International Journal of Engineering & Technology, 7 (4.39) (2018) 105-108

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

An Innovative Technique based on Topical Modeling for Personalized Movie Searched Engine

Lokendra Birla[1]*, Rohit Dhangar[2], Anand Kumar[3], Yogesh Singh[4]

[1]*Deparment of Computer Science and Engineering, Siksha 'O' Anusandhan (Deemed to be University), Odisha, India*
[2,3,4]*IIT Kharagpur, India*
**Corresponding author E-mail: lokendrabirla@soa.ac.in*

Abstract

A large an amount of information is becoming available and this information is valuable source of intelligence but it is very difficult to extract or mine useful information for making decision. To overcome this type of problem, there are information filtering systems, such as the personalized movie searched engine (PMSE). This PMSE identifying interest of person on his/her preferences. This PMSE utilized user's reviews to create a personalized profile of user. First of preprocess the reviews then use Latent Dirichlet Allocation (LDA) for generating topic from user reviews. Finally, Dual Personalized Ranking function (D-PR) [1] will rank movies when the user enters a query. Experiment result shows that our system is able to give personalized movie search results to the user.

Keywords: *Personalized movie search engine(PMSE), Latent Dirichlet Allocation (LDA), Dual Personalized Ranking function (D-PR).*

1. Introduction

With the arrival of so many movie streaming services, thousands of movies are available to the user by just a click of button. And movies are an essential source of entertainment for everyone to relax the mood and re-energize. Today we have movie from Hollywood, Bollywood, Documentaries and International films, etc. With so many options available, consumers face the conundrum of what they should watch. This is where the personalized movie search plays the role. The advantage of a personalized search is that it allows to tailor and fine tune search results according to individual preferences [4, 5]. Penalization is important in other ways as well as it helps to improve the search performances, and it has also grabbed quiet an attention from data mining and information retrieval communities [6, 7]. In order to give precise personalized search result to the user, it is important to understand the movie first. And for the system to understand
the movie it will use movie features (mood, review, genre, keywords, etc). Using these features a system can understand and classify a movie under different categories. Among these features movie review can be used as the key features for movie personalized search system. As movie review contain essential information about movie in a summarized and precise form. It contains significant information about lead characters, important scene description and location etc. It also contain the information about the users views and opinion about a movie. So, it can be used to outline a user profile [9, 10]. As the reviews are written by the user so, it contains both useful (keywords) and useless (stop words). So, before using the user reviews as the input to the system, some prepossessing needs to be done to extract keywords. It involves steps like stemming, stop words removal etc. Then these key words can be used as the input to the Latent Dirichlet Allocation model in order to generate topics [18] which will be used later to find movies that user may like, and ranks them. The experiment is done on a small data-set to show how personalized movie search system is important.

1.1 Problems

1. Can features of movie, extracted from user reviews?
2. Can the extracted features be used to classify movies into different topics/themes?
3. Can the system find similar movies in accordance to the user profile?
4. How to rank different movies according to user interest.

1.2 Objective

1. To design a system that gives personalized search result to the user.
2. To categorize movies under different topic based on extracted features.

2. Related Work

In the research activities in the last decade [11, 12, 13, 14] movie data is in the form of dialog, script, keywords, user review, etc. Chen et al. [11] combines the model based collaborative filtering along with heterogeneous information network in order to form a efficient recommendation model. Bougiatiotis et al. [12] illustrates the extraction of multi modal representation model for movies, based on textual in- formation from subtitle, along with some cues from audio and video channel. Ahn et al. [13] presents a movie recommendation system which utilizes cultural meta data like plot, user rating, user reviews, genres and keywords and shows that user comments and genres give high precision. We choose to use reviews written by the users because:
i. Movie reviews are easily available.
ii. Each review can be considered as a document which represents the movie as a whole.
iii. Reviews being directly written by the user so it gives users thoughts about a movie.

All the reviews can be combined into a single document, it can be used to find semantic pattern in movie level. For movie reviews to be used as data, it is important to remove html tags, irrelevant words, symbols, etc. There are many open source libraries and toolkit which are easily available, and used as primary step in any type of kind of text processing [15, 16]. Text filtering not only removes unwanted and unnecessary noise form the data, but it also allows us to use complex mathematical models. It is necessary to remove noise and perform preprocessing as it can affect the result of any type of Natural Language Processing(NLP) based model. After performing preprocessing of data, generating a term-document matrix is an essential step for extracting feature. Term-document matrix is used as input for many semantic analysis techniquelike performing tf-idf scheme to complex models such as LSA, PLSA and LDA etc. Topic models are used to uncover the hidden thematic structure i.e. discover the topicswithin document. Latent Dirichlet Allocation (LDA) is a popular topic model, it performs well in sentiment analysis as well as in different types of personalized recommendation system and text categorization, etc. Then its result is inputted to the LDA model for sentiment analysis. Zhao et al. [19] proposes a Latent Dirichlet Allocation (LDA) based topic modeling method named as Hashtag LDA, which unearths latent information and utilizes it for personalized hashtag recommendation in microblog environment. Here the users are represented by user topics distribution then calculate the frequencies of all hashtags gathered from these user and then recommends the most relevant hashtag to the user. The input is the dataset of reviews along with the ordinal rating of thousands of movie for examining the different features of collaborative filtering. The result of the paper shows that LDA-based extraction of movie aspect yield better result than candidate extraction and clustering. Xu, et al. [1] presents a framework for search engine. The web today is growing rapidly, and it has turned into a fundamental and essential thing for people to access as well as convey information. Finding relevant information from this broad repository of data is not easy especially for learners. As learners presently depend on the traditional search engine to retrieve relevant learning material from the huge amount information present in the Web. So, to address these problems this paper proposes a framework that elevates the existing Google search engine with the capability to filter its search results by considering the learners educational background, their learning behaviors while using the tool. The rapid growth of Web services makes it tough for users to choose among numerous Web services available. Thus, the selection and ranking of Web services is a challenge for the Web service community. Xu et al. [1] proposes dual personalized ranking (D-PR) function, presently in order to understand and identify a users taste social summary of a document, the user profile and general document profile is extensively used in the existing system. But in the real world scenario, using only these two profiles is not enough to effectively personalize the search results on Social Web. There are two major issues with previous personalized ranking methods: i) The general document profile that is used for ranking is not enough to summarize a specific users personal perception about a document as in such a generalized profile the tags which are given by all users are assigned equal weightage ii) The information present in a users profile alone is not sufficient enough to characterize a users preference. So, to solve these problems D-PR was proposed it introduces two novel profiles personalized document profile and extended user profile. Personalized document profile is the perception of a specific user about a document and extended user profile is the sum up of all personalized document profile. It uses a bag of wordsapproach, where every document to be bag of words which has no particular structure other than word and the topic statistics. The simple idea of LDA is that it attempts to imitate how the writing process is. So, it simply generates a document on the topic that is given. It gives us the theme or the topic running through a corpus. The application of this model is in the problems in collaborative filtering, text classification and text modeling. With topic modeling methods like LDA, excavating semantic similarities from textual movie features becomes easier. The semantic knowledge which is extracted by such way can be further processed and used to rank top movies which are most relevant, according to the query entered by the user.

3. Experiment

In this paper we will discuss about the different major steps involved in implementation of the designing a personalized movie search system which gives results to the user biased on the past reviews of the user. Figure 3.1 below shows our proposed model. Outlining of the system : 1. An IMDB dataset is taken and for this project work. The dataset consists of 4430 movies of the last few decades, reviewed and rated by 2585 user on web. 2. Firstly the movie reviews given by different users are extracted. 3. Then preprocessing is done on the extracted review. 4. An LDA model is trained in order to get the topic-distribution over movies. 5. User to topic mapping and movie to topic mapping is done. 6. Then Social similarities, topic similarity and Penalization similarity is found and final ranking results are computed and accordingly result are displayed to the user.In order to implement the above steps in the system, python is chosen as the programming language because of the huge number of Machine Learning (ML) tools and large standard libraries associated with it. It also has automatic memory management and other dynamic features. It also support multi programming paradigm. We will begin with a brief description of the movie corpus followed by a elaborate discussion on the different steps involved in data preprocessing.

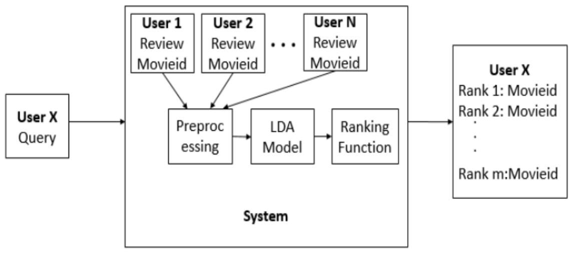

Figure 1: Proposed Model

Next, is the features are extracted from processed data. Then topic modeling is done followed by mapping which is later used in ranking movies for the user.Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

3.1. Movie Corpus

In order to rank and give personalized results to the user we first need an appropriate dataset or corpus on which we will work. We need a corpus which contains the required data i.e movie review from immense number of users. Wealso need a wide variety of movies of different generes, to check if the system is efficiently giving appropriate results to the user. In this paper we will be working with one of the well-known movie database IMDB. IMDB also known as Internet Movie Database consists of information related to all kinds of films, television program, video games and other internet videos. This online database consists of all sorts of data including cast, crew, fictional character bibliographies, plot, rating, user reviews etc. Our corpus is in json format it can be expanded as Java Script Object Notation it is a syntax to store and exchange data. It has a total of 127503 entries, the movie corpus has data about 4468 movies, reviewed by over 2585 users. So it is a vast dataset. Every single entry in the corpus consists of the following: 1. user: it stores the individual user id for every user 2. movie: it stores the movie id of every movie 3. title: stores the title of the review 4. rating: stores the rating given by the user to the movie 5. review: stores the review written by user about the movie, his/her opinion and thoughts about the movie. So, from the movie corpus the required data can be taken and feature extraction can be performed. In this project we use user written reviews to understand a users interest so we extract review and title i.e. title of the review from the movie corpus. Further preprocessing will be done on the extracted features.

3.2. Text Preprocessing

A corpus in Natural Language Processing is a large collection of texts which mainly serves in verifying a hypothesis about a language. Like example extraction of feature from text or finding different pattern of word usage. Before applying any kind of model on the text data, it requires daa preparation. This data preparation is done by using data preprocessing. In this paper after assembling the user reviews from corpus different data preprocessing is performed, it involves steps like tokenization, removal of stop words and punctuation, stemming, changing the tokens into lower case, etc. Each of this step is elaborated below. In this paper the words data preprocessing and text preprocessing are used interchangeable. 1. Tokenization. 2. Removal of stop words, alphanumerical, punctuation. 3. Stemming. 4. Change to lower case. The preprocessing is done by importing the nltk i.e the natural language toolkit package the input is the review file, then different steps involved in preprocessing will be done. The conversion of the words into lower case will be done using the lower() function. While tokenization, removal of stopwords and stemming is done by creating the objects of RegexpTokenizer, Stopwords and PorterStemmer respectively. The output of this will be processed tokens stored in form of dictionary, corpus followed by tf-idf and tf models ready to be used as the input to the LDA model for generating topics. The purpose of performing preprocessing before using the data is to reduce the size of the target corpus by removing unwanted and irrelevant noise which in turn will increase the efficiency and effectiveness of the system [5].

3.3. Topic Model Generation

The preprocessing step is followed up by Topic model generation, before getting into its details lets begin with the simple idea of the What is a Topic?. A topic usually consists of a cluster or group of words which are semantically related and so can be grouped under a common topic. Topic model gives a superficial view of the theme or the topics present in the dataset. It is a heuristic tool which can be used in sorting large corpus of text. Topic modeling finds the pattern in the words, of a collection of document and categorize them into different topics. There are different types of topic models available like Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA), etc. In this project we will be be using Latent Dirichlet Allocation as it is simple and most widely used topic modeling methodology. We will generate LDA model by importing the gensim package present in python which is widely used for topic modeling. The corpus and the dictionary generated previously will be used as the inputs. The LDA model will be given with the corpus, the number of topics to be generated, the dictionary and the number of passes i.e. the number of iterations we want the model to run. The output will be the LDA model for the corresponding number of topics and passes. We generated the LDA model for 30 topics and 50 passes. LDA represents every documentto be composed of a mixture of topics[6]. So, it is appropriate in analyzing the topics distribution within the reviews.

3.4. Mapping

After the generation of topics by using LDA model the next step is to perform mapping. Two types of mapping will be done. 1. User to topic mapping 2. Movie to topic mapping In user to topic mapping, a mapping is done between every user file that contains all the reviews written by the that user to the topic model generated by the LDA model on reviews. So, the different topics associated with every user is understood and analyzed. In Movie to topic mapping, a mapping is done between the every movie file that contains all the reviews about a particular movie, by all users to the topic model generated by LDA model on reviews. So, the different topics associated with every movie is analyzed.For the user to topic mapping, we use the LDA model generated in the previous step along with the dictionary generated in preprocessing and a loop which loads every single user file are the inputs. The preprocessing of the user file is done first, then according to the words present in every individual users file they are associated with different types of topics in varying proportions. The output is file showing topic distribution of every user. For the movie to topic mapping, the LDA model generated in the previous step along with the dictionary generated in preprocessing and a loop that loads every single movie file are the inputs. The preprocessing of the movie file is done first, then according to the words present in every individual users file they are associated with different types of topics in varying proportions. The output is file showing topic distribution of every movie.

3.5. Ranking

This is the last and most important phase of implementation. When a user enters a query like comedy movieor action movie, the system gives a list of movies which it estimates to be appropriate according to the query. This ranking is done on the basis of a ranking score. In this project to perform this ranking we will use Dual Personalized Ranking (D-PR) Function [1]. D-PR function uses two profiles i.e. personalized document profile, it contains the users perception about each document and a extended user profile which is the sum up of all personalized document profile. Dual personalized ranking function[1] can be defined as follows:

$$\text{Rank}(d, q, u) = \alpha.\text{Sim}(P^1_u , P_{u,d}) + (1 - \alpha).[\beta.\text{Sim}(q, P_d) + (1 - \beta).\text{Score}(q, d)] \quad (1)$$

$$P^1_u = \sum^D (P_{u,d}) \quad (2)$$

Here, u is the user while d is the document,p u,d is the perception of a user u for a document d,p d is the general document profile and p u is the extended user profile, and and are constants. Sim(q, p d) measures the similarity between q and the general documents profile to compute the query-related social matching score Score(q, d) is the textual matching score between query and each document. And lastly Sim(p u , p u,d) is the similarity betwen personalized document profile and extended user profile. So in order to calculate the personalized ranking using D-PR Function in our paper we need three other scores 1. The textual similarity between the query entered by the user and every document(i.e. movie file) 2. The cosine similarity score between the query entered by the user and the social summary of the document(i.e. movie file). 3. The cosine similarity matching score between the personalized document profile and extended user profile. All these scores are computed and summed up for calculation of the Rank(q,d,u) which is the rank of a document d for a user u for a given query q. Step 1: Query q is entered by user u. Step 2: Preprocessing is done on the query and a tf-idf model is generated for the query. Step 3: The textual similarity between query q and every document d is computed, Score(q,d) and the result for all documents are stored in textualsim scores. Step 4: The cosine similarity between the query q and the general document profile Pd , is computed and results for all document are stored in socialmatching scores. Step 5: The cosine similarity between extended user profile p u and the personalized document profile, p u,d is computed Sim(p u , p u,d) and stored in personalization scores. Step 6: Calculate the finalscores by putting everything in a single formula as follows:

Final scores = α.personalizationscores+((1−α).((β.socialmatchingscore)+(1 − β).textualsimscores)) (3)

Step 7: According to the final scores for all movie top n movies will be given to the user.

4. Results

There are different kinds of measures to evaluate the performance of a search engine like the ability of the search engine to deliver complex queries, the speed the search engine to process the query and give results to the user, to display results properly to the user etc. But the ultimate measure to evaluate a search engine is by user happiness. The above factors do contribute to user happiness but, a major factor that drives user happiness is that the results given to the user should be relevant then only a search engine can be deemed to be effective. Measurement of the relevance is with respect to the information need of the user. Here i am taking three different user with different types of interest. First user(ur0555595)'s interest in comedy type of movies. When this user search for different types of keyword like Action, Adventure or Comedy, search results are different. different keyword gives different precision. As shown in table 1. Similarly we compare four different types of user from different interest and if our search query belongs to user's area of interest then precision for that query is high. If our search query is not belong to user's area of interest then precision is low. Search results are shown in table 1.

Table 1: Precision of Different User for Different Type of Keyword

User	Adventure	Action	Comedy	Horror
ur44722272 (Action)	0.8	0.8	0.2	0.4
ur3754831 (Horror)	0.4	0.5	0.6	0.8
ur44722272 (Comedy)	0.8	0.8	0.2	0.4
ur53088790 (Adventure)	0.6	0.9	0.5	0.4

5. Conclusion

In this paper we tried to design a personalized movie search system which gives personalized search result of the user based on the previous reviews of the user. First of all extracted movie features by training the movie reviews to generate topics. Then used the generated topic to perform topic mapping with user and movie. Finally, the final ranking score for query is calculated by using Dual Personalized Ranking Function. Evaluation of the result shows that this approach gives fair results. It also shows that movie reviews are an efficient factor, to be considered to understand a users interests. Reviews also contains substantial information about the movie. The comparison of the personalized search with non-personalized search using precision gives satisfactory results. Lastly, the role of personalization is increasingly growing in all fields as allows to tailor and finetune search results according an individuals preferences and user review are an efficient feature to be used for giving personalized search result to the user. In this project we are using user written reviews for the extraction of features but it can be further combined with other forms movie data like keywords, user rating, genres etc. Further extension can also be done in orderto optimize the query execution time and to improve the system so that it can also give movie recommendation to the user based on his or her likes and dislikes. We are using LDA model for topic modeling but other extensions of it like Correlated topic model (CTM), Dynamic topic model (DTM), Hierarchical LDA can also be used as they still remain unexplored in this field.

References

[1] Xu, Z., Lukasiewicz, T. and Tifrea-Marciuska, O., 2014, September. Improving personalized search on the social web based on similarities between users. In International Conference on Scalable Uncertainty Management (pp. 306-319). Springer, Cham.
[2] Dou, Z., Song, R. and Wen, J.R., 2007, May. A large-scale evaluation and analysis of personalized search strategies. In Proceedings of the 16th international conference on World Wide Web (pp. 581-590). ACM.
[3] Jeh, G. and Widom, J., 2003, May. Scaling personalized web search. In Proceedings of the 12th international conference on World Wide Web (pp. 271-279). ACM.
[4] Feng Qiu and Junghoo Cho. Automatic identification of user interest for personalized search. In Proceedings of the 15th inter- national conference on World Wide Web, pages 727736. ACM, 2006.
[5] Zhongming Ma, Gautam Pant, and Olivia R Liu Sheng. Interest- based personalized search. ACM Transactions on Information Systems (TOIS), 25(1):5, 2007.
[6] Alessandro Micarelli, Fabio Gasparetti, Filippo Sciarrone, and Susan Gauch. Personalized search on the world wide web. In The adaptive web, pages 195230. Springer,2007.
[7] Alexander Pretschner and Susan Gauch. Ontology based per- sonalized search. In Tools with artificial intelligence, 1999. pro- ceedings. 11th ieee international conference on, pages 391398. IEEE, 1999.
[8] Ahu Sieg, Bamshad Mobasher, and Robin Burke. Web search personalization with ontological user profiles. In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pages 525534. ACM, 2007.
[9] Jian-Tao Sun, Hua-Jun Zeng, Huan Liu, Yuchang Lu, and Zheng Chen. Cubesvd: a novel approach to personalized web search. In Proceedings of the 14th international conference on World Wide Web, pages 382390. ACM, 2005.
[10] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Implicit user modeling for personalized search. In Proceedings of the 14th ACM international conference on Information and knowledge management, pages 824831. ACM, 2005.
[11] Mirco Speretta and Susan Gauch. Personalized search based on user search histories. In Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on, pages 622628. IEEE, 2005.
[12] Yannan Chen, Ruifang Liu, and Weiran Xu. Movie recom- mendation in heterogeneous information networks. In Informa- tion Technology, Networking, Electronic and Automation Con- trol Conference, IEEE, pages 637640. IEEE, 2016.
[13] Konstantinos Bougiatiotis and Theodoros Giannakopoulos. Enhanced movie content similarity based on textual, auditory and visual information. Expert Systems with Applications, 96:86102, 2018.
[14] Shinhyun Ahn and Chung-Kon Shi. Exploring movie recom- mendation system using cultural metadata. In Transactions on Edutainment II, pages 119134. Springer, 2009.
[15] Shujuan Zhang, Zhen Jin, and Juan Zhang. The dynamical modeling and simulation analysis of the recommendation on the usermovie network. Physica A: Statistical Mechanics and its Applications, 463:310319, 2016.
[16] Steven Bird, Ewan Klein, and Edward Loper. Natural language processing with Python: analyzing text with the natural language toolkit. OReilly Media, Inc., 2009.
[17] LB Krithika and Kalyana Vasanth Akondi. Survey on various natural language processing toolkits. World Applied Sciences Journal, 32(3):399402, 2014.
[18] David M Blei, Andrew Y Ng, and Michael I Jordan. La- tent dirichlet allocation. Journal of machine Learning research, 3(Jan):9931022, 2003.
[19] Feng Zhao, Yajun Zhu, Hai Jin, and Laurence T Yang. A person- alized hashtag recommendation approach using lda-based topic model in microblog environment. Future Generation Computer Systems, 65:196206, 2016.