World Wide Web Information Retrieval using Clustering

Manjunath Naik[1] Sudha V[2]
[2]Assistant Professor
[1, 2]Information Science Department
[1, 2]RNS Institute of Technology, Bangalore VTU, Belgaum, Karnataka, India.

Abstract— Crowdsourcing, a distributed process that involves outsourcing tasks to a network of people, is increasingly used by companies for generating solutions to problems of various kinds. In this way, thousands of people contribute a large amount of text data that needs to already be structured during the process of idea generation in order to avoid repetitions and to maximize the solution space. This is a hard information retrieval problem as the texts are very short and have little predefined structure. We present a solution that involves three steps: text data preprocessing, clustering, and visualization. In this contribution, we focus on clustering and visualization by presenting a Support Vector machine approach that is able to learn the principal components of the data while the data set is continuously growing in size. We compare our approach to standard clustering applications and demonstrate its superiority with respect to classification reliability on a real-world example.

Keywords: Crowdsourcing, Clustering, Preprocessing, SVM

I. INTRODUCTION

Blogs, message boards, crowdsourcing forums, chatrooms,... – the Web is full of platforms, where people share information. The content on these platforms is typically structured according to the temporal order of the entries, leading to topical threads. With a growing number of entries, users lose track of the information and repetitions with regard to content are likely to occur. In the case of crowdsourcing platforms, where contributors submit ideas or solutions to a posted problem and the initiator of a crowdsourcing process (crowdsourcer) awards the best contributions, repetitions are undesirable and the lack of a general overview hinders the exploration of the solution space. The crowdsourcer aims to gather a huge variety of ideas, spanning a wide solution space that maximizes the potential to find the best solution for the problem posed. As a crowdsourcing process may yield thousands of contributions, both the crowdsourcer and the contributor face the tasks of structuring, classifying and evaluating the contributions. These tasks should be performed in real-time, i.e. during the submission.

This information retrieval problem involves the steps preprocessing, clustering and visualization. The basic methodology, as used for instance in internet search engines, starts with a vectorization of the texts and constructs a 'term by document' or 'term frequency inverse document frequency' (TF-IDF) matrix1[1], [2]. The matrix contains information about the occurrence of each (semantically relevant) term in a document, based on which, a proximity matrix for document clustering can be constructed.

For the visualization of data from a document corpus,a variety of methods have been suggested whose commonality is the idea of dimensionality reduction. Among the most popular methods are multidimensional scaling (MDS), self-organizing maps (SOM), Principal component analysis (PCA) or Linear Discriminant Analysis (LDA). Depending on the characteristics of the data set, certain approaches are favored over others [4].

Applying these standard methods to our problem is faced with various challenges: First, the texts are usually very short (ideas are described mostly by one to three sentences), lack formal structure, and are prone to errors, sloppy writing, and multilingualism (ideas may contain words of different languages). This means that the information content per text is low and unstable (i.e. may change dramatically if a misspelled word is not recognized), which makes it difficult to compare and group the texts such that the latent topics of each group can be identified. Second, the number of topics is high (as maximizing the number of topics is one of the goals of the crowdsourcing process), which leads to a high dimensional feature space. This complicates the visualization task and consequently the identification of yet unexplored topics, as the results have to be presented in an easy-to-grasp visual form representing the basic structure of the underlying textual space.

Traditional visualization techniques such as MDS or PCA are inadequate for this task, because they employ a 2-dimensional projection with a high information loss. Third, the process is dynamic, i.e. the number of texts and thus the vocabulary and the number of topics increases with time. The users need immediate, i.e. real-time, feedback requiring efficient algorithms. Nonetheless, the visualization should not display abrupt changes as they can be confusing to users and crowdsourcers. Rather, changes should happen continuously and almost imperceptibly. In this contribution, we introduce a novel information retrieval method that masters these challenges and gives appropriate results based on the technique. This shows the Visualization process of the documents.

II. RELATED WORK

The procedure stated by the author discusses three major steps namely: text data pre-processing, clustering, and visualization.

A. Text Data Pre-processing

The enormous amount of information stored in unstructured texts cannot simply be used for further processing by computers, which typically handle text as simple sequences of character strings. Therefore, specific pre-processing methods and algorithms are required in order to extract useful patterns. Text mining refers generally to the process of extracting interesting information and knowledge from unstructured text. Text mining can be also defined similar to data mining as the application of algorithms and methods from the field's machine learning and statistics to texts with the goal of finding useful patterns. For this purpose it is necessary to pre-process the texts accordingly. Many authors use information extraction methods, natural

language processing or some simple pre-processing steps in order to extract data from texts.

Today's real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size (often several gigabytes or more) and their likely origin from multiple, heterogeneous sources. Low-quality data will lead to low-quality mining results. There are a number of data pre-processing techniques. Data Cleaning can be applied to remove noise and correct inconsistencies in the data. Data Integration merges data from multiple sources into a coherent data store, such as a data warehouse, Data Transformation such as normalization, may be applied. For example, normalization may improve the accuracy and efficiency of mining algorithms involving distance measurements, Data reduction can be used for reducing the size by aggregating, eliminating redundant features or clustering for instance. These techniques are not mutually exclusive; they may work together.

For optimization in Preprocessing it will use the Probabilistic topic models which takes frequency of each word in corpus and forms a weight for the word in a document, As the collective knowledge continues to be digitized and stored in the form of news, blogs, web pages, scientific articles, books, images, sound, video, and social networks it becomes more difficult to discover what it is looking for. It needs new computational tool to help organize, search and understand these vast amounts of information Right now, it works on online information using two main tools search and links.

B. Clustering

Clustering is a data mining (machine learning) technique used to place data elements into related groups without advance knowledge of the group definitions. Popular clustering techniques include k-means clustering and expectation maximization (EM) clustering. Data Clustering is a technique in which, the information that is logically similar is physically stored together. In order to increase the efficiency in the database systems the number of disk access are to be minimized. In clustering the objects of similar properties are placed in one class of objects and a single access to the disk makes the entire class available.

A combined clustering method was implemented based on the group-average clustering algorithm (Yang *et al*., 2002) by considering the Euclidean distance between records. The clustering algorithm was run several times adjusting the maximum size of the clusters. Ultimately, the goal is to identify as outliers those records previously containing outlier values. However, computational time prohibits multiple runs in an every-day business application on larger data sets. After several executions on the same data set, it turned out that the larger the threshold value for the maximum distance allowed between clusters to be merged, the better the outlier detection.

Clustering objects into groups is a common task that arises in many applications such as data mining, web analysis, computational biology, facility location, data compression, marketing, machine learning, pattern recognition, and computer vision. Clustering algorithms for these and other objectives have been heavily investigated in the literature. Clustering is a very popular descriptive data mining technique that aids describing characteristics of data sets. The goal of clustering is to form groups of objects with similar characteristics. More recently, clustering has been used for scientific discovery, for instance in medical field to identify cancers clusters, and in environmental sciences where scientists look for associations between pollutants and other factors. The use of clustering algorithms to aid scientific discovery faces several challenges. Firstly, almost all clustering algorithms require the setting of input parameters which is a non-trivial task and choosing proper values for those parameters is critical for obtaining high-quality clusters. Moreover, many clustering algorithms are probabilistic and different runs, even with the same parameters, lead to different results.

K-Means is a typical distributed clustering algorithm based on partition developed a parallel version of K-Means in multiprocessors of distributed memory put forward another parallel version of K-Means, which transfers the clustering center. As the amount of data in clustering center is usually smaller than the amount of data being divided, the traffic is reduced. Proposes another parallel version of K-Means which only transfer statistic data and has a higher efficiency put forward a distributed clustering algorithm. K-D Means which is based on K-Means. The main site divides the data set into k subset randomly and stores these k subsets in k sub site. Each sub site calculates its central point and informs other k-1 sub site of its central point. Each sub site calculates the distance from each data point of its local point set to each central point, and cluster data by the idea that each data point belongs to a cluster whose central point is the closest to this data point among all central points.

C. Visualization

Data visualization is a quite new and promising field in computer science. It uses computer graphic effects to reveal the patterns, trends, relationships out of datasets. To get deeper about it, some discussions about multidimensional data visualization are done. With the combination of some known methods, it presents a new algorithm to do 4 dimensional data visualization. Human has a long history with basic data visualization, and data visualization is still a hot topic today. The history of visualization was shaped to some extent by available technology and by the pressing needs of the time, they include: primitive paintings on clays, maps on walls, photographs, table of numbers (with rows and columns concepts), these are all some kind of data visualization although it may not call them under this name at that time.

Visualization is the graphical presentation of information, with the goal of providing the viewer with a qualitative understanding of the information contents. It is also the process of transforming objects, concepts, and numbers into a form that is visible to the human eyes. When it says "information", it refers to data, processes, relations, or concepts. Here, we restrict it to data. Data visualization is all about understanding ratios and relationships among numbers. Not about understanding individual numbers, but about understanding the patterns, trends, and relationships that exist in groups of numbers. From the point of user understanding, it may involve detection, measurement, and comparison, and is enhanced via interactive techniques and

providing the information from multiple views and with multiple techniques.

Data visualization is closely related to information graphics, information visualization, scientific visualization, and statistical graphics. In the new millennium, data visualization has become an active area of research, teaching and development. Data visualization is a general term used to describe any technology that lets corporate executives and other end users "see" data in order to help them better understand the information and put it in a business context.

Visualized data is frequently displayed in business intelligence dashboards and performance scorecards that provide users with high-level views of corporate information, metrics and key performance indicators (KPIs). The images may include interactive capabilities, enabling users to manipulate them or drill into the data for querying and analysis. Indicators designed to alert users when data has been updated or predefined conditions occur, can also be included. Never before in history data has been generated at such high volumes as it is today.

III. METHODS

A. *Preprocessing*

For preprocessing, we use well known methods from natural language processing: The document corpus is converted to a bag of word model [3] describing the frequency of each word. A typical document corpus consists of 300 to 1000 texts with an average length of 54 words. Each text consists of a title, a number of tags and a description. The texts are usually in English without any additional information or structure. English words are translated using standard dictionaries and orthographic errors are identified and corrected using standard spelling checkers. Then, all documents are transformed to lowercase. Technical sequences such as URLs are removed and country specific characters (e.g., German umlauts) are replaced by corresponding letters from the English alphabet. Each document is then split into a sequence of single words (1-gram model). Words without semantic information (stop words) are removed from the sequence [6], while the semantic content is enriched by comparing the remaining words to an open source synonym data set [7] and replacing synonyms by one word. Finally a German Porter stemmer [9] is applied in order to replace words by their word stem.

The result is used to calculate the TF-IDF matrix that consists of a term frequency part and a normalizing factor which reduces the importance of very frequently occurring terms (IDF part). The inverse document frequency for the term i is defined as the logarithm of the total number of documents N in the corpus divided by the number of documents containing the term n_i:

$$idf(i) = \log(N\ n_i) \quad (1)$$

The element v_{di} of the TF-IDF matrix is then defined as

$$V_{di} = tf\ (i, d) * idf(i) \quad (2)$$

where the term frequency tf (i, d) denotes how often the term i occurs in document d. The resulting N × l matrix (where l stands for the number of terms) gives a description on how often a term occurs in a text. This matrix serves as input for the clustering algorithm.

B. *Clustering and Visualization*

Our clustering method is based on a SVM approach that is able to learn the principal components of the data in a continuous way by LDA method. Several papers in natural language processing already proposed using PCA. However, none of them considered the problem of a growing dataset in a real-time environment, so we come across LDA. For the visualization part, we had to find a way to embed several clusters, that would store topic related or subject related words in those clusters from the document which we define TF-IDF method. The number of clusters taken should be defined earlier as the result visualization is much simpler and easier to understand.

IV. PROPOSED MODEL

In proposed model, we emphasize on the base paper with two different models as of the base paper. Here we focus on K-means for clustering the data and also we use Support vector machine as our learning tool so that understanding of the paper and subject goes easy. Against PCA we are making use of LDA which is also used for dimensionality reduction of the document which will save time as lesser memory will be used.

A. *System architecture*

The System Architecture gives the knowledge of the system flow of the proposed model. There are two phases in the design in the first it shows which is also called backend phase, here the texts are taken which contains some words in the text. Those texts are then redirected to TF-IDF phase where the each text is taken and then it is processed and gives its unique words and weight of the words from the text or document.

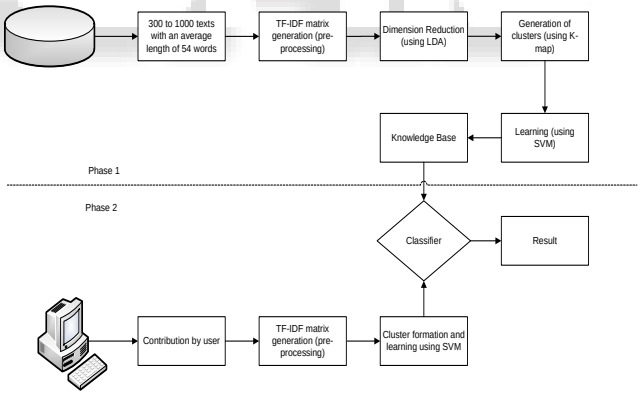

Fig. 1: System Architecture

Then it is processed by LDA, where the dimensionality is reduced of the text document and then it is passed to Clustering, where we define how many clusters we equire for the paper and then we clusters those with some subject related words. With the help of SVM as learning principle, we integrate the 3 modules and store in knowledge base. This all takes place in backend phase. In front end user writes a text document based on question and submits the file. That file is taken and processed with TF-IDF, LDA and K-means and then it compares with knowledge base and gives the result.

B. TF-IDF

Tf-idf stands for term frequency-inverse document frequency, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

One of the simplest ranking functions is computed by summing the tf-idf for each query term; many more sophisticated ranking functions are variants of this simple model. TF-IDF can be successfully used for stop-words filtering in various subject fields including text summarization and classification.

TF-IDF calculates values for each word in a document through an inverse proportion of the frequency of the word in a particular document to the percentage of documents the word appears in. Words with high TF-IDF numbers imply a strong relationship with the document they appear in, suggesting that if that word were to appear in a query, the document could be of interest to the user.

C. LDA (Linear Discriminant Analysis)

Linear Discriminant Analysis, or simply LDA, is a well-known classification technique that has been used successfully in many statistical pattern recognition problems. It was developed by Ronald Fisher, who was a professor of statistics at University College London, and is sometimes called Fisher Discriminant Analysis (FDA). The primary purpose of LDA is to separate samples of distinct groups. We do this by transforming the data to a different space that is optimal for distinguishing between the classes.

LDA is used in statistics, pattern recognition and machine learning to find a linear combination of features which characterize or separate two or more classes of objects or events. The resulting combination may be used as a linear classifier or, more commonly, for dimensionality reduction before later classification. The objective of LDA is to perform dimensionality reduction while preserving as much of the class discriminatory information as possible.

D. K-means

Many types of data analysis, such as the interpretation of Landsat images discussed in the accompanying article, involve datasets so large that their direct manipulation is impractical. Some method of data compression or consolidation must first be applied to reduce the size of the dataset without losing the essential character of the data. All consolidation methods sacrifice some detail; the most desirable methods are computationally efficient and yield results that are at least for practical applications representative of the original data. Here we introduce several widely used algorithms that consolidate data by clustering, or grouping, and then present a new method, *k*-means algorithm, developed at the Laboratory specifically for clustering large datasets.

Clustering involves dividing a set of documents into non-overlapping groups, or clusters, where text in a cluster are “more similar” to one another than texts in other clusters. The term “more similar,” when applied to clustered points, usually means closer by some measure of proximity. When a dataset is clustered, every text is assigned to some cluster, and every cluster can be characterized by a single reference point, usually an average of the text in the cluster. Any particular division of all texts in a dataset into clusters is called a partitioning of clusters. The Clusters are formed after this partitioning process depending on the Crowdsourcer. Crowdsourcer is the one who gives the number of clusters to be done for the Process.

V. IMPLEMENTATION

In Proposed Model, the dataset is taken and trained with SVM training. Once the dataset is trained it is stored into knowledge base. The users give answers which are stored and are taken as documents for TF-IDF. Once the TF-IDF is done the weight of the words are noted or the weights are listed. After the TF-IDF method the document needs to reduced Dimensionally so the LDA is used to reduce the size. Once the size is reduced it is processed by K-means algorithm and then clusters are formed. With SVM classifier it classifies the document and stores in the cluster to which the text document is related to depending on the query.

VI. CONCLUSIONS

In this paper, it is presented that novel method for information retrieval adapted to a hard problem in text data analysis with low and instable information content of the single documents and a high dimensional and increasing feature space. The method outperforms standard clustering algorithms with respect to classification reliability, has the potential for real-time classification during a crowdsourcing process, and provides an innovative visualization for dealing with the problem of dimensionality reduction. In order to implement and validate the method for real world applications, further steps are needed: First, the clustering has to be validated with respect to accuracy compared to manual clustering. Second, tests must demonstrate the positive effects of using our methods with respect to solution space exploration and improvement of the innovation process through crowdsourcing.

ACKNOWLEDGMENT

We would like to thank Director Dr. H N Shivashankar, Principal Dr. M K Venkatesha and Dr. M V Sudhamani, professor and Head, Dept of Information Science and engineering, RNSIT, for constant encouragement in for implementing this project and pursuing this paper.

REFERENCES

[1] D. Jurafsky, J. H. Martin. Speech and language processing. London (Prentice Hall), 2009.
[2] D. M. Blei, A. Y. Ng, M. I. Jordan. Latent Dirichlet Allocation. In Journal of Machine Learning Research 3, pp. 993–1022, 2003.
[3] M. Steyvers, T. Griffiths. Probabilistic Topic Models.In Latent Semantic Analysis: A Road to Meaning.(Lawrence Erlbaum),2007.
[4] A. Skupin, S. I. Fabrikant. Spatialization Methods: A Cartographic Research Agenda for Non-geographic Information Visualization.In Cartography and

Geographic Information Science 30(2), pp. 95–115, 2003.
[5] S. Haykin. Neural networks. A comprehensive foundation. London(Prentice Hall), 1999.
[6] S. Bird, E. Loper, E. Klein. Natural Language Processing with Python. Sebastopol (O'Reilly Media Inc.), 2009.
[7] D. Naber. OpenThesaurus: ein offense deutsches Wortnetz. In Beitr¨age zur GLDV-Tagung, pp. 422,433, 2005.
[8] J. Rissanen. Modeling By Shortest Data Description In Automatica 14, pp. 465–471, 1978.
[9] M.F. Porter. An algorithm for suffix stripping, In Program 14(3), pp. 130–147, 1980.
[10] T. Ott, A. Kern, A. Schuffenhauer, M. Popov, P.Acklin, E. Jacoby, R. Stoop. Sequential Superparamagnetic Clustering for unbiased Classification of High-Dimensional Chemical Data. In J.Chem.Inf.Comput.Sci. 44, pp. 1358–1364, 2004.

