

PRIORITIZING BOOKS USING SENTIMENT ANALYSIS OF READER REVIEWS ON VARIOUS ASPECTS OF OTHER LITERATURE

¹R. Latha, ²R.Kamatchi Priya, ³S. Muthuselvan, ⁴T. Vijila

^{1,2,3,4}Department of Computer Science Engineering

Aarupadai Veedu Institute of Technology

Vinayaka Mission's Research Foundation

latha.rbe@gmail.com, kamatchipriyamrajendran@gmail.com, csmuthuselvan@gmail.com,

vijila@avit.ac.in

ABSTRACT

It is a daunting task for readers to manually identify important and interesting criteria from various other reader reviews and their opinions on different aspects of a book. This paper aims at proposing a system that identifies important features of the storyline from reader reviews automatically and determine the orientation of opinions for each feature. Using Sentiment Analysis, a natural language processing task that deals with determining the orientation of opinions in a chunk of text corresponding to an aspect, the system carries out effective clustering and segregation of reviews from readers based on various aspects to provide fine-tuned priorities on literature to readers with similar interests.

Keywords : Prioritizing books, Sentiment Analysis, Review Mining, Support Vector Machine.

I. INTRODUCTION

It is common knowledge that with the evolution and rapid progression of the Internet and its services, many users turn to online recommendation systems for choosing books and other literature. There are dedicated websites with a large collection of literature databases for book and author reviews [1]. Registered users of these websites can review and rate books and authors. These entries are maintained for referential look up when other readers with similar interests browse for books. Most of the recommendations are based on the author of the book, and the genre. Sub genre categorization is also applied in some filtering techniques. Assuming that a person reviews a comical fiction book written by x, and rates it good, a recommendation system that utilizes author-aspect recommendations lists other books written by the same author even if they are of the tragedy or mystery genres. This is not a desirable recommendation.

According to earlier researches [2, 3], the addition of a temporal aspect to the filtering technique enhances the recommendation system to include choices with a wider acceptability. This also may not generate a very specific list of suggestions since the reason that a book has been liked by many over a long period does not guarantee that it matches the liking of the user. Some recommendation systems use Web crawling to gather suggestions after identifying the preference of the user based on the input search string.

II. PROPOSED MODEL

The model proposed conceptualizes a recommendation system that provides a feature-opinion pairing on categories not limited to Author, Title, and Genre, thereby producing a list of recommendations with minimal deviation from the likes and interests of the user. This scheme is carried out in two levels. Firstly, a registered user can review and rate books on the system. Doing so, the user allows the reviews to be subject to Opinion Mining and Analysis, generating reference criteria. Based on the reference criteria, recommendations are listed. The second level of the scheme generates recommendations based on the user's feed on their micro blogging accounts linked to the system.

III. TEXT CLASSIFICATION

Sentiment Analysis is a natural language processing task that enables extraction of required information from source data. Sentiment or Opinion classification is done in three

levels - Document, Sentence, and Aspect. In our system, Sentiment Analysis is an influential factor since recommendations for books are provided based on the user's reviews of other books. There are many techniques in the discipline of Sentiment Analysis that enable us to garner only the vital information from the reviews. This allows us to create a reference pattern for the reader. Based on this reference, new suggestions are made after a pattern match on the books database using Mining techniques.

In a typical input scenario, there is always a chance that review submissions are made in a disorderly fashion. They may contain erroneous spellings, repeated texts, emoticons, and other slang or non-standard words. The primary step in any text mining and classification procedure is addressing this issue and purging the data before proceeding to the next step. This comes as part of data pre-processing attenuating noisy data. Though misspelt and non-standard words need to be removed during tokenization, mapping an emoticon to its sentiment could in fact help in arriving at a stronger polarity to the opinion. A tool called WordNet[4], developed by the Princeton University provides a way to group words with a similar meaning together, thereby making the sentiment analysis more effective. This helps in determining the semantic orientation of a word. If an opinion text uses the words 'good', 'great', 'exciting' 'loved', the inclination is towards positivity. Words such as 'boring', 'dull', 'bad' are deviant from neutrality towards negativity.

Emoticons added in the reviews are converted to Unicode and then looked up a emoticon mapping table to substitute a polarity for the same [5]. That is, if a :) emoticon is available in the raw text, it is looked up the table and the corresponding polarity - in this case, good - is assigned. Any punctuation that does not match the Unicode list in the table is removed. The following table shows a sample set of 3 emoticons and the corresponding polarity. The most positive being +5 and negative being -5.

Smiley	Emotion	Unicode	Polarity
☺	Happy	1F642	+5
:I	Neutral Face	1F610	0
☹	Sad Face	1F641	-5

Table 1.1 Sample set of 3 emoticons and the corresponding polarity

There is also a characteristic with young users to use repeated letters to convey a stronger emotion. For instance, *so* is written as *sooooo*, *great* as *grrreaaattttt*. Such creative spellings can be processed by replacing a continuously repeated pattern with a single letter. Assuming we replace a 3-repetition-pattern, *sooooo* will be converted to *so* and *grrreaaattttt* will be converted to *greattt*. This makes for narrowing erroneous spellings also. Removal of slang and a set of black-listed words from the raw text so as to reduce outliers enhances data pre-processing[6][7].

IV. FEATURE SET SELECTION

One of the most important tasks of building an effective book recommendation system is identifying feature sets. A recommendation engine basically analyzes a user's interests and selects certain features from the emerging pattern. This training set is then compared against acquired datasets containing a vast number of reviews. Selection of such features can be performed using many methods. Majorly used classifiers for this procedure are Naive Bayesian, Maximum Entropy and Support Vector Machines.

Filtering is a mechanism used in identifying interest-oriented reference criteria to resolve feature sets. By general conclusion, Collaborative Filtering method is found to be more successful than Content-Based Filtering methods since content analyses over a large dataset are very difficult. However, with the advent of Hybrid recommendation techniques [8][9], which is effectually a combination of Content Based Filtering and Collaborative filtering, the recommendations made to

users can be more conformed. The problem with using only filtering techniques for recommendation systems is that it leads to the issue of cold start. When a new user enters the system, there is no information a priori. This poses the question of customizing the required preference profile for that user. Unless there is some way of predicting what the user preferences might be, the system will be rendered unsuccessful.

Feature sets are defined as feature-opinion pairs, where major attributes of a book are identified and trained into a set which can be used as a reference for later mapping [10]. In the proposed system, a typical feature set includes sub-genre, storyline, writing style, protagonist, and location.

V. HEURISTIC SVM APPROACH FOR MACHINE LEARNING

Machine learning is an approach used in many recommendation engines to predict values, of which Support Vector Machines [SVM] is a highly effective technique. However, in a dynamic prediction system using the SVM classifier, there a sparsity problem arises. With a large database to verify against, especially in scenarios like recommendation systems, there is an issue of a data lack. Many users provide reviews for only a limited subset of the data due to the enormity of its size. This leads to lacking data. The SVM approach on a sparse scenario such as this produces an ineffective solution.

Manually filling the empty elements with default values may yield deviated results. Therefore this would not be the way to go around for creating an effective recommendation system. According to [11], a heuristic approach is used to resolve these issues and build a better system. This uses the Smooth Support Vector Machine classifier to identify feature sets eliminating empty elements by iteratively estimating missing values. This algorithmic approach is explained by Xia et al [ref]. Applying this heuristic method, a much efficient pattern generation model emerges.

VI. UTILIZATION OF SENTIMENT ANALYSIS

Feature Set classes are first identified and manually trained in the system. For a book, the recommendation features include title, author, genre, sub-genre, lead character, location, plot, and storyline. These features are then paired with the corresponding opinion that have been mined and analysed from the review of users. This mapping is used as a training set to learn and match the user pattern. When a relevant match is obtained, the recommendation engine adds the book to the user profile. The Hybrid filtering method enables the engine to arrive at a learning pattern for a particular user, thereby customizing his recommendations.

Since the system uses a feature – opinion pair to determine the relevancy of preference, the effectiveness of the recommendation system is increased.

VII. CONCLUSION AND FUTURE WORK

Since this system does not use the predominantly used author/genre recommendation method, we can avoid suggestions that are provided solely based on preliminary matches but might have a huge deviation from the actual preference of the user. This paves a way for a very effective recommendation system by means of narrowing user preferences to specifics.

Future work for this recommendation engine focuses on expanding suggestions from books to other forms of literature like poetry, and short prose. Raising the accuracy of machine learning is also an area to address in the future.

REFERENCES

1. <https://www.goodreads.com>
2. Zimdars, Chickering, Meek, Using Temporal Data for Making Recommendations, <http://research.microsoft.com/en-us/um/people/dmax/publications/dataorder-rev.pdf>

3. Chhavi Rana, Sanjay Kumar Jain, Building a Book Recommender system using time based content filtering, <http://www.wseas.org/multimedia/journals/computers/2012/54-571.pdf>
4. <http://wordnetcode.princeton.edu/5papers.pdf>
5. <http://unicode-table.com/en/>
6. <http://www.ijritcc.org/download/1437462752.pdf>
7. <http://www.analyticsvidhya.com/blog/2014/11/text-data-cleaning-steps-python/>
8. <https://www.computer.org/csdl/proceedings/hicss/2005/2268/08/22680217b.pdf>
9. <http://recommender-systems.org/hybrid-recommender-systems/>
10. Zhuang et al, Movie Review Mining and Summarization
11. Support Vector Machines For Collaborative Filtering; Zhonghang Xia et al.

