Ranking System using User Feedback

Kavita Kimmatkar[1], Kaustubh Alandkar[2], Sushant Ghodnadikar[3], Kishor Karbhari[4]
[1, 2, 3, 4] Department of Computer Engineering
[1, 2, 3, 4] ZCOER, Pune - 411041

Abstract- *Digital content is increasing continuously on the internet. From such large amount of information, the user should get required information when demanded with ease. In this paper, we have come up with a system that will rank the content using a user provided feedback and improve its visibility to the user. The user feedbacks are powerful means for improving the quality of content, as they show the interest of the user in a particular content. This paper proposes an approach in which a set of keywords that are essential for providing the usefulness of feedback are considered. By using such sets of keywords a diverse set is created. This approach of using different types of keywords yields better results while performing data analysis. By analyzing such sets the content rank can be evaluated. The proposed system helps in presenting most interesting content to the user with ease using fewer resources.*

Keywords- Content visibility, user feedback, diverse sets, ranking content

I. INTRODUCTION

In this Digital world, there is continuous growth in data volume day-by-day on the Internet. So there is a need for providing quality content when the user demands it. To address this problem visibility of most interesting and useful content should be increased. For increasing the visibility of content this paper proposes a ranking system for ranking content using user feedbacks. For ranking content feedbacks are important for studying the usefulness factor of the content. This usefulness factor can be further used to analyze the content and to evaluate its rank. Our system is efficient because it ranks a particular content automatically when user feedback for that content is given as input to the system. The content's visibility is increased and displayed to the user.

II. LITERATURE REVIEW

Many of the present systems which use user data for data classification satisfy the solution for limited data up to certain extent. Some of the techniques considered for research are discussed below.

P. Melville and R. Mooney [1] used a technique which uses only most useful data elements to reduce the supervision for efficient performance.

The active learning process involves finding relevant data, processing it and using it for a certain application. One common approach in active learning is to select one classifier and choose data points that help the training of this classifier, which normally includes choosing data points according to some confidence measure[2].This approach involves uncertainty sampling[3][4], in which data points that the current classifier is most uncertain about are considered informative[2].

Heterogeneous committee members adapt in different ways and are able to solve different problems. Measuring the efficiency of committee members helps in making satisfactory and accurate decisions [5].

Generally, efficient classifier can be constructed by using domain expert for more accuracy. Considering implementation point of view domain experts proves themselves more useful by answering generalized queries[6]. The significance of generalized query is that they are equivalent to many specific queries. Many times a general query is not sufficient when answers from domain expert are not effective.

A. Blum, T. Mitchell [7] used large unclassified information to increase the efficiency of algorithm when limited information is available. In this method, the description of each portion of data is partitioned into distinct views and learning method is used classify content[7]. The basic consideration of this method is any view can be used for learning if an adequate amount of labeled data is available.
J.Stefanowski, M.Pachocki [8] uses a technique based on constructing active learning systems based on the query by committee. This technique considers the use of levels of disagreement among different classifiers to select the most effective example to query for its label. A technique is introduced based on analyzing the neighborhood of examples, which is applied to create a starting training set for generating the ensemble[8]. Results of this technique confirm that by using limited examples a final classifier can be generated.

III. PROPOSED SYSTEM

By analyzing the data and information acquired by researching the relevant topics, this paper proposes a system which ranks content using a user provided feedback. The block diagram of the system is given below which shows its architecture.

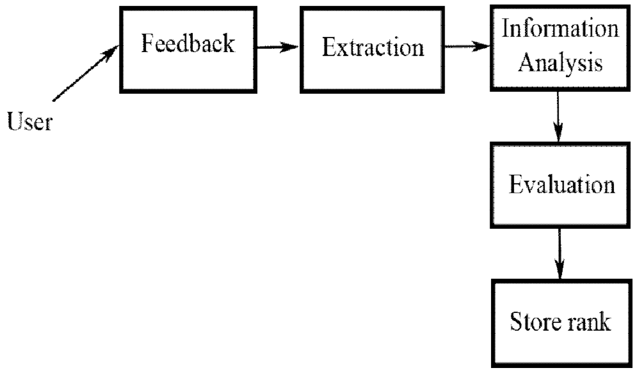

Figure 1. Block Diagram of system

The proposed system consists of three phases which are elaborated below.

3.1 Extraction phase

In this phase, information is collected from the data provided by the user using a frontend. The frontend provides various files, for which user can provide feedback or review. The approach used here considers keywords in reviews or feedback. According to our setting, the ranking score will signify the importance of the review, so useful keywords are extracted using extraction logic. Our interest is in determining the review's impact on content rank by analyzing whether the review is positive or negative. The positive review contains only positive words while the negative review will contain negative words. So this phase will involve identifying such keywords and storing them. These keywords will be used further for determining the context of the review and rank accordingly.

3.2 Information Analysis phase

This phase will process the information provided by extraction phase and use it for further analysis. Each keyword is processed to find its meaning using patterns to determine the context of the review. The keywords that imply positivity are stored in a set containing positive words known as pset. While keywords implying negativity are stored in a set containing negative words known as nset. Using these sets the positive and negative impact of review can be determined. A set containing diverse elements is formed using psets and nsets for a content. Different pset and nset are combined in one diverse set known as dset. The duplicates are removed from dset. The final dset will contain keywords selected randomly that are useful for determining the context of review with no redundant elements. After the formation of dset, we classify each feedback by labeling it as positive, negative or mixed. Using these labels the evaluation of content rank can be done by applying certain formula using pset, nset and dset. The dset formed for a particular content is stored for further processing.

3.3 Evaluation phase

The reviews are analyzed using the labels given to them in Information Analysis phase. Each review is ranked determining the number of elements in pset, nset and dset. The Review Rank can be calculated using below mentioned formula.

$$\text{Review Rank} = \frac{\text{No. of elements in pset} - \text{No. of elements in nset}}{\text{No. of elements in dset}}$$

The above formula considers positive, negative and mixed (containing both positive and negative words) type of reviews for calculating the rank of each review.

Further for the evaluation of content rank we need to find out the total number of reviews for the file under consideration because it will help in signifying the overall number of psets and nsets. Another consideration will be the file's total rating which is a summation of all review ranks for a particular file. And finally, the average number of reviews and average of file ranks of same content type can be considered to evaluate content rank amongst the files of same content type. The above considerations can be formulated in a formula as mentioned below for evaluating the content rank (CR).

$$CR = \frac{(x * y) + (\text{ total no. of reviews} * \text{file's total rating})}{x + \text{total no. of reviews}}$$

where,
x = Average number of reviews across all files of the same content type
y = Average ranking of all files of same content type

The calculated content rank associated with the particular file is stored in the database. The content is displayed to the user by retrieving the data stored in database

IV. ALGORITHM

4.1 Algorithm for extraction

Step 0:	Start
Step 1:	Get text of review
Step 2:	Split into words
Step 3:	Remove special symbols

Step 4: Identify unwanted articles
Step 5: Remove unwanted articles
Step 6: Identify numbers in word
Step 7: Remove words containing numbers
Step 8: Add words in Array called set
Step 9: for i = 0 to Len(where Len is length of set)
Step 10: for i word of Len check for its presence in dictionary
Step 11: if present then add in array called keywords
Step 12: end of for
Step 13: Store keywords in database
Step 14: Stop

4.2 Algorithm for diverse set formation

Step 0: Start
Step 1: Get count of keywords from database in N
Step 2: for i = 0 to N
Step 3: for i word of N check its presence in usefulwords dictionary
Step 4: if present then Add set in Array called dset
Step 5: end of for
Step 6: Identify the duplicate words in the dset and remove them
Step 7: Store dset in database
Step 8: Stop

4.3 Algorithm for ranking review

Step 0: Start
Step 1: Get count of keywords elements from database in N
Step 2: Get count dset from database in dcount
Step 3: Set sum = 0
Step 4: Set rank = 0
Step 5: for i = 0 to N
Step 6: for i keywords of N get count of keyword in ckeys
Step 7: sum = sum + ckeys
Step 8: end of for
Step 9: rank = sum / dcount
Step 10: Store rank in database
Step 11: Stop

4.4 Algorithm for ranking content

Step 0: Start
Step 1: Get count of total no. of reviews from database in TNR
Step 2: Get average ranking of files from database in ARF
Step 3: Get average no. of reviews across all files from database in ANR
Step 4: Set rating = 0, rank = 0
Step 5: for i = 0 to TNR
Step 6: for i keywords of N get rank of review from database in RR
Step 7: rating = rating + RR
Step 8: end of for
Step 9: rank = ((ANR * ARF) + (TNR * rating)) / ANR + TNR
Step 10: Store rank in database
Step 11: Stop

V. ADVANTAGES AND LIMITATIONS

5.1 Advantages of system

1. Only useful information from feedback is considered which increases processing speed.
2. Diverse set produces efficient results.
3. Different kind of reviews are considered for ranking.
4. Automatic ranking of content is done.

5.2 Limitations of system

1. For providing feedback, only English language is considered.

VI. RESULTS

The effectiveness of the proposed system was tested by conducting some experiments using java based windows machine using and Netbeans as IDE. To determine the performance of system we considered a number of reviews(feedbacks) for each content as a benchmark.

To evaluate the system, the system is required to rank different files based on a number of useful reviews. Files are ranked using ranks of each review.

Table 1. Result of the Experiment

File No	No. of useful reviews	Content rank
1	10	102.5905
2	2	40.1762
3	3	43.4892
4	7	60.592
5	6	58.620

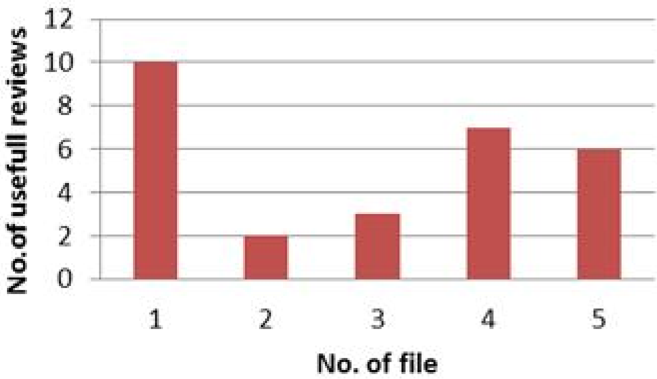

Figure 2. Useful reviews of different files

In Fig. 2, we observe useful reviews for different files are shown. The useful review is a review which consists either positive or negative words. So the positive words in review increase the content rank and negative words decrease the content rank.

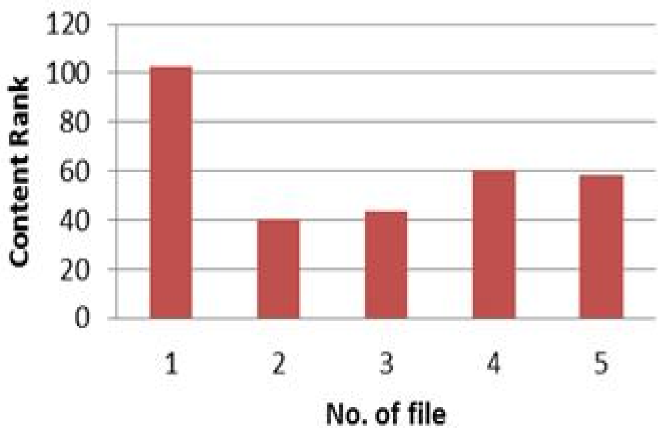

Figure 3. Content rank of different files

In Fig. 3, we observe that number of useful reviews yields ranks for files. The content rank is calculated using the CR formula. Different types of reviews considered while ranking content.

VII. CONCLUSION

It is a tedious task to analyze the text provided by the user as feedback and determine its usability in ranking content. Thus, our system will save time and efforts to perform suck task. Content is automatically ranked in an efficient and quicker way by using above techniques.

VIII. FUTURE SCOPE

The future enhancements to the system would be to add additional languages support in the system.

REFERENCES

[1] P. Melville and R. Mooney, Diverse Ensembles for Active Learning, Proc.Intl Conf. Machine Learning (ICML), pp. 584-591, 2004..

[2] Z. Lu, X. Wu, and J. Bongard, Active Learning with Adaptive Heterogeneous Ensembles, Proc. IEEE Ninth Intl Conf. Data Mining (ICDM), pp.327-336, 2009.

[3] D. D. Lewis and W. A. Gale, “A sequential algorithm for training text classifiers,” in Proceedings of Research and Development in Information Retrieval, 1994, pp. 3-12.

[4] D. D. Lewis and J. Catlett, “Heterogeneous uncertainty sampling for supervised learning,” in Proceedings of the 11th International Conference on Machine Learning, 1994, pp. 148–156.

[5] N. Jankowski and K. Grabczewski. Heterogenous committees with com-petence analysis. In HIS ’05: Proceedings of the Fifth International Conference on Hybrid Intelligent Systems, pages 417–424, Washington, DC, USA, 2005. IEEE Computer Society.

[6] J. Du and C.X. Ling, Asking Generalized Queries to Domain Experts to Improve Learning, IEEE Trans. Knowledge and Data Eng., vol. 22, no. 6, pp. 812-825, June 2010.

[7] A. Blum, T. Mitchell, Combining labeled and unlabeled data with cotrain- ing, in Proceedings of the Workshop on Computational Learning Theory, 1998.

[8] J. Stefanowski, M. Pachocki, Comparing Performance of Committee based Approaches to Active Learning.In: M. Kopotek, A. Przepirkowski, S. Wierzcho, K. Trojanowski (red.) Recent Advances in Intelligent Information Systems, Wydawnictwo EXIT, Warszawa, 2009, 457-470.