International Journal of Research in Engineering, Science and Management
Volume-2, Issue-3, March-2019
www.ijresm.com | ISSN (Online): 2581-5792

Vicinity Modernization Models for Paperback Interpretation

G. Pradeep Kumar[1], N. Vijay[2], R. Kalpana[3], P. Veeralakshmi[4]
[1,2]*Student, Dept. of Information Tech., Prince Shri Venkateshwara Padmavathy Engg. College, Chennai, India*
[3]*Asst. Prof., Dept. of Information Tech., Prince Shri Venkateshwara Padmavathy Engg. College, Chennai, India*
[4]*Assoc. Prof., Dept. of Info. Tech., Prince Shri Venkateshwara Padmavathy Engg. College, Chennai, India*

***Abstract*: Books, as a representative of lengthy documents, convey rich semantics. Traditional document modeling methods, such as bag-of-words models, have difficulty capturing such rich semantics when only considering term-frequency features. In order to explore term spatial distributions over a book, a tree-structured book representation is investigated in this paper. Moreover, an efficient learning framework, Tree2Vector, is introduced for mapping tree-structured book data into vectorial space. In particular, we present two types of locality reconstruction (LR) models: Euclidean-type and cosine-type, during the transformation process of tree structures into vectorial representations. The LR is used for modeling the reconstruction process, in which each parent node in a tree is supposed to be reconstructed by its child nodes. The prominent advantage of this Tree2Vector framework is that it solely utilizes the local information within a single book tree. In addition, extensive experimental results demonstrate that Tree2Vector is able to deliver comparable or better performance in comparison to methods that consider the information of all trees in a database globally. Experimental results also suggest that cosine-type LR consistently performs better than Euclidean-type LR in applications of book and author recommendations.**

***Keywords*: vicinity modernization models, Paperback**

1. Introduction

According to Forrester money spent on e-books will continue to increase at an 18% compound annual growth rate through 2019, as consumers shift their reading and book purchases toward the digital channel. In response, publishers in the digital publishing business have made significant investments in digital content and technology to better serve the needs of their audiences. These efforts seem to correlate with the major societal change in the way that people read books. It is now usual to observe fellow commuters reading ebooks on their portable devices in coffee shops, campus galleries, and on public transportation. However, recommending appropriate books for readers from best sellers written by established authors, to cutting-edge material by innovative new authors, or anything in between, constitutes a challenge for publishers and online stores. In fact, book recommendations are becoming essential to book readers, book sellers, and freelancers. Existing techniques for recommender systems are mainly categorized into collaborative filtering and content based recommendations. Collaborative filtering depends heavily on user activities, e.g., ratings of items according to their preferences. The recommender functions under the assumption of similar preferences among users and a sufficient number of user ratings available in the system. Collaborative filtering, however, has difficulty handling items without sufficient numbers of user ratings and new items that one has purchased or rated, i.e., the so-called cold start problem. In particular, most books are seldom utilized by many patrons according to library use statistics.

Thus, effective content-based recommendations become important when these user activities are sparse. A content based method has been initially developed for book recommendations. Its system, however, depends on a careful feature selection process by labeling every book with values, which is a labor-intensive task. Specific attributes of users must also be provided in advance when evaluating recommended books. Automated text-classification approaches were then employed for exploring content-based book recommender systems. However, the relevance of the recommendations only considered textual metadata, partially extracted from the Internet, rather than actual book text. In industrial applications, e.g., Google Books, full-text indexing has been used commonly for book retrieval via search queries [1]-[7].

2. Related work

We have been motivated through various related works done by various people all over the world. They are as follows:

Now a day's e-books usage was increased. Most of the people moved to online e-books reading. It does not respective to place and time. Many researchers made many research to enhance the e-book usage efficiently. Firstly, A new dual wing harmonium model for document retrieval. vector zed multiple features extracted from different graphs of document representation are able to enhance the retrieval accuracy. Second, Domain Sensitive Recommendation with User Item Subgroup Analysis. Develop a novel Domain-sensitive Recommendation algorithm, which makes rating prediction assisted with the user-item subgroup analysis. Third, develop a novel Domain-sensitive Recommendation algorithm, which makes rating prediction assisted with the user-item subgroup

analysis. Stability and Diversity are important features of recommender systems. It is an important property of recommendation algorithms.

3. System design

In this paper, we introduce a new learning framework, Tree2Vector, for mapping tree-based book structures into vector space. The resulting vectorial representation can improve the querying efficiency in various database applications with no requirement of calculating the tree edit distance. In order to evaluate the importance of children nodes for a parent node, we introduce a locality reconstruction (LR) method to model the reconstruction process, in which each parent node is assumed to be reconstructed by its children. We use a tree structure to represent the feature of each book. By using different book partition strategies, this can be further improved by a finer form of description. Finally, we get a books from library along with its information like page no of search query, books self-number and row number etc. by using the tree2vector. We also use the image comparison technique to identify the book information by comparing its pixel values.

A. Block diagram

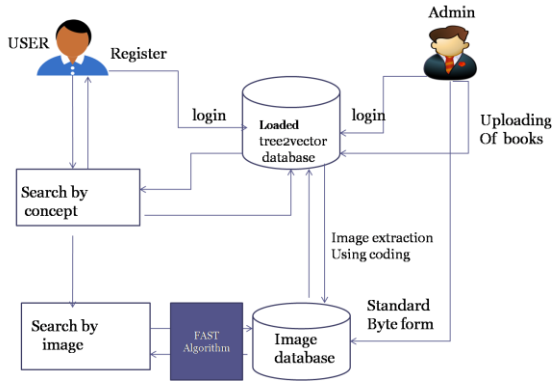

Fig. 1. Block diagram

We have two actors user and admin. The user should register with our system. He can login with their unique id and password. Now he can search the book with his query. User can also search with their images. Admin can register and login with the system. He can upload the books. The book uploaded is converted into tree2vector dataset using tree2vector algorithm to reduce the searching time. The images in the uploaded files are extracted separately using the Bytescount pdf extractor. The image uploaded by user is compared with images in the database using fast algorithm.

B. Algorithm

Tree2vector:

Tree2Vector delivers two desirable features: 1) generality, as a vectorial representation accomplished by Tree2Vector can be used independently in real-world applications once it is formulated by the Tree2Vector framework, while different tasks based on book contents are entangled in the MLSOM testing process; and 2) efficiency, as there is no requirement of utilizing global information by simultaneously comparing nodes of all of the trees at the same level, as with MLSOM. In other words, Tree2Vector does not require a pre-training procedure; instead, each book tree can be processed separately in a once for-all manner. StepsTree2vector.

Algorithm:

1) The root node in the tree represents the entire book.
2) The nodes at the second level indicate pages that are segmented from the book.
3) he nodes at the third level represent paragraphs of the pages. Thus, a “book-->pages-->paragraphs” tree is constructed for the book description.

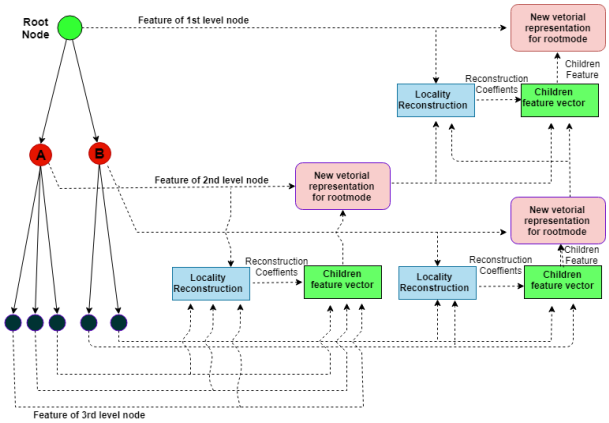

Fig. 2. Tree2vector algorithm

C. FAST algorithm

1. Select a pixel p in the image which is to be identified as an interest point or not. Let its intensity be I_p.
2. Select appropriate threshold value t.
3. Consider a circle of 16 pixels around the pixel under test. (See the image below)

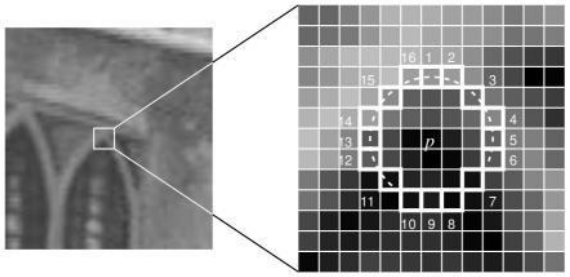

Fig. 3. FAST algorithm

4. Now the pixel p is a corner if there exists a set of n contiguous pixels in the circle (of 16 pixels) which are all brighter than $I_p + t$, or all darker than $I_p - t$. (Shown as white dash lines in the above image). n was chosen to be 12.
5. A high-speed test was proposed to exclude a large number of non-corners. This test examines only the

four pixels at 1, 9, 5 and 13 (First 1 and 9 are tested if they are too brighter or darker. If so, then checks 5 and 13). If p is a corner, then at least three of these must all be brighter than $I_p + t$ or darker than $I_p - t$. If neither of these is the case, then p cannot be a corner. The full segment test criterion can then be applied to the passed candidates by examining all pixels in the circle. This detector in itself exhibits high performance, but there are several weaknesses:

- It does not reject as many candidates for n < 12.
- The choice of pixels is not optimal because its efficiency depends on ordering of the questions and distribution of corner appearances.
- Results of high-speed tests are thrown away.
- Multiple features are detected adjacent to one another.

First 3 points are addressed with a machine learning approach. Last one is addressed using non-maximal suppression.

D. Machine Learning a Corner Detector

1. Select a set of images for training (preferably from the target application domain)
2. Run FAST algorithm in every images to find feature points.
3. For every feature point, store the 16 pixels around it as a vector. Do it for all the images to get feature vector P
4. Each pixel (say x) in these 16 pixels can have one of the following three states:

$$S_{p\to x} = \begin{cases} d, & I_{p\to x} \le I_p - t & \text{(darker)} \\ s, & I_p - t < I_{p\to x} < I_p + t & \text{(similar)} \\ b, & I_p + t \le I_{p\to x} & \text{(brighter)} \end{cases}$$

5. Depending on these states, the feature vector P is subdivided into 3 subsets, P_d, P_s, P_b.
6. Define a new boolean variable, K_p, which is true if p is a corner and false otherwise.
7. Use the ID3 algorithm (decision tree classifier) to query each subset using the variable K_p for the knowledge about the true class. It selects the x which yields the most information about whether the candidate pixel is a corner, measured by the entropy of K_p.
8. This is recursively applied to all the subsets until its entropy is zero.
9. The decision tree so created is used for fast detection in other images.

E. Non-maximal Suppression

Detecting multiple interest points in adjacent locations is another problem. It is solved by using Non-Maximum Suppression.

1. Compute a score function, V for all the detected feature points. V is the sum of absolute difference between p and 16 surrounding pixels values.
2. Consider two adjacent key points and compute their V values.
3. Discard the one with lower V value.

4. Practicality of tree2vector

The Tree2vector is used in the library searching system so that searching can be enhanced to efficient level. Students can avail the library service at anytime from anywhere. Researchers manual effort can be greatly reduced.

5. Result

The tree2vector is implemented in library system. The images in the PDF are extracted by using the bytescout pdf extractor. By using different book partition strategies, this can be further improved by a finer form of description. Finally, we get a books from library along with its information like page no of search query, books self-number and row number etc. by using the tree2vector. We also use the image comparison technique to identify the book information by comparing its pixel values

6. Conclusion

It is concluded that this locality reconstruction model using tree2vector algorithm enhance the library system to the modern world. User's manual effort for searching of books and topics can be greatly reduced. The whole system can be hosted up with a server link and it can be distributed to students for non-restriction access of library can be developed in future process.

References

[1] H. Yang, G. Ling, Y. Su, *et al.*, "Boosting response aware modelbased collaborative filtering," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 8, pp. 2064-2077, Jun. 2015.
[2] Y. Cai, H.-F. Leung, Q. Li, H. Min, J. Tang, and J. Li, "Typicalitybased collaborative filtering recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 766-779, Mar. 2014.
[3] J. Bu, X. Shen, B. Xu, *et al.*, "Improving collaborative recommendation via user-item subgroups," *IEEE Trans. Knowl. Data Eng.*, vol.28, no. 9, pp. 2363-2375, Sept. 2016.
[4] G. Guo, J. Zhang, N. Yorke-Smith, "A novel recommendation model regularized with user trust and item ratings. *IEEE Trans.Knowl. Data Eng.*, vol. 28, no. 7, pp. 1607-1620, Jul. 2016.
[5] J. Liu, Y. Jiang, Z. Li, *et al.*, "Domain-sensitive recommendation with user-item subgroup analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 939-950, Apr. 2016.
[6] G. Adomavicius, J. Zhang, "Improving stability of recommender systems: A meta-algorithmic approach," *IEEE Trans. Knowl. Data Eng.*, vol 27, no. 6, pp. 1573-1587, Jun. 2015.
[7] A. Kent *et al.*, Use of library materials: The University of Pittsburgh study. New York, NY, USA: Marcel Dekker, 1979.
[8] E. Rich, "User modeling via stereotypes," *Cognit. Sci.*, vol. 3, no. 4, pp. 329-354, Oct. 1979.
[9] E. Rich, "Users are individuals: Individualizing user models," *Int.J. Man-Mach. Stud.*, vol. 18, no. 3, pp. 199-214, 1983.
[10] R. J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *Proc. 5th ACM Conf. Digit. Libraries*, San Antonio, TX, USA, Jun. 2000, pp. 195-204.

[11] G. Salton and M. J. McGill, Introduction to Modern Information Retrieval. New York, NY, USA: McGraw-Hill, 1983.
[12] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391-407, 1990.
[13] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Berkeley, CA, USA, Aug. 1999, pp. 50-57.
[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993-1022, Mar. 2003.