LDA based Topic Modeling in Context Aware Movie Recommendations

Mentor: Dr. Pooja Jain

Author:

Piyush Ojha (BT17CSE084)

Himanshu Gupta (BT17CSE093)

Indian Institute of Information Technology, Nagpur

Abstract

Traditional recommender systems, such as those based on content-based and collaborative filtering, tend to use fairly simple user models. They relies on ratings provided by viewers in the movie watching community and the interest of the user to make recommendations to the user. But their performance greatly suffers when little information about the users preferences and few rating are available. In this paper, We attempt to combine Context- awareness approach with probabilistic topic modeling techniques to make intelligent and useful recommendations. We tried to propose an automated movie recommendations based on the similarity of movie and the context. Let given a target movie selected by the user, the goal is to provide a list of those movies that are most similar to the target one, without knowing any user preferences.

Introduction

Recommendation systems are an important technology for TV/movie streaming services like Netflix, audio/music streaming sites like Spotify, Pandora, news article feeds like newshunt, online retailers such as Amazon, Flipkart etc. Indeed, any service provider or content management system that has large quantities of information (or the ability to extract such information) such as usage patterns,browsing and click history, natural text descriptions etc. can and should make use of recommendation methods to help find items of interest. Among various information sources, data in the form of natural text is a particularly rich and expressive source of information, however it is highly unstructured in general. Topic models are used to extract latent structures from large volumes of unlabeled text, that can be used for analysis of content and in turn, aid end goals such as making recommendations. There are dozens of movie recommendation engines on the web. Some require little or no input before they give you movie titles, while others want to find out exactly what your interests are, however all of these systems rely on ratings directly or indirectly expressed by users of the system (some examples are Netflix, Rotten Tomatoes, Movielens, IMDb, Jinni). But these engines fail when the users are new to a system, the first time a system is launched on the market (no previous users have been logged), for new items (where we do not have any history on preferences yet).

The focus of the paper is to provide an automatic movie recommendation system that does not need any a priori information about users. This paper uses Latent Dirichlet Allocation (LDA) with the Context-aware approach. For a give movie, the automatic movie recommendation system supplies to user with a list of those movies that are most similar to the target one. The way the system detects the list of similar movies is based upon an evaluation of similarity among the plot of the target movie and a large amount of plots that is stored in a movie database.

The context where our system works is that of video-on-demand (VOD). Generally speaking, this is the case when a user is looking for an item without being registered on the site in which he is looking for (searching a book on Amazoon, a movie on IMDb etc.). We assumed the only information we have about the user is his first choice, the movie he has selected/ he is watching (we do not have a history about his past selections nor a profile about his general inter-ests). When watching a VOD movie, users explicitly request to buy and to pay for that movie, then what our system attempt to do is proposing a list of similar movies assuming that the chosen film has been appreciated by the user (the system assumes the user liked the movie if his play time is more then 3/4 of the movie play time). Here, we also assume that we have no knowledge about the preferences of the users; namely, about who is watching the film, and also with regard to other users who have previously accessed the system.

Related Works:

Today, recommendation systems are information filtering systems that recommend products available in e-commerce, entertainment items (books music, videos, Video on Demand, books, news, images, events etc.) or people (e.g. on dating sites) that are likely to be of interest to the user. Since almost all these systems use users preferences and atleast 20 rating for an item.

Examples include a recommendation system using the ALS algorithm, a recommendation based on the weighting technique, item similarity-based collaborative filtering, content-based models.

For Example,

Nirav Raval,Vijayshri Khedkar [1], have proposed a Collaborative Filtering, Content-based,and hybrid-based approaches for movie recommadation sysytem. They classified collaborative filtering using various approaches like matrix factorization, user-based recommendation, item-based recommendation. They tried to improve rating based recommadation system.

Sang-Ki Ko,Sang-Min Choi, and Yo-Sub Han [2], proposed a system movie recommendation system based on genre correlations. They modified the previous algorithm; they used a list of movies as input instead of genre combinations and implement a new recommendation algorithm as Android application with additional functions.

Abhishek Bhowmick, Udbhav Prasad and Satwik Kottur [5] proposed a recommendation model based on Collaborative Topic Modeling. They used Probabilistic Matrix Factorization (PMF) for collaborative filtering on movie ratings and Latent Dirichlet Allocation (LDA) for topic modeling of the corpus of movie plot summaries. They combine the latent factor model learned through PMF and the topic model learned through LDA into a single collaborative topic regression model (CTR). They used the observed rating and observed document to make predication.

Also there have done some researches on Topic modeling based on LDA for recommadation system.

Hariri and et al. proposed a combined approach based on contentand collaborative filtering methods from the sequence of songs listened to generatemusic recommendation. They applied a LDA model to reduce the dimensionality ofthe feature and obtain the hidden relationships between songs and tags. They collected218,261 distinct songs from "Art of the Mix" website for evaluation their approach Haririet al [3].

Sonia Bergamaschi, Laura Po and Serena Sorrentino [4] proposed a movie recommadataion system using Topic Models of Latent Semantic Allocation (LSA) and Latent Allocation (LDA). They applied and extensively compared on a movie database of two hundred thousand plots. They examined the topic models behaviour based on standard metrics and on user evaluations, they conducted performance assessments with 30 users to compare their approach with a commercial system.

Parket al. [11] proposed a location-based personalized recommender system, which can reflect users' personal preferences by modeling user contextual information through Bayesian Networks.

Bader et al. [12] have proposed a novel context-aware approach to recommending points-of-interest (POI) for users in an automotive scenario.

4. Our Work

4.1 Motivation:

As we progressed through the process of doing literature survey on the topic- “Topic modeling in web data”, we noticed that Recommender systems are the most interested in and rather exploiting this NLP technique to get the best out of it. Also, the baseline techniques in recommender systems is obviously the Collaborative filtering, Content-based filtering techniques. Netflix is the best fit example today for incorporating all these under the same roof.

So, in this paper, we have embarked on the path of analysing and evaluating the strength that a “Movie Recommendation Engine” can live up to and proposing our methodology using topic Modeling and context-awareness.

4.2 Problem Statement:

The most essential aspects of working of a recommendation system are user rating and item features. Both of them are excapaded to the extreme in traditional RS models. So, basically, this recommendation is the prediction of measure that how highly a user would be intended to rate a particular item(here, movie).

If we have a set of users, U = {u1, u2, u3,..., um} and a set of movies, M = {m1, m2, m3,..., mn}. Here, every user has given some ratings to a list of movies, say Luk. Now, for any random user ui ε U or ui ∉ ui, if we are required to recommend some movies to him/her, we have to tackle through three main problems:

4.2.1 **In-bound prediction problem**: where a preprosessed data allows us to predict rating for a movie which is already rated by a threshold limit number of users.

4.2.2 **Out-bound prediction problem**: also called sparcity problem, where we have to deal with the item which has more or less no ratings at all.

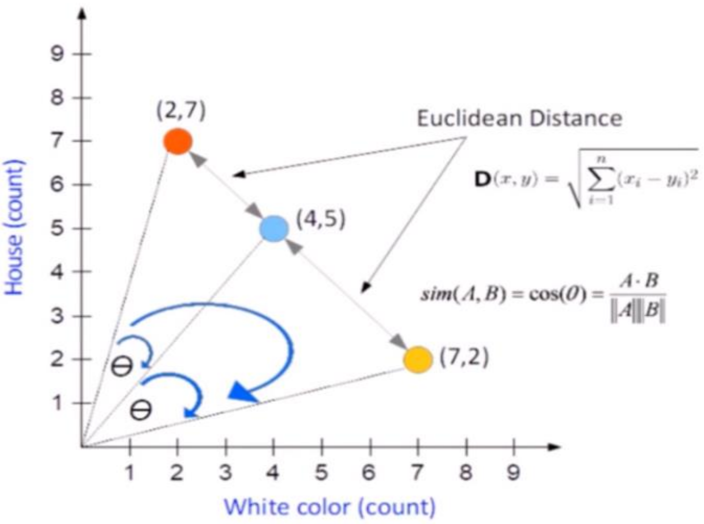
4.2.3 **Closest relevancy problem**: where the sole idea is that the model has to predict the best movies which the user would be interest the most in.

Now, the main idea is to evaluate how well the Topic modeling could affect the performance metrics comaparing to the baseline models.

4.3 Proposed Methodology

4.3.1 Cosine similarity

The model : It’s a representation of the angle between two data points. It results in a value between 0 and 1. The smaller the cosine angle, the bigger the value, indicating higher level of similarity(in this case euclidean distance is less). Comaparing to euclidean distance method, this proves out to be more authentic and efficient, but we will not go into quantitaive analysis of that.


(2,7)
(4,5)
(7,2)
Euclidean Distance
$\mathbf{D}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$
θ
θ
House (count)
White color (count)
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9

$$cosin(vi\ ,\ v\ j) = (\sum k\ (vi\ [k] \cdot vj\ [k])) \div (\surd\ \sum k\ vi\ [k]\ 2 \cdot \surd\ \sum k\ v\ j\ [k]\ 2)$$

In collaborative filtering, the movie similarity score can be calculated using cosine
similarity. So, it solves the [4.2.1] problem by taking a movie as input and throwing a bunch of movies most similar to it.

4.3.2 Context awareness

The model : These systems are known for providing personalised recommendations by making use of user contextual situation. So, unlike the Item-based models which cares about the users ratings and movie features, CA model takes into account the users personal preferences also, like time, location, hobbies and so on.

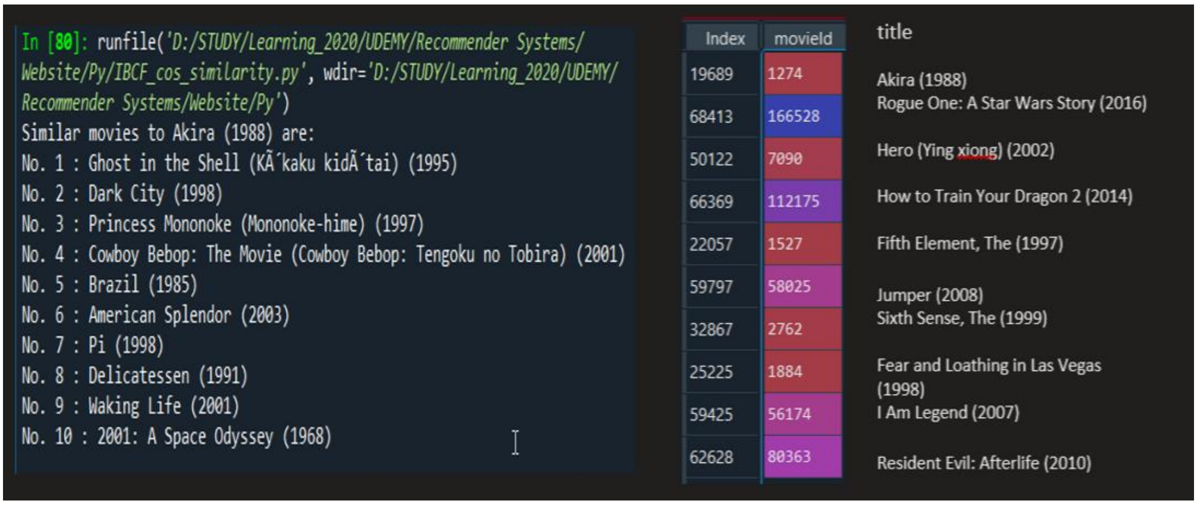
Here, we are considering the time of preference of a user and will evaluate and compare
how likely the Collaborative filtering model (4.3.1) suggests the movies in favour of users' preference.

If we have the dataset such as ‘movies.csv’ containing movie Id, movie title and genres of the movie, and ‘ratings.csv’ containing ‘user Id’ movie Id, movie ratings and movie timestamp.

Then, Merging of these datasets would provide us with the time prefernece of a user corressponding to each movie, which would help the model giving the best possible recommendation at any time of the day.

Evaluation and Comparision:

Cosine similarity in collaborative filtering VS Context aware model



```
In [80]: runfile('D:/STUDY/Learning_2020/UDEMY/Recommender Systems/
Website/Py/IBCF_cos_similarity.py', wdir='D:/STUDY/Learning_2020/UDEMY/
Recommender Systems/Website/Py')
Similar movies to Akira (1988) are:
No. 1 : Ghost in the Shell (KÃ´kaku kidÃ´tai) (1995)
No. 2 : Dark City (1998)
No. 3 : Princess Mononoke (Mononoke-hime) (1997)
No. 4 : Cowboy Bebop: The Movie (Cowboy Bebop: Tengoku no Tobira) (2001)
No. 5 : Brazil (1985)
No. 6 : American Splendor (2003)
No. 7 : Pi (1998)
No. 8 : Delicatessen (1991)
No. 9 : Waking Life (2001)
No. 10 : 2001: A Space Odyssey (1968)
```

Index	movieId	title
19689	1274	Akira (1988)
68413	166528	Rogue One: A Star Wars Story (2016)
50122	7090	Hero (Ying xiong) (2002)
66369	112175	How to Train Your Dragon 2 (2014)
22057	1527	Fifth Element, The (1997)
59797	58025	Jumper (2008)
32867	2762	Sixth Sense, The (1999)
25225	1884	Fear and Loathing in Las Vegas (1998)
59425	56174	I Am Legend (2007)
62628	80363	Resident Evil: Afterlife (2010)

As we can see from the above results produced by both the models that when it comes to

personalised recommendation(recommendation especially for a particular user) how badly

Collaborative model fails.

4.3.3 Topic modeling using LDA

The model: TM algorithm discovers the hidden patterns in a large collection of

documents that correspond to a topic.

LDA is a probabalistic Topic Model, where the main aim is to maximize the separation between means of projected topics and minimize variance within each projected topic.

LDA defines each topic as a bag of words with 3 steps:

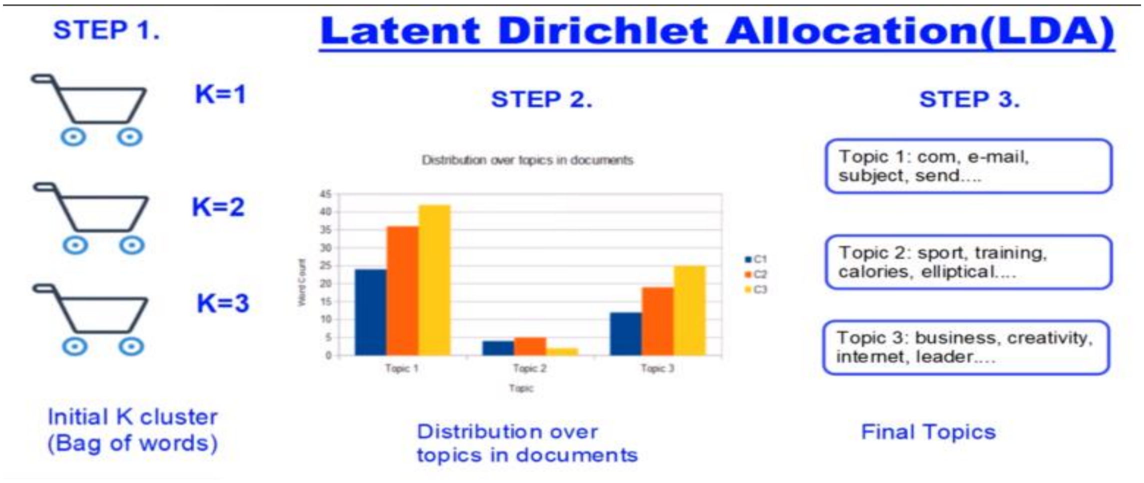
1) Initialize ‘k’ clusters and assign each word in the document to one of the k topics

2) Re-assign word to new topic based on:

   a. how is the proportion of words for a document to a topic, and

   b. how I sthe proportion of a topic widespread across all documents

3) Repeat step 2 until coherent topics result.



Latent Dirichlet Allocation(LDA)

STEP 1.

Initial K cluster (Bag of words)

STEP 2.

Distribution over topics in documents

STEP 3.

Topic 1: com, e-mail, subject, send....

Topic 2: sport, training, calories, elliptical....

Topic 3: business, creativity, internet, leader....

Final Topics

For each topic, we will have different movies’ meta data and movie reviews, and each

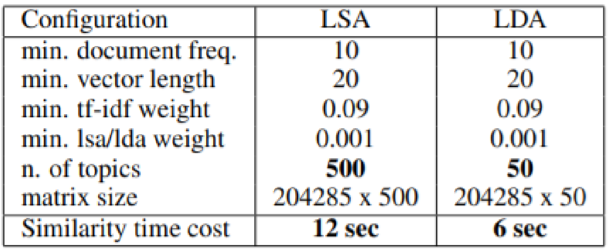
movie detail in a corpus will represent as a document.

LDA can also be considered as a matrix factorization model, since here the approach is to split the document over topic probability distribution.

5. EXPERIMENTS AND RESULTS

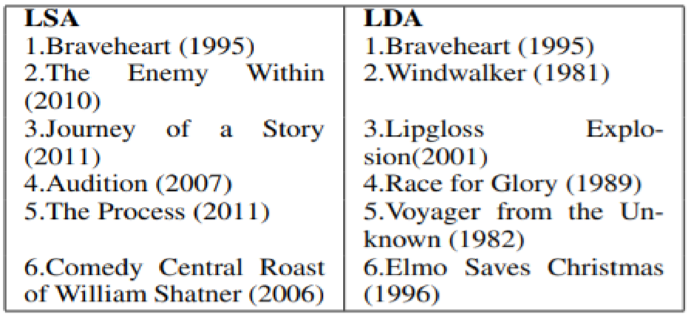
Sonia Bergamaschi et. Al.(2016) presented the comparison for effectiveness of LDA and LSA techniques by doing evaluation of the performance of the system on real users. So, that strengthened our choice of using LDA for topic modeling in our Recommendation system.

Table 1, below shows the time performance of both topic models. Table 2 shows a comparison on the movie “Braveheart”.



Configuration	LSA	LDA
min. document freq.	10	10
min. vector length	20	20
min. tf-idf weight	0.09	0.09
min. lsa/lda weight	0.001	0.001
n. of topics	**500**	**50**
matrix size	204285 x 500	204285 x 50
Similarity time cost	**12 sec**	**6 sec**

Table 1.



LSA	**LDA**
1.Braveheart (1995)	1.Braveheart (1995)
2.The Enemy Within (2010)	2.Windwalker (1981)
3.Journey of a Story (2011)	3.Lipgloss Explosion(2001)
4.Audition (2007)	4.Race for Glory (1989)
5.The Process (2011)	5.Voyager from the Unknown (1982)
6.Comedy Central Roast of William Shatner (2006)	6.Elmo Saves Christmas (1996)

Table 2.

5.1 Context extraction from IMDb

The movies in the IMDb database have a set of hidden features(could be called as topics) that a Topic Modeling technique can be trained on. IMDb provides the features and subsequently, we can extract the contextual information as the subsets of those features. The description of a movie includes the other related entities(here, movies).

We used the plot summary of a movie “Koi Mil Gaya” from the IMDb movie corpus and extracted the movie description. Fig. 3 below, depicts how the movie ‘Koi Mil Gaya’ description looks like after preprocessing the data.

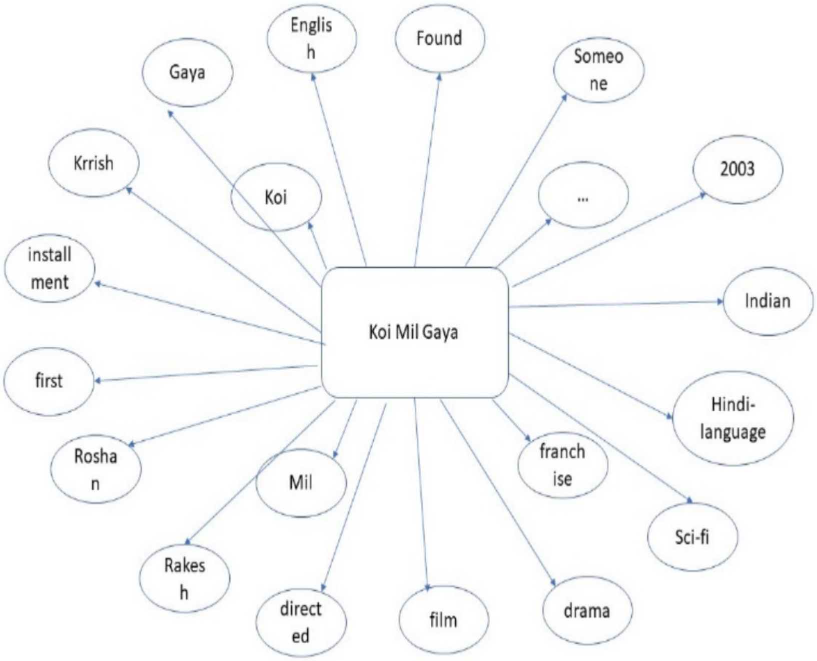
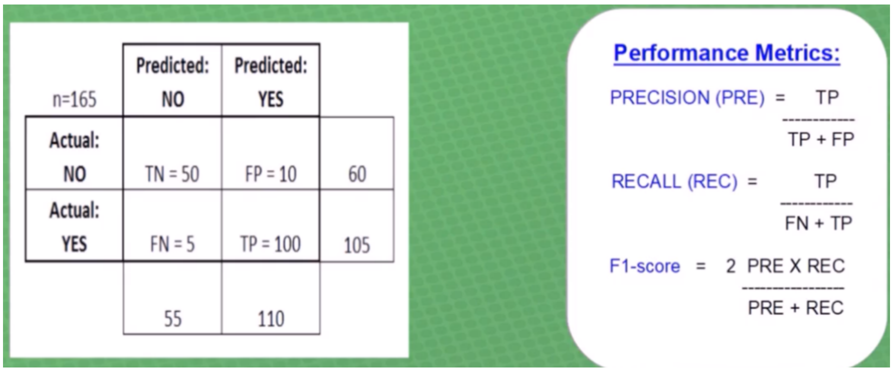


Fig. 3

For each movie, we extract all the related properties viz., movies, genres, directors, actors and so on from the IMDb database and a bag of entities are created. So, each movie is like a distribution over the set of features, and a feature is a distribution over the words in the plot summary. Thus, we perform the extraction operation using LDA model and e set of features, ‘F’ gets generated for a collection of movies. Clearly, the probability of a feature fk under the movie mk would indicate how significant fk is for the movie mk.

5.2 Performance evaluation:

Considering the Table 3 below, we have evaluated the accuracy and precision of the results based on user preference as follows:



n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Performance Metrics:

$$\text{PRECISION (PRE)} = \frac{TP}{TP + FP}$$

$$\text{RECALL (REC)} = \frac{TP}{FN + TP}$$

$$\text{F1-score} = 2 \frac{PRE \text{ X } REC}{PRE + REC}$$

Accuracy = (TP+TN)/(TP+TN+FP+FN)

5.3 Results:

As discussed till now and by performing with few more evaluation metrics, we came to understand that the precision for Recommendation System based on the user preference is most likely to be satisfied strongly by the *Context Topic Modeling using LDA approach*, since it involves a whole bundle of movie corpus available in IMDb database taking care of dimensionality reduction operation which makes the classification process simpler.

6. DATASET

For the whole evaluation process, we incorporated the datasets from mainly two sources: MovieLens1, and IMDb2. The former is the most commonly used dataset for evaluation over the ratings of the movies, while the later provides whole collection of movies' metadata and all movie features along with plot summary.

1 http://grouplens.org/datasets/movielens/1m/

2 http://sisinflab.poliba.it/semanticweb/lod/recsys/datasets/

7. CONCLUSIONS

In this paper, we proposed an LDA based topic modeling method for a Context-aware Recommendation System which extracts the contextual properties of a movie for a user and then the model is trained over the IMDb knowledge base. The 'timestamp' data provides us with the whole user preference hours of the day for each movie and based on that the recommendation is made at any time by the system. Also, the context analysis of the movie corpus via LDA allows us to go even more accurate with the probability of liking from a set of movies that is predicted till now.

8. FUTURE WORKS

For future works, there are still a lot of scopes available already. In context awareness only, we can utilize some dataset containing location preferences of a user and the corressponding likelihood of what movies he/she prefers at any place. Also, till now, we have just used IMDb dataset for context analysis and MovieLens 1M for movie items and ratings information, so in future, we would be interested in utilizing some other huge and widely used dataset viz., Dbpedia, the moviedb.org etc.

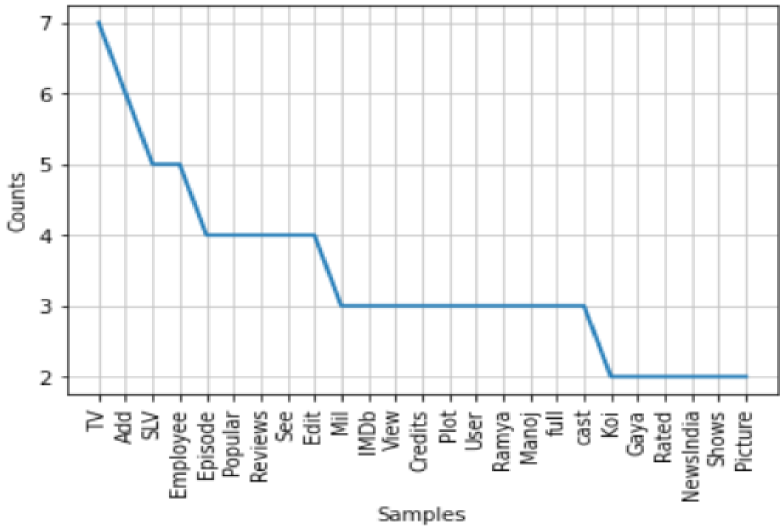


Fig. 4. Context analysis of movie Koi Mil Gaya, depicts the occurrence of keywords in the movie meta data.

Similarly, the whole document over movie and movie over keywords analysis can be maintained and then the Context awareness algorithm can produce a significant predictions based on users' context.

References:

1. http://www.ijstr.org/final-print/dec2019/A-Review-Paper-On-Collaborative-Filtering-Based-Moive-Recommedation-System-.pdf

2. https://link.springer.com/chapter/10.1007/978-3-642-21793-7_63

3. Hariri, B. Mobasher, R. Burke, Context-aware music recommendation based on latenttopicsequential patterns, in: ACM Conference on Recommender Systems, 2012, pp. 131-138

4.https://www.researchgate.net/publication/287741396_Comparing_topic_models_for_a_movie_recommendation_system

5. Movie Recommendation based on CollaborativeTopic Modeling, by Abhishek Bhowmick, Udbhav Prasad and Satwik Kottur, Dept. of Computer ScienceCarnegie Mellon UniversityPittsburgh, PA 15213

6. Adomavicius, G. and Tuzhilin, A. (2005). Toward the nextgeneration of recommender systems: A survey of thestate-of-the-art and possible extensions. IEEE Trans.on Knowl. and Data Eng., 17(6):734–749.

7. Farinella, T., Bergamaschi, S., and Po, L. (2012). A non-intrusive movie recommendation system. In OTMConferences (2), pages 736–751.

8. Gunawardana, A. and Shani, G. (2009). A survey of ac-curacy evaluation metrics of recommendation tasks.The Journal of Machine Learning Research, 10:2935–2962.

9. Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factor-ization techniques for recommender systems. Com-puter, 42(8):30–37.

10. Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. (2011).Collaborative filtering recommender systems. Found.Trends Hum.-Comput. Interact., 4(2):81–173.

11. Park, M.-H., Hong, J.-H., Cho, S.-B.: Location-Based Recommendation SystemUsing Bayesian User's Preference Model in Mobile Devices. In: Indulska, J., Ma,J., Yang, L.T., Ungerer, T., Cao, J. (eds.) UIC 2007. LNCS, vol. 4611, pp. 1130–1139. Springer, Heidelberg (2007)

12. Bader, R., Neufeld, E., Woerndl, W., Prinz, V.: Context-aware poi recommenda-tions in an automotive scenario using multi-criteria decision making methods. In:CaRR 2011, pp. 23–30 (2011)

13. "Python Natural Language Toolkit." http://www.nltk.org/.