT user topic distribution for user j is given by

$$\O r^B(j) = \left\{\O r^B(1,j), \O r^B(2,j), \ldots \O r^B(20,j)\right\}, \tag{9}$$

where $\O^B(k,j)$ is the k^{th} topic distribution of book user j.

3.6. Compute topic coherence between topics

Module M5 computes the topic coherence between topics of source and DTs. Topic coherence matrix C is a 20×20 between all pairs of topics in domain DS and DT. Each element of the matrix is denoted by $\mathrm{C(p,q)}$ where p is the topic from domain DS and qis the topic in domain DT. For finding topic coherence$\mathrm{C(p,q)}$ between two topics, module M5 of TC-CDR first concatenates the topic words from the two topics. Topic coherence matrix is built using PMI and ESA topic coherence methods, as explained in Subsection 2.1 and 2.2.

3.7. Translated user profile in DT

Module M6 finds the translated user profile of user u in DT T using topic coherence between the topic words of DS. For computing this translated user profile, consider a user u that is active in the DS with his/her topic distribution profile$\text{Ø}r^M(u)$as given in Equation 6. Now, the translated topic distribution profile for user u based on topic coherence and topic distribution of topic q is $\text{Ø}t^B(u)$ given by

$$\left(\varnothing t^B\left(j,u\right)\right) = \frac{\sum_{\mathrm{i}=1}^{20} p\left(\varnothing r^M\left(\mathrm{p},u\right)\right) \times C\left(q,p\right)}{\sum_{\mathrm{i}=1}^{20} C\left(q,p\right)}. \quad (10)$$

The topic distribution of translated profile for user u is given by

$$\left(\text{Ø}t^B(u)\right) = \left\{\left(\text{Ø}t^B\left(1,u\right)\right), \left(\text{Ø}t^B\left(2,u\right)\right), \ldots, \left(\text{Ø}t^B\left(20,u\right)\right)\right\}. \quad (11)$$

3.8. Executing TC-CDR

Let us consider a user u who is active in a DS and seeks recommendations from a DT. The TC-CDR system uses the topic distribution profile $\text{Ø}r^M(j)$ of a user in the DS. Then, using topic coherence score between the topic words of the topics from both domains, it generates translated profile $\text{Ø}t^B(u)$ of the user u. Equivalent topic distribution profiles of users in DT that are similar to the translated profile of u are found. It then performs collaborative filtering to recommend target items.

3.8.1. *Finding similar users across domains*

Module M7 calculates the distance between the given translated user profile $\text{Ø}t^B(u)$ for a user u and topic distribution profiles of other users in DT. We adopt the symmetric KLD to measure the distance between the topic distributions of translated profile and DT topic distribution profiles. The standard KLD is not symmetric [15]. Therefore, it is not strictly a distance metric. The symmetric KLD measures the distance between two probability densities $p(.)$ and $q(.)$ of the form

$$\mathrm{D}(p\,||q) = \sum_x ((p(x) - q(x))\ln\frac{p(x)}{q(x)}. \quad (12)$$

Accordingly, the parameter $\Delta_{u,v}$ that is the symmetric KLD between the topic distributions of user u active in the movie domain and user v active in the book domain is given by

$$\Delta_{u,v} = KLD((\varnothing t^M(u)), (\varnothing r^B(v)))$$
$$= \sum_{k=1}^{M} ((\varnothing t^M(k,u)) - (\varnothing r^B(k,u)))))\ln\frac{p(\varnothing t^M(k,u))}{p(\varnothing r^B(k,v))}. \quad (13)$$

It can be seen that the lower the value of $\Delta_{u,v}$, the greater the similarity between the topic distributions profiles of users u and v.

3.8.2. *Generating recommendations*

The cross-domain recommender module M8 generates recommendations from the DT to a user in the DS. The pseudo-code CDR_CF(.) in Fig. 2 gives the steps involved in obtaining the recommendations. Given a candidate user u in DS, the process returns a set K of items from the DT, which is recommended to user u.

The process begins by initializing K to null (line 1) and evaluating the topic distribution-based profile $\text{Ø}r^M(u)$ of user u from DS (line 2). Next, it computes topic coherence matrix between each pair of topic from both domains (line 3). Using Equation (11), it computes translated user profile $\text{Ø}t^B(u)$ (line 4). Next, it evaluates the topic distribution-based profiles of all users in the DT. Given a user v in the DT, CDR_CF calculates KLD $\Delta_{u,v}$. It arranges all users in the DT in ascending order of $\Delta_{u,v}$ (line 9) to yield U_{TN}, the top N users with similar profiles (line 9). Next, it performs collaborative filtering to predict ratings for user u for items in DT by using Resnick's formula [22] (lines 11–16):

$$r'(u,i) = \bar{r}(u) + \frac{\sum_{v\in U_{TN}}(r(v,i) - \bar{r}(v))}{\sum_{v\in U_{TN}}\Delta_{u,v}} \times \Delta_{u,v}. \quad (14)$$

Here, $\bar{r}(u)$ is the average rating of user u for all their rated items in DS, $\bar{r}(v)$ is the average rating of user $v \in U_{TN}$ for all their rated items in DT and $r(v,i)$ is the actual rating given by user v to item i. If the predicted rating$r'(u,i)$ is 4 or 5, the process adds item i to the set of recommended items K (line 14). The process finally returns the set K of recommended items (line 17).

3.9. Evaluating the performance of TC-CDR

Module M9 conducts a performance evaluation of the TC-CDR system. In the case of movie and book domains, their respective datasets Movielens and Bookcrossing have no common users. Therefore, it is not possible to directly assess the accuracy of cross-domain recommendations. We adopt an approach suggested in [8] to evaluate the system performance.

Module M9 first recommends items in target that is book domain to the user from source or movie domain. The module using the recommended items as input to TC-CDR approach

```
CDR_CF(.)
Input: User $u$, Source Domain DS, Target Domain DT
Output: Set of Recommended Items $K$
Initialize:
K= $NULL$
Evaluate source domain candidate user's topic distribution $\emptyset r^{M}(u)$ using eq 6
For each pair of topics $p \in \boldsymbol{W}$, $q \in \boldsymbol{W}$, compute Topic Coherence $\boldsymbol{C}(\boldsymbol{p}, \boldsymbol{q})$
Find translated user profile $\emptyset t^{B}(u)$ of $u$ in target domain using Eq. 11
// Find similarity with target domain users:
for each user $v \in U_{DT}$ do
    Evaluate rating-weighted topic distribution $\emptyset r^{B}(v)$ using equation 9
    Find KL Divergence $\Delta_{u,v}$ using Eq. 13
end for
// Find Top N similar users in target domain:
Sort users in $U_{DT}$ in ascending order of $\Delta_{u,v}$
Set $U_{TN} = TopN$ users in sorted $U_{DT}$
// Predict candidate user's ratings and recommend:
for each user $z \in U_{TN}$ do
    for each item $I$ from DT that $z$ has rated do
        Calculate corresponding predicted rating $r'(u, i)$ for
        $u$ using Eq. 14
        If ($r'(u, i) \geq 4$ ) then $K = K \cup i$
    end for
end for
return $K$
```

FIGURE 2. Generating cross-domain recommendations using topic distributions of users.

again makes reverse predictions for the original DS. By comparing these reverse predictions with original preferences, the module evaluates the performance of TC-CDR approach.

4. EXPERIMENTAL RESULTS

This section discusses the experimental evaluation of the proposed TC-CDR approach. We first describe the setup for the experiments in Section 4.1. Then, we present the results of experiments on using Movielens and Bookcrossings datasets. Section 4.2 analyzes the relationship between ratings and topic distribution profiles of movie items. Section 4.3 analyzes the relationship between ratings and topic distribution profiles of book items. We evaluate the results of our proposed TC-CDR approach based on different topic coherence measures between topic words of items across domain based on PMI and ESA in Section 4.4. Section 4.5 compares our TCDR with the two existing approaches, namely cosine based (CD) and SCD [8].

4.1. Experimental setup

For topic modeling of the review corpus of each item, we employ MALLET. MALLET is a Java-based package under open source software that is used for clustering, classification, topic modeling, information extraction, etc. [23]. After the data preparation module, the whole textual review corpus for each item in a given domain forms a single MALLET format file. Then, a topic model is created that gives two output files as shown in Tables 1 and 2.

For finding topic coherence$C(p, q)$ between two topics p and q using PMI, module M5 of TC-CDR first concatenate the topic words from the two topics. Then, it uses the Palmetto library developed by Roder *et al.* [24] to measure topic coherence between the topics. Palmetto library offers different coherence types such as normalized pointwise mutual information (NPMI), UCI and UMass [25]. For our work, we use UCI as topic coherence. The word co-occurrence counts are derived using a sliding window of all word pairs of the given topic

TABLE 3. Performance of topic-based distribution for movies dataset.

S.No	Accuracy	Precision	Recall	F-measure
KNN	0.91	0.91	0.99	0.95
Decision tree	0.84	0.90	0.89	0.90
Random forest	0.91	0.91	0.99	0.95
Bayes classifier	0.74	0.90	0.76	0.83

words as proposed in [1]. Finally, the arithmetic mean of all PMI values is computed to get the overall coherence between two topics.

Similar to PMI, after concatenating topic words from the two topics, we compute topic coherence using ESA. We make use of open source software implemented in Java, using Lucene for indexing [13].

4.2. Relationship between ratings and topic distribution profiles of movie items

For this experiment, we considered all active users who rated at least 50 movies. In this experiment, we examine whether users' ratings indicate their acceptance or rejection of topic-based distribution profiles of books rated by them. For each user, we grouped their ratings into two classes: like and dislike. Movies with ratings 4 or 5 were labeled as like and those with ratings 1 or 2 ratings were labeled as dislike. Movies with rating 3 were not considered as it indicates a neutral opinion. Using the topic distribution of rated movies as features, we trained different classifier taking 80% of dataset as training and 20% as testing dataset. We evaluated our approach with 10-fold cross-validation to predict the labels (like or dislike) of test cases using different machine learning classifiers such as k nearest neighbor (KNN), decision trees, random forest and Bayes classifier.

Table 3 shows the results of the classifiers taking movies as a dataset. Random forest and KNN have the highest F-measure of 0.95 followed by the decision tree and Bayes classifier. Precision is 0.91 for KNN and random forest and 0.90 for the decision tree and Bayes classifier. The results validate our contention that statistically speaking, a typical user's ratings do reflect their liking or disliking for the underlying topic distribution profiles of movies rated by them.

4.3. Relationship between ratings and topic distribution profiles of book items

Proceeding for books in the same way as for movies, we computed the prediction accuracy for all eligible users in Bookcrossing dataset by using the topic distributions of books as classification features. Table 4 shows the results of the classifiers taking books as a dataset.

TABLE 4. Performance of topic-based distribution for books dataset.

S.No	Accuracy	Precision	Recall	F-measure
KNN	0.81	0.83	0.95	0.885
Decision tree	0.72	0.83	0.76	0.79
Random forest	0.80	0.82	0.93	0.87
Bayes classifier	0.62	0.84	0.72	0.78

TABLE 5. Prediction performance of TC-CDR for top-N recommendations.

TC-CDR top N	Precision	Recall	F-measure
CD top 5	0.2962	0.00026	0.0005
CD top 10	0.1318	0.001037	0.0005
CD top 15	0.112	0.00031	0.0006
PMI top 5	0.509	0.00677	0.01337
PMI top 10	0.3757	0.00779	0.01527
PMI top 15	0.256	0.00862	0.0167
ESA top 5	0.65957	0.00453	0.00900
ESA top 10	0.47333	0.00560	0.0110
ESA top 15	0.345	0.00657	0.0129

As can be seen from the table, KNN gives the highest F-measure of 0.885 and accuracy of 0.81, respectively, followed by the random forest classifier with an F-measure of 0.87 and accuracy of 0.80. These results give credence to our assumption that users' ratings reflect their overall assessment of the topic distribution-based profile of books rated by them.

4.4. Evaluating the performance of TC-CDR

The results of experiment based on TC-CDR approach taking movies as DS and books as DT are presented in Table 5. Table 5 shows the effect of top-N recommendations on precision, recall and F-measure using PMI and ESA as topic coherence measures. CD finds cosine similarity between topic words of topic p in DS with topic words of topic q in DT. Figure 3 depicts the variation of precision with N for top-N recommendations and Fig. 4 depicts the variation of recall with N for top-N recommendations. With an increase in the number N for top-N recommendations, recall increases steadily while precision decreases. As the number of recommendations increase from 5 to 15, for PMI recall increases by 27% while precision drops by as much as 98%. For ESA, as the number of recommendations increase from 5 to 15, recall increases by as much as 45% while precision drops by 48%. This shows that, as more recommendations are provided to the user, there is a tradeoff between the precision and recall. Due to the participation of more users in the recommendation process, a user gets more relevant choices of their liking although the percentage of relevant ones among all choices decreases.

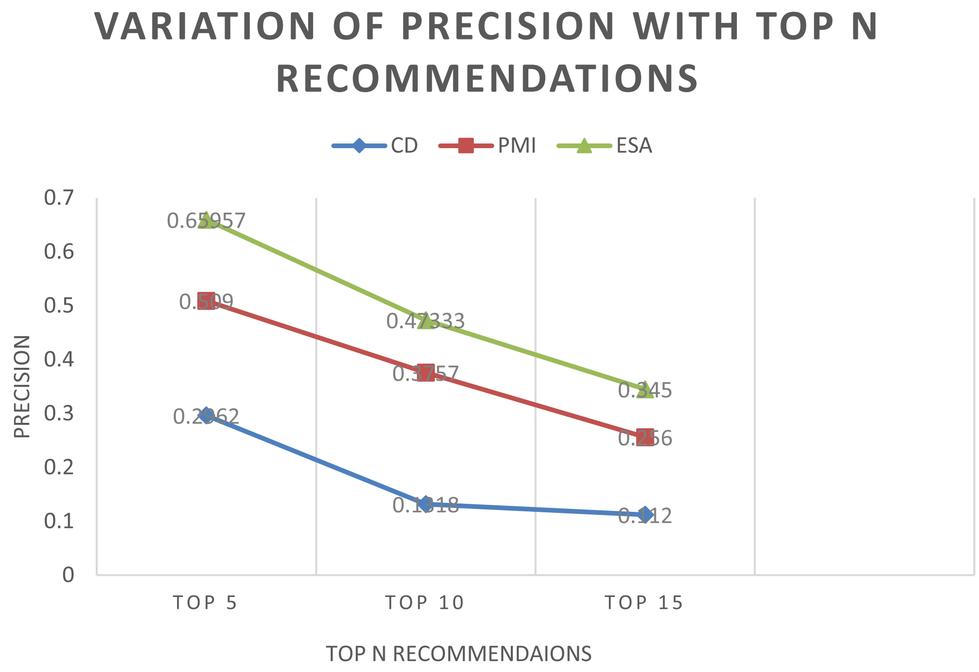

FIGURE 3. Variation of precision with N for top-N recommendations in TC-CDR.

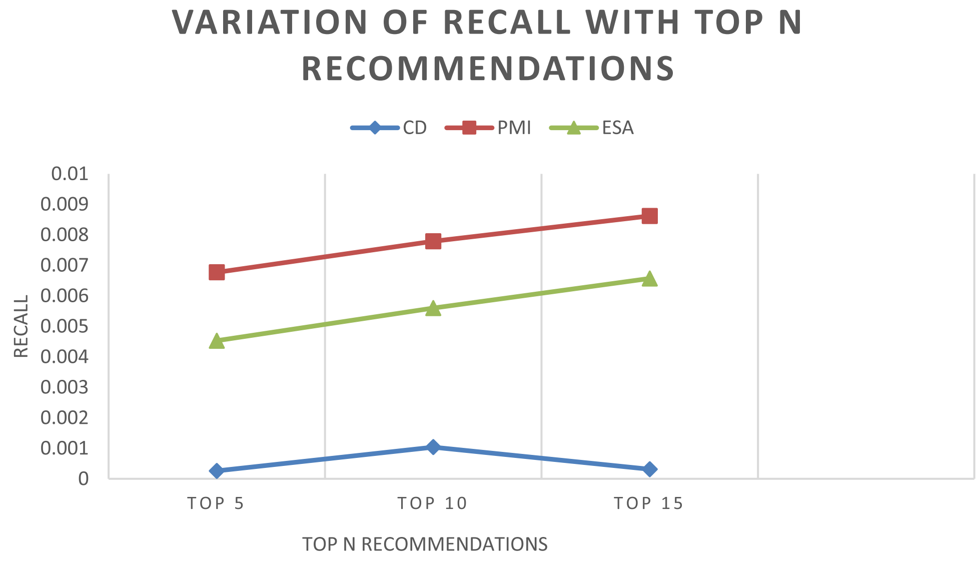

FIGURE 4. Variation of recall with N for top-N recommendations in TC-CDR.

4.5. Comparison with recent work

The SCD approaches reported in [8] apply topic modeling on reviews of movies and books for performing CDR. While the SCD defines topics as distributions over word clusters and applies cosine distance in order to capture the inherent similarities between the vocabularies of multiple domains, our proposed TC-CDR approach applies new semantic topic coherence measures PMI and ESA between the topics of multiple domains. Table 6 compares the precision, recall and F-measure gathered from an analysis of our experimental results with the corresponding values reported for the SCD approach for top 10 recommendations. On comparing PMI with CD, we find that PMI yields a precision of 0.376 that is 184.8% higher precision than the precision of CD at 0.1318 more than 3-fold gain. Similarly, on comparing ESA with CD, we find that ESA

TABLE 6. Performance of CD, SCD and TC-CDR for top 10 recommendations.

Metric	CD	SCD	PMI	ESA
Precision	0.1318	0.3064	0.3758	0.4733
Recall	0.1037	0.1166	0.00779	0.0056
F-Measure	0.116	0.1689	0.01527	0.0110

yields a precision of 0.473 that is 259.1% higher precision than the precision of CD at 0.1318 that is more than 4-fold.

Comparing with SCD, PMI results show 22.6% increase in precision with SCD precision of 0.3064. ESA yields an increase of 54.4% in precision compared with SCD approach. Recall of PMI is <93% than SCD. The recall of ESA at 0.0056 is 95% less than SCD. Results clearly show that ESA has the highest precision followed by PMI. CD has the least precision. However, this comes at the cost of a lower recall.

From the above observations, we infer that TC-CDR approach based on ESA performs better than cosine, SCD and PMI in terms of precision. This is because both PMI and ESA use rich semantic space provided by Wikipedia instead of cosine similarity, which is just based on topic similarity itself. Thus, our approach using ESA and PMI using symmetrical Kullback–Leibler divergent method perform better than existing SCD. We conclude that ESA provides a better bridge between the two domains using Wikipedia as common space for knowledge transfer followed by PMI.

5. CONCLUSION

In this paper, we elaborated upon our proposed topic coherence-based CDR approach that exploits user-generated data in the form of reviews. We first used topic modeling to find probability distribution of topics in the given individual corpora of items in both source and DTs. Computing semantic relatedness of natural language texts from both domains requires access to vast amounts of common sense and domain-specific world knowledge. So we used (i) ESA that represents the meaning of texts in a high-dimensional space of concepts

derived from Wikipedia and (ii) PMI that use Wikipedia as reference corpus to count word co-occurrences centered on a topic word. Then using CF, cross-domain recommendations are generated. Using ESA and PMI for computing topic coherence between topics of different domains, we showed that these methods provide improvements over precision for recommendations across domains compared with topic matching using cosine similarity.

REFERENCES

[1] Newman, D., Lau, J.H., Grieser, K. and Baldwin, T. (2010) Automatic Evaluation of Topic Coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 100–108. Association for Computational Linguistics, Los Angeles, CA, USA.

[2] Newman, D., Karimi, S. and Cavedon, L. (2009) External evaluation of topic models. *Australasian Doc. Comp. Symp*, 2009, 11–18.

[3] Sriurai, W. (2011) Improving text categorization by using a topic model. *Adv. Comput.*, 2, 21.

[4] Fang, A., Macdonald, C., Ounis, I. and Habel, P. (2016) Topics in Tweets: A User Study of Topic Coherence Metrics for Twitter Data. In *European Conference on Information Retrieval*, pp. 492–504. Springer, Cham.

[5] Wang, C. and Blei, D.M. (2011) Collaborative Topic Modeling for Recommending Scientific Articles. In *Proc. 17th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Diego, California, pp. 448–456. Association for Computing Machinery, New York, NY, USA.

[6] Tang, J., Wu, S., Sun, J. and Su, H. (2012) Cross-Domain Collaboration Recommendation. In *Proc. 18th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Beijing, China, pp. 1285–1293. Association for Computing Machinery, New York, NY, USA.

[7] Low, Y., Agarwal, D. and Smola, A.J. (2011) Multiple Domain User Personalization. In *Proc. 17th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Diego, California, pp. 123–131. Association for Computing Machinery, New York, NY, USA.

[8] Kumar, A., Kumar, N., Hussain, M., Chaudhury, S. and Agarwal, S. (2014) Semantic Clustering-Based Cross-Domain Recommendation. In *2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pp. 137–141. IEEE ,Orlando, FL, USA.

[9] Aletras, N. and Stevenson, M. (2014) Measuring the Similarity Between Automatically Generated Topics. In *Proc. 14th Conf. European Chapter of the Association for Computational Linguistics, Volume 2: Short Papers*, pp. 22–27. Association for Computational Linguistics, Gothenburg, Sweden.

[10] He, Q., Chen, B., Pei, J., Qiu, B., Mitra, P. and Giles, L. (2009) Detecting Topic Evolution in Scientific Literature: How Can Citations Help? In *Proc. 18th ACM Conf. Information and Knowledge Management*, pp. 957–966. Association for Computational Linguistics, Gothenburg, Sweden.

[11] Ramage, D., Hall, D., Nallapati, R. and Manning, C.D. (2009) Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-Labeled Corpora. In *Proc. 2009 Conf. Empirical Methods in Natural Language Processing* (Vol. 1), pp. 248–256. Association for Computational Linguistics, Gothenburg, Sweden.

[12] Mu, R. (2018) A survey of recommender systems based on deep learning. *IEEE Access*, 6, 69009–69022.

[13] Gabrilovich, E. and Markovitch, S. (2007) Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *IJcAI*, 7, 1606–1611.

[14] Gabrilovich, E. and Markovitch, S. (2009) Wikipedia-based semantic interpretation for natural language processing. *J. Artif. Intell. Res.*, 34, 443–498.

[15] Johnson, D.H. and Sinanovic, S. (2001) Symmetrizing the Kullback–Leibler distance. Technical Report. Rice University.

[16] Harper, F.M. and Konstan, J.A. (2016) The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5, 19.

[17] Ziegler, C.N., McNee, S.M., Konstan, J.A. and Lausen, G. (2005) Improving Recommendation Lists Through Topic Diversification. In *Proc. 14th Int. Conf. World Wide Web*, Chiba, Japan, pp. 22–32. Association for Computing Machinery, New York, NY, USA.

[18] Blei, D.M., Ng, A.Y. and Jordan, M.I. (2003) Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3, 993–1022.

[19] Haghighi, A. and Vanderwende, L. (2009) Exploring Content Models for Multi-Document Summarization. In *Proc. Human Language Technologies: The 2009 Annual Conf. North American Chapter of the Association for Computational Linguistics*, pp. 362–370. Association for Computational Linguistics, Boulder, CO, USA.

[20] Boyd-Graber, J., Blei, D. and Zhu, X. (2007) A Topic Model for Word Sense Disambiguation. In *Proc. 2007 Joint Conf. Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 1024–1033. Association for Computational Linguistics, Prague, Czech Republic.

[21] Griffiths, T.L. and Steyvers, M. (2004) Finding scientific topics. *Proc. Natl. Acad. Sci.*, 101, 5228–5235.

[22] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. (1994) GroupLens: An Open Architecture for Collaborative Filtering of Net News. In *Proc. 1994 ACM Conf. Computer Supported Cooperative Work*, pp. 175–186. ACM, Chapel Hill, NC, USA.

[23] McCallum, A.K. (2002) Mallet: A machine learning for language toolkit. 424–433. http://mallet.cs.umass.edu/

[24] Röder, M., Both, A. and Hinneburg, A. (2015) Exploring the Space of Topic Coherence Measures. In *Proc. Eighth ACM Int. Conf. Web Search and Data Mining*, Shanghai, China, pp. 399–408. Association for Computing Machinery, New York, NY, USA.

[25] Pecina, P. and Schlesinger, P. (2006) Combining Association Measures for Collocation Extraction. In *Proc. COLING/ACL on Main Conf. Poster Sessions*, pp. 651–658. Association for Computational Linguistics, Sydney, Australia.