DISTINCTIVE SMART HOTEL GUIDER FOR BIG DATA APPLICATIONS

[1]Mrs.K.Krishnakumari, [2]Ms.S.Aarthi

[1]Associate professor, Department of Computer Science and Engineering, A.V.C. College of Engineering, Mannapandal-609 305, Tamil Nadu, India
krishna.41999@gmail.com

[2] PG Student, Department of Computer Science and Engineering, A.V.C. College of Engineering, Mannapandal-609 305, Tamil Nadu, India
aarthisadagopanvb@gmail.com

Abstract:

The main aim of this project is to develop a smart hotel recommendation system to address scalability and inefficiency problem in Big Data with traditional hotel recommender systems, which fails to meet users' personalized requirements and diverse Preferences. This project uses user collaborative search to meet the users' personalized recommendations. The users' are asked to give their amenity preferences in the beginning, which are then used for populating the most appropriate hotel with the services that the current user prefers. The Top k algorithm is used to list out the hotels meeting the users' preferences. The more the preferences of the user, the search gets narrower and most appropriate hotels are listed out. The review of current user also gets added for future users; so that they can use it as review of previous user.

I. Introduction:

Big data is a broad term for data sets so large or complex that traditional data processing applications are inadequate. Challenges include analysis, capture, data curation, search, sharing, storage, transfer, visualization, and information privacy. The term often refers simply to the use of predictive analytics or other certain advanced methods to extract value from data, and seldom to a particular size of data set. Accuracy in big data may lead to more confident decision making. And better decisions can mean greater operational efficiency, cost reduction and reduced risk.

Analysis of data sets can find new correlations, to "spot business trends, prevent diseases, combat crime and so on." Scientists, business executives, practitioners of media and advertising and governments alike regularly meet difficulties with large data sets in areas including Internet search, finance and business informatics. Scientists encounter limitations in e-Science work, including meteorology, genomics, connectomics, complex physics simulations, and biological and environmental research.

II. Motivation:

Data sets grow in size in part because they are increasingly being gathered by cheap and numerous information-sensing mobile devices, aerial (remote sensing), software logs, cameras, microphones, radio-frequency identification (RFID) readers, and wireless sensor networks. The world's technological per-capita capacity to store information has roughly doubled every 40 months since the 1980s; as of 2012, every day 2.5 exabytes (2.5×1018) of data were created; The challenge for large enterprises is determining who should own big data initiatives that straddle the entire organization.

Work with big data is necessarily uncommon; most analysis is of "PC size" data, on a desktop PC or notebook that can handle the available data set.

Relational database management systems and desktop statistics and visualization packages often have difficulty handling big data. The work instead requires "massively parallel software running on tens, hundreds, or even thousands of servers". What is considered "big data" varies depending on the capabilities of the users and their tools, and expanding capabilities make Big Data a moving target. Thus, what is considered "big" one year becomes ordinary later. For some organizations, facing hundreds of gigabytes of data for the first time may trigger a need to reconsider data management options. For others, it may take tens or hundreds of terabytes before data size becomes a significant consideration.

III. Existing System

In most existing service recommender systems, such as hotel reservation systems and restaurant guides, the ratings of services and the service recommendation lists presented to users are the same. They have not considered users' different preferences, without meeting users' personalized requirements.

Most existing service recommender systems are only based on a single numerical rating to represent a service's utility as a whole. In fact, evaluating a service through multiple criteria and taking into account of user feedback can help to make more effective recommendations for the users.

Existing Approaches solve the scalability problem by dividing dataset. But their method doesn't have favorable scalability and efficiency if the amount of data grows.

IV. Proposed System

A keyword-aware service recommendation method, named KASR, is proposed in this paper, which is based on a user-based Collaborative Filtering algorithm.

In KASR, keywords extracted from reviews of previous users are used to indicate their preferences. Moreover, we implement it on a distributed computing platform, Hadoop, which uses MapReduce as its computing framework.

In Kasr, keywords are used to indicate both of users' preferences and the quality of candidate services. A user-based CF algorithm is adopted to generate appropriate recommendations. KASR aims at calculating a personalized rating of each candidate service for a user, and then presenting a personalized service recommendation list and recommending the most appropriate services to him/her. Moreover, to improve the scalability and efficiency of our recommendation method in “Big Data” environment, we implement it in a MapReduce framework on Hadoop by splitting the proposed algorithm into multiple MapReduce phases.

V. Architecture Diagram:

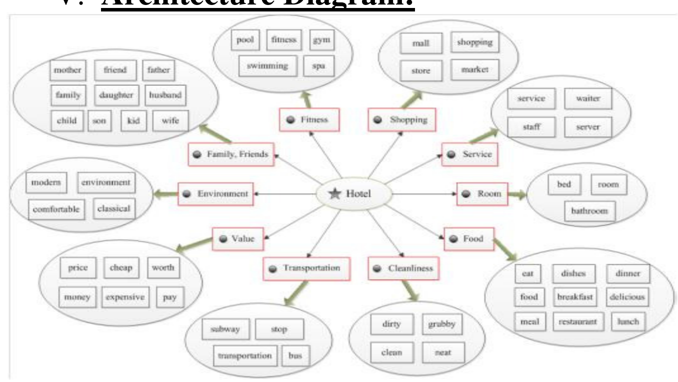

Fig: Domain thesaurus of hotel reservation system

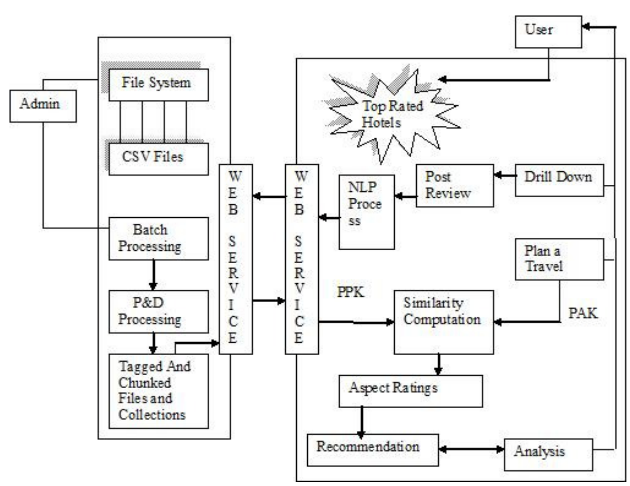

Fig: Detailed architecture

VI. Modules:

1. Big Data and Environment
2. Batching and Preprocess
3. Digging in Big Data & Service Recommender Application
4. MapReduce and Hadoop
5. KASR and Analysis

VII. Module Description:

1. Big Data and Environment

Huge Collection of data is retrieved from open source datasets that are publicly available from major Travel Recommendation Applications. Big Data Schemas were analyzed and a Working Rule of the Schema is determined. The CSV(Comma separated values) files were read and manipulated using Java API that itself developed by us which is developer friendly ,light weighted and easily modifiable.

2.Batching and Preprocess

The Traditional View of Service Recommender Systems that shows Top-K Results are displayed with Paginations with which a user can navigate Back and Forth of the Result sets. All Services Ratings and Reviews of Each Hotels are listed. POS(Parts of Speech) Tagger and Chucker Process are done on each and every review of all hotels for all countries in a Parallel and Distributed Manner as Batch jobs. The Master Job is Split up into 'n' no of small Batch jobs based on the slave machines Connected with the Master. POS Tagger tags each words of a review with its tags and the Clunker Process will take POS tagged output as input for Groping the Words based on meaning of the Review.

3.Digging in Big Data & Service Recommender Application

The CSV Files in distributed Systems are invoked through Web Service Running in the Server Machine of the Host Process through a Web Service Client Process in the Recommendation System. The data that Retrieved to the Recommendation Systems are provided with a clean GUI and can be queried on Demand. Each and Every process on the Recommendation Application invokes Web Service which uses light weighted traversal of data using XML. The Users can Review each hotel and can post comments also. The Reviews gets updated to the CSV Files as it get retrieved.

A User can Plan or Schedule a Travel highlighting his requirements in a detailed way that shows the Preference Keywords Set of the Active User. A Domain Thesaurus is built depending on the Keyword Candidate List and Candidate Services List. The Domain Thesaurus can be Updated Regularly to get accurate Results of the Recommendation System.

4.MapReduce and Hadoop

(1) Capture user preferences by a keyword-aware approach:

In this step, the preferences of active users and previous users are formalized into their corresponding preference keyword sets respectively. In this paper, an active user refers to a current user needs recommendation.

a) Preferences of an active user.

An active user can give his/her preferences about candidate services by selecting keywords from a keyword-candidate list, which reflect the quality criteria of the services he/she is concerned about. Besides, the active user should also select the importance degree of the keywords. The importance degree of the keywords as "1" represents the general, "3" represents important and "5" represents very important.

b) Preferences of previous users.

The preferences of a previous user for a candidate service are extracted from his/her reviews for the service according to the keyword-candidate list and domain thesaurus. And a review of the previous user will be formalized into the preference key-word set of User.

(2)The keyword extraction process is described as follows:

a) Preprocess:

First, tags and stop words in the reviews snippet collection should be removed to avoid affecting the quality of the keyword extraction in the next stage. And the Porter Stemmer algorithm is used to remove the commoner morphological and in flexional endings from words in English.

b) Keyword extraction:

In this phase, each review will be transformed into a corresponding keyword set according to the keyword-candidate list and domain thesaurus. If the review contains a word in the domain thesaurus, then the corresponding keyword should be extracted into the preference keyword set of the user Known as (PPK).

(3) Similarity computation:

The Third step is to identify the reviews of previous users(PPK) who have similar tastes to an active user by finding neighborhoods of the active user(PAK) based on the similarity of their preferences. Before similarity computation, the reviews unrelated to the active user's preferences will be filtered out by the intersection concept in set theory. If the intersection of the preference keyword sets of the active user and a previous user is an empty set, then the preference keyword set of the previous user will be filtered out.

5. KASR and Analysis:

The Chunked Reviews of the Similar User List is retrieved and the Keywords corresponding to the User is analyzed for its Valence and Arousal. Valence Means Weather the Keywords Means a positive or Negative thing and Arousal answers, how much it is? Ratings are given for each Domain based on the Valence and Arousal for each User of each hotel. The Overall Hotel Rating is now manipulated by taking average values of each rating of several users of a particular hotel. Now ranking is done for all hotels based on Ratings and will be sorted based on Bubble Sort Algorithm to have the Most appropriate personalized Recommendation for the User. The Results will be analyzed with Graphical Views so as to understand easier.

VIII. Enhancement:

The Natural Language Processing is implemented to analyze the reviews of the previous user. The NLP Process Comprises Tokenizing a Sentence or a word, POS (Parts of Speech) Tagging, Extraction of Nouns and Verbs, Synonym Retrieval and Spell Check of Extracted Keywords using WordNet Dictionary .Valence and Arousal will be implemented for calculating Ratings of Aspects of a Hotel. The BigData manipulation from CSV through Our Own JAVA API enforces developer friendly access.

IX. Conclusion:

In this project, we have proposed a personalized hotel recommender for users with diverse recommendations. Active user gives his/her requirements by selecting keywords from keyword candidate list, along with the importance degree of the keywords. User-based collaborative filtering algorithm is used to generate appropriate recommendations. POS tagging and chunking process are done to extract a meaningful keyword. The positive and negative preferences can be distinguished from the previous user reviews via NLP, valence and arousal. KASR is implemented in Hadoop platform to address the big data problems so that it improves the scalability and efficiency over existing approaches

X. References:

[1] J. Manyika et al., "Big Data: The Next Frontier for Innovation, Competition, and productivity," 2011.

[2] C. Lynch, "Big Data: How Do Your Data Grow?" Nature, vol. 455, no. 7209, pp. 28-29, 2008.

[3] F. Chang, J. Dean, S. Ghemawat, and W.C. Hsieh, "Bigtable: A Distributed Storage System for Structured Data," ACM Trans. Computer Systems, vol. 26, no. 2, article 4, 2008.

[4] W. Dou, X. Zhang, J. Liu, and J. Chen, "HireSome-II: Towards Privacy-Aware Cross-Cloud Service Composition for Big Data Applications," IEEE Trans. Parallel and Distributed Systems, 2013.

[5] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, Jan. 2003.

[6] M. Bjelica, "Towards TV Recommender System Experiments with User Modeling," IEEE Trans. Consumer Electronics, vol. 56, no. 3, pp. 1763-1769, Aug. 2010.

[7] M. Alduan, F. Alvarez, J. Menendez, and O. Baez, "Recommender System for Sport

Videos Based on User Audiovisual Consumption," IEEE Trans. Multimedia, vol. 14, no. 6, pp. 1546-1557, Dec. 2012.

[8] Y. Chen, A. Cheng, and W. Hsu, "Travel Recommendation by Mining People Attributes and Travel Group Types from Community-Contributed Photos," IEEE Trans. Multimedia, vol. 25, no. 6, pp. 1283-1295, Oct. 2013.

[9] Z. Zheng, X. Wu, Y. Zhang, M. Lyu, and J. Wang, "QoS Ranking Prediction for Cloud Services," IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 6, pp. 1213-1222, June 2013.

[10] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and Evaluating Choices in a Virtual Community of Use," Proc. SIGCHI Conf. Human Factors in Computing System (CHI '95), pp. 194-201, 1995.

[11] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," Proc. ACM Conf. Computer Supported Cooperative Work (CSCW '94), pp. 175-186, 1994.

[12] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331-370, 2002.

[13] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the Stateof- the-Art and Possible Extensions," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 6, pp. 734-749, June 2005.

[14] D. Agrawal, S. Das, and A. El Abbadi, "Big Data and Cloud Computing: New Wine or Just New Bottles?" Proc. VLDB Endowment, vol. 3, no. 1, pp. 1647-1648, 2010.

[15] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Comm. ACM, vol. 51, no. 1, pp. 107-113, 2005.

[16] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazons Highly Available Key-Value Store," In Proc. 21st ACM Symp. Operating Systems Principles, pp. 205-220, 2007.

[17] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, "Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks," Proc. European Conf. Computer Systems, pp. 59-72, 2007.

[18] S. Ghemawat, H. Gobioff, and S. T. Leung, "The Google File System," Proc. 19th ACM Symp. Operating Systems Principles, pp. 29- 43, 2003. [19] L. Zhang, "Editorial: Big Services Era: Global Trends of Cloud Computing and Big Data," IEEE Trans. Services Computing, vol. 5, no. 4, pp. 467-468, Fourth Quarter, 2012.

[20] Z. Luo, Y. Li, and J. Yin, "Location: A Feature for Service Selection in the Era of Big Data," Proc. IEEE 20th Int'l Conf. Web Service, pp. 515-522, 2013