Multi Agent Distributed Clustering Framework for Large Scale Social Data: Twitter Case Study

Shaheen Khatoon

College of Computer Science and Information Technology

Department of Information System

King Faisal University, Al Ahsa, Saudi Arabia
shaheenkhatoon@acm.org

***Abstract*- Social data refer to data created and intentionally shared by individuals and has a huge potential to gain insight of user's interpersonal behavior and societal patterns. However, such data is often huge, incomplete and heterogeneous which make data analysis task extremely challenging. Application of traditional centralized approaches to analyze such data often ended up with low throughput and response time. One way of reducing the computational complexity of analysis task through the application of distributed computing, However, typical distributed environment offers bandwidth limitation problem for the intense analysis task that require frequent data exchange and interaction with central site. This paper presents distributed multistage data clustering for analyzing large scale social data by utilizing Multi- Agent Systems (MAS), where each agent is independent system and capable of making decision at local level. Instead of sending the entire data to central site they send compact data models which can be integrated to get a global view of data, hence reduced data interaction between agents and central sit. Scalability and efficiency of model is demonstrated by using big corpus of twitter data. Particular issue address is detecting tweet clusters from particular area of interest by determining the relevant locations associated with the tweets. The obtained results are promising and highlight the efficiency and accuracy of proposed framework.**

I. INTRODUCTION

With the advent of Web 2.0, ubiquitous computing and advancements in corresponding technologies, social media has drastically changed the concepts of information contribution, dissemination, and exchange [1]. Increasing use of mobile phone communication, e-commerce transaction and Social Network activities such as Tweet, Facebook, blogs has enables virtually every citizen a potential contributor or user of information. People can easily share opinion, experience, expertise and other contents at the push of single button. These different kind of contents can be leveraged in order to make interesting and valuable inferences from social networks. For example, user activities can be collected and analyzed to identify individual behavior and societal pattern which can be used for policy making, strategic planning decision making, target marketing as well as behavior influence. However, ability to extract knowledge from these large complex heterogeneous data sources is still limited due to data size, dynamism and noise. Use of traditional framework of centralized data analysis and mining involves the batch processing of large static dataset which is not suitable for processing high volume stream. Application of such approaches on large scale real time data result in insufficient throughput to process high volume social streams. One way of reducing the computational complexity of analysis task through the application of distributed computing, by partitioning the data across multiple machines to achieve desired throughput and reduce response time. However typical distributed environment offers bandwidth limitation problem for the intense analysis task that require frequent data exchange and interaction with central site. Distributed Data Mining (DDM) can be used to perform data analysis and mining operation in fundamentally distributed manner where each site can send compact representative model of data instead of sending entire raw data. This approach can sufficiently reduce the data transfer rate, however each distributed site has to communicate with central site to get environmental information to optimize their operation, this does not scale well in presence of large scale network such as sensor network where hundreds of sensors are deployed in environment. Multi-Agent Systems (MAS) can be another option to decouple distributed site with central site. MAS deal with complex applications that require distributed problem solving, where each agent is autonomous which can perceive their environment, dynamically reason out actions based on conditions, and can interact with each other. Since MAS are also distributed systems, therefore, combining DDM with MAS for large scale data set (multimillion users) is active area of research for data analytic community.

This paper undertook the synergy between MAS and DDM, particularly focusing on distributed data clustering to analyze large scale social data. The proposed framework partition data across multiple independent machines (agents) for parallel clustering to achieve computational efficiency. Each agent builds a local clustering model based on local data and lower level coordination between agents is achieved by using Message Passing Interface(MPI). To reduce the communication between agents and central site approximation of each cluster is sent to central site where local models are integrated to computing global model which provide overall data clustering. Main contribution of this includes:

- Detail literature review on distributed data clustering and identify the challenges in different approaches.
- Proposed a conceptual framework which can improve the combined power of multi agent systems with local and global data clustering techniques.
- Implementation and validation of proposed framework on Twitter case study for clustering micro- messages.

The motivational example for MAS and DDM scenario is for whatever reason it is not practical for data to bring together at single repository. Therefore, each agent has its own personal data repository containing records different from those held by other agents (i.e. the data sets are disjoint). Agents can communicate with each other using MPI to further improve initial data partition created at initial stage. It is the message passing approach in proposed framework to harness the true potential of MAS apart from other typical MAS approaches used to achieve distributed clustering.

Rest of paper is organized as follow: Section II provides in-depth literature review on distributed multi agent based clustering and highlights strength and weaknesses of existing studies. A general framework is proposed in section III for distributing data analysis task across different machines to achieve both scalability and efficiency. In Section IV a case study from twitter is presented to demonstrate the validity of proposed framework. Finally, Section V concludes the paper.

II. Related Work

To increase the scalability of data Ahmed et. al [2] used the enhanced version of DBSCAN to discover clusters from large scale uneven data set. Scalability is achieved by applying K-mean to get an initial partition. Subsequently enhanced DBSCAN is applied on each partition and finally merge process is used to find actual number of clusters in large dataset. Algorithm is scalable in the sense that instead of inspecting the whole dataset, the EDBSCAN searches within the objects of each partition. The algorithm is beneficial because of its capability to discover varied density clusters.

To improve the configuration of initial clustering S. Chaimontree [3] proposed a Multi Agent Based Clustering (MABC) framework where cluster agent bid for a record and data agent (owner of data source) act as auctioneer to build initial clustering. In next pahse cluster agent try to pass out unwanted record by inviting other cluster agents to participate in bidding process and act itself as local auctioneer. This phase further improves the preliminary clustering configuration. A metrics approach is used for measuring the goodness of a cluster.

Swarm intelligence based clustering for solving the problem of group formation and task allocation in a distributed approach is used in [4]. The approach is not reliant on the preliminary information about quantity of classes, number or volume of partitions. Furthermore, without the usage of original data points, the algorithm aims to create a single best clusters or group of agent based upon similar or complementary expertise and abilities. For the purpose of visualization of large scale social data Z. Wang et al. [5] used divide-analyze-recombine approach by performing data partition, subset clustering and result recombination in integrated fashion. User tags and behavior information are identified to find link with other users. This technique provides good performance and efficiency but formulation of user personal data and behavior information into a point data set offers complexity for this approach.

D. Dehideniya et al.[6] proposed multi-agent based clustering algorithm for dynamic dataset by using two types of agents called Data Record and Cluster Agents. Data records agents represent the single data record to get the membership of nearest cluster agent. Cluster agents communicate and negotiate with record agent to accept or reject data record based upon data variance. Cluster agents which are too close to each other can be merged by using Silhouette coefficient internal cluster validation measurement. In 2D spaces the proposed method is trapped with the problem of local maxima, so real number of clusters are not accurately determined.

Wang et al. [7] presented an enhanced clustering method based on density peaks which incorporates both structural and attribute information of users for social circle discovery in social network. Gaussian kernel is used for the construction of density estimation to circumvent big numerical faults and discover overlapped social circles keeping in view the distances among users in dissimilar social circles. Authors utilized Balanced Error Rate (BER) and F1-Scoreas metrics for comparisons purpose.

García et al.[8] implements a hybrid approach by combining agent base simulation and clustering for simulating sociogram called Agent-Based Simulator (ABS). It uses clustering to classify individual in certain group based on psychological characteristics. BY using ABS practitioner can classify each new individual according to their psychological characteristics such as in secondary education teachers can predict aggression and victimization of student and can intervene in time to bring the positive change. The validation experimentation concluded that result generated by simulated sociograms are similar to the real sociograms.

In order to cluster large scale complex network Z. Li et al. [9] proposed genetic algorithm based multi agent approach MAGA-Net to optimize modularity value of community detection. Agents are represented by using locus-based adjacency representation to automatically generate communities. A series of operators are designed, namely split and merging based neighborhood competition operator, hybrid neighborhood crossover, adaptive mutation, and self-learning operators to increase the modularity. This approach finds best partition in less time but is limited by the design of a series of operators for increasing the performance of MAGA-Net.

The methods mentioned above are ideal for simple task distribution, by partitioning the predefined datasets across multiple machines. Most of them are using store and process model of data analysis which is not suitable for real time continuous data streams. Proposed framework is differing in the sense that it is capable of handling real time large scale social media data which is often noisy. Furthermore, each distributed site is autonomous and interaction with central site is minimum.

III. Proposed Multi Agent Clustering Framework

In this section a conceptual framework for clustering large scale social data is proposed. The framework utilizes a multi agent based distributed environment for the discovery of global clusters based on the local clusters produced by multiple agents. The model shows the overall framework in client/server paradigm for the purpose of illustration as shown in Figure 1. It is a generic framework in the sense that one can use their own choice of clustering algorithm at local as well as global level.

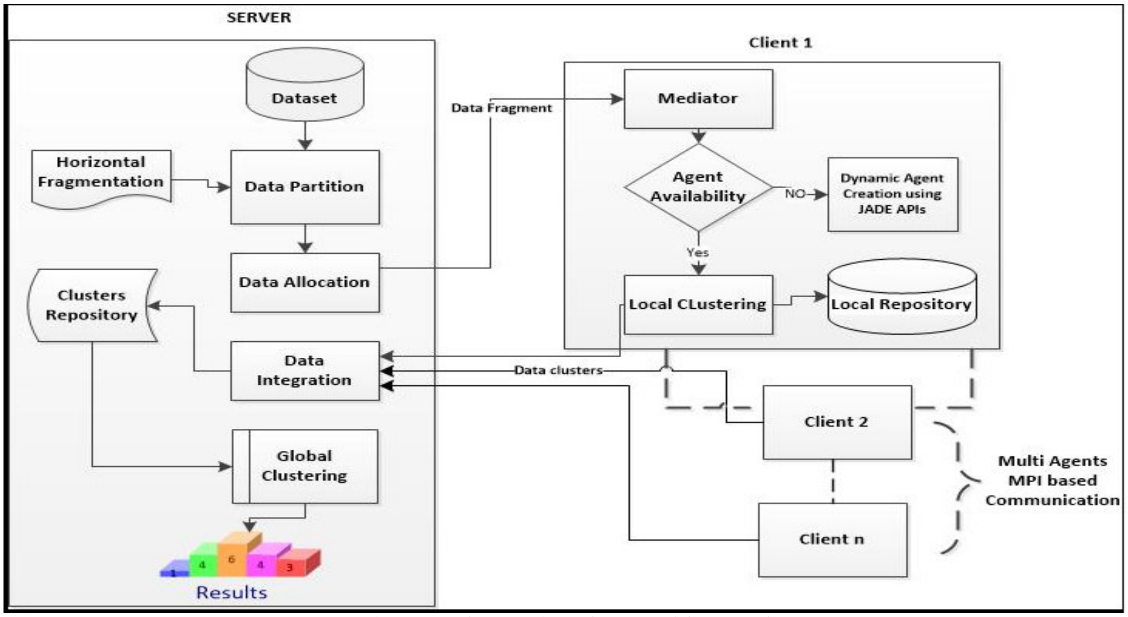

Figure 1. Abstract view of proposed framework

Proposed framework composes of following main modules:

A. Data Collection and Preprocessing

Original social media data can be retrieved from source data providers through queries. This entails submitting a query in the form of an http request and receiving in response data in XML format (e.g. Atom or RSS). The query parameters may be for example, based on location (e.g. specifying an area of interest to which the feed is related), time (e.g. specifying a period of interest), content (e.g. specifying keywords or hashtags), or even user handle/ID.

Twitter is a representative example of social data, where the data received in response to a query are actual tweets and associated metadata (e.g. user information, time of tweet publication, geolocation when available, and information on whether this particular tweet is in response to or retweet of an earlier message. Data collected from such sources is often noisy, therefore appropriate preprocessing task are require to bring the data in a format adoptable by given data mining algorithm.

B. Data Partitioning and Allocation

The main purpose of this step is to partition the dataset into smaller datasets, since most algorithms perform well with small datasets. So, this stage mainly improves the scalability of large dataset. Most important issue here is getting initial partition. To full fill this any clustering algorithm with time complexity O(n) can be used depends upon type of data. This study is evaluated on twitter dataset to identify trending topics in given region therefore, horizontal data partitioning [10] is adopted, where every partition will be having all the attributes but the total data records will be divided among agents. It will promote parallelism and each agent views only the data related to it not the entire application.

In order to achieve balance distribution of data at each agent, data allocation module at server is developed which allocate data to agent using non redundant allocation strategy. Upon receiving data from server mediator module at client will check the availability of agent to perform local clustering. If agent is not available new agent is created and registered into agent pool. To facilitate dynamic agent creation, the proposed model is implemented in JADE [11]. The inter agent communication is achieved by using Message Passing Interface (MPI).

C. Local Clustering at Agents level

At this stage subset of data is clustered using any clustering algorithm of choice. Local clustering will be efficient since it will perform on reduced set of data called partition. To further improve the initial partition agent frequently broadcast their information to environment. Any data record can join the neighboring agent if it is closely related to it as compare to previous group. Two agent can be merged to single agent if their data is closely related. This process of splitting and merging will continue until optimal number of clusteres are achieved at each agent.

D. Data Integration at Central Site

After the construction of local clusters, they are sent to server for global clustering. Instead of sending entire data, center of each cluster and the distances between each object within a cluster are obtained. DBSCAN [17] is a type of density based algorithm selected for tweet clustering. In DBSCAN clusters are identified as dense region of data objects surrounded by low density regions where density is evaluated *Eps* radius and MinPts (Minimum points) specified by users. A cluster is formed when the number of neighbors is greater than or equal to *MinPts* with in *Eps* distance. If the number of data points is less than minimum points in given distance the point is marked as outlier.

IV. EXPERIMENTAL EVALUATION

Proposed framework is validated on dataset collected from twitter. The reason of choosing twitter due to its popularity and currently one of the largest social media producing over 160 billion posts each year and has been shown to be a good source of information about events as they happened[13]. Hence, Twitter dataset is used containing a sample of million tweets from the beginning to middle of September 2016.

By utilizing proposed framework an event observation system is developed capable of collecting recent events or topic under discussion in given territory. The goal of this case study is solely to demonstrate validity of proposed framework by using twitter data. This show how framework can be used to automatically obtain current topic of interest in given geographical location and cluster the corresponding tweet under certain group efficiently. By doing so government agencies can continuously monitor nationwide current topics/events and able to identify unusual event for timely intervention. The key idea is that people in same geographic proximate often tweet about same event and their instant updates would be the starting point to perform the global real-time search, mine, statistics, and further social analysis. The purpose of using tweeter dataset is to evaluate efficiency of proposed framework by distributed data across different agents collected from twitter.

A. Data Collection and Preprocessing

The process of data collection is start with real time extraction of tweets. To construct our evaluation dataset, we used *R* to scrapped publicly accessible data from twitter using its open APIs. For each tweet following information are extracted: tweet location (specified by longitude and latitude), time-stamp (most recent tweets) and message detail. Tweets collected through API contain different attributes such as tweet itself, how many times it was re-twitted and favorited, the date and time it was twitted, etc. All of them is not required for subsequent steps. Therefore, ETL process is designed to select required attributes such as Tweet Id, Text and geolocation. Further filtering is done on tweet dataset such as removing missing values since there are some tweets which only have double quotes in it so we process and filter out. To make sure data is ready for analysis a series of transformation such as transform some attributes into text so that text processing can handle it. To extract more value out of text further cleaning steps are performed subsequently to construct lexical corpus and term documents matrix e.g. tokenization to convert sentence into set of tokens, punctuation, numbers and URLs are removed, characters appears more than twice are stripped in order to correct spelling (soooo is converted to so). Finally stop words are removed and all terms are stemmed using standard stemmer and term n-gram is generated.

To identify topic of interest in certain area it is required to get full address of each tweet. In fact, raw text data for location is scraped from twitter. From textual we need to perform geo-coding to identify the exact coordinates by translating places name into the corresponding exact locations. In order to do so, google map API ggmap [14] are used. The ggmap package enable us to get state, city, zip code and street address of tweet. However, it returns result in text format. Regular expression and string manipulation are used to separate street address, city, zip code, province and country in column. Finally, term document matrix with above additional field is created. Now data is ready for clustering having geographic information and weighted term to be used by proposed framework.

In next phase proposed multi agent based framework is utilized to perform subsequent analysis tasks on preprocessed data. Data preprocessing steps are shown in Fig. 2. The tweet from Saudi Arabia are collected by setting geolocation of different cities.

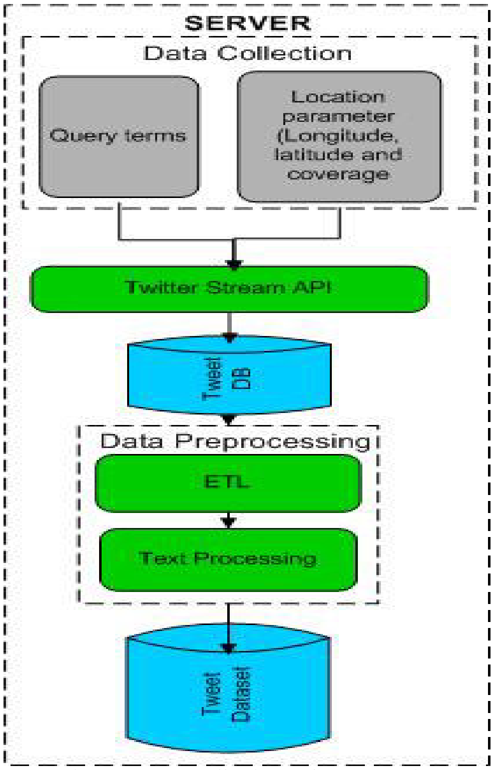

Figure 2. Data collection and preprocessing

B. Data Partitioning

To detect local events for a given large area, we first determined how to partition the target region into sub-areas. One way of partitioning data is grid-based partitioning [15] which allow to partition data into equal sized cells. In this approach identifying size of cell is difficult, For example, splitting a region into excessively small cells may result in poor clustering. Since we did not consider the geographical distribution of tweets, the balance over the target region becomes inefficient and consequently results in a poor event detection. The other way is horizontally partitioning on the basis of administrative unit, city or town. In this case if two neighboring regions are strongly connected to each other in term of current social event simply splitting them in different cluster is not good choice. However, in proposed approach this problem can be solved at agent level clustering, where agent can frequently broadcast their information in environment. Based upon closely relatedness or dissimilarities cluster can merge and split. Therefore, horizontal partitioning is adopted to partition data at initial stage.

C. Local Clustering

At each agent, local clustering is performed. The main purpose here is to automatically group tweets into sets of tweets, such that each cluster contains tweets pertaining to a specific topic in given region. Along with each cluster we associate a feature vector (i.e., weighted list of keywords) using Term Frequency and Inverse Document Frequency (TF-IDF) [16] measure. The TF-IDF is a statistical measure of weight to determine how important a term is in a given corpus, by using a vectorial representation. The importance of each term increases proportionally to the number of times this term appears in the document (frequency) hence, relevance of specific words for each tweet can be identified. The process of local clustering is shown in Figure 3.

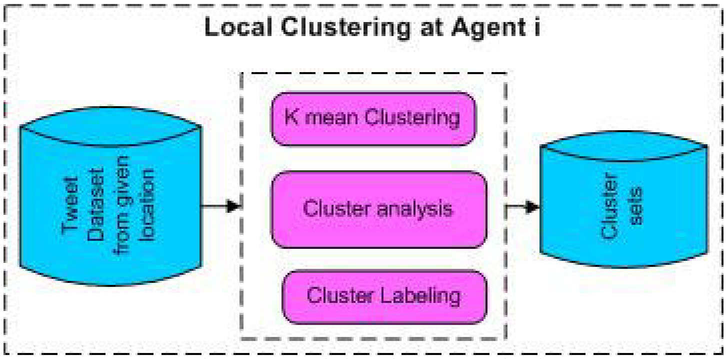

Figure 3. Local clustering at Agent level

Local clustering is achieved by using K- mean clustering [15] which is one of the most popular iterative clustering algorithms, in which the number of clusters *k* has to be fixed a-priori. K-means chooses *k* different centroids and associates each data item to the nearest centroid. *K,* new centroids are then re-calculated and the process is repeated iteratively. Similarity matrix used by K-mean is constructed by using TF-IDF representation of tweets with intra-tweets similarity calculated by means of the cosine similarity measure. Here each tweet represent one document and idea behind this clustering is that tweets which have relatively large number of similar keywords are placed in same group. These words are the relevant words for the cluster based on the TF-IDF weight. They occur with higher frequency in tweets in the cluster than in tweets contained in other clusters.

Result shown in Figure 4, are drawn from one of local agent based upon centroid and corresponding data point. In subsequent phase each cluster is labeled with most frequent keyword among the members in give cluster, and we then compute the average accuracy of such assignments by comparing with manually classified tweets.

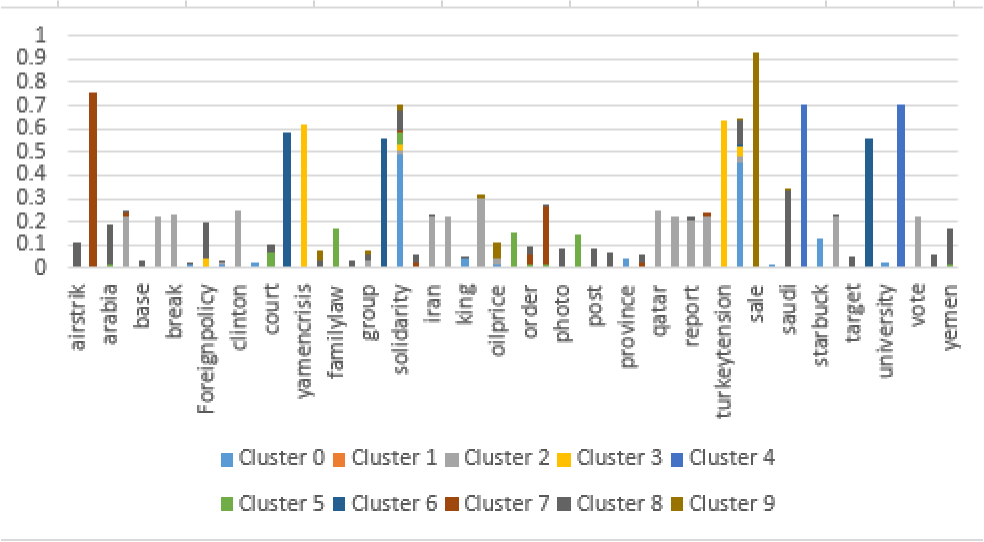

Figure 4. Clustering result at agent I.

Figure 5, shows Inter-cluster distance and the average Intra-cluster distance of cluster set created at local agent. These measures described how good data is separated, higher value of inter-cluster centroid mean better separation whereas lower the value in the same cluster mean data points with in cluster are more similar cluster.

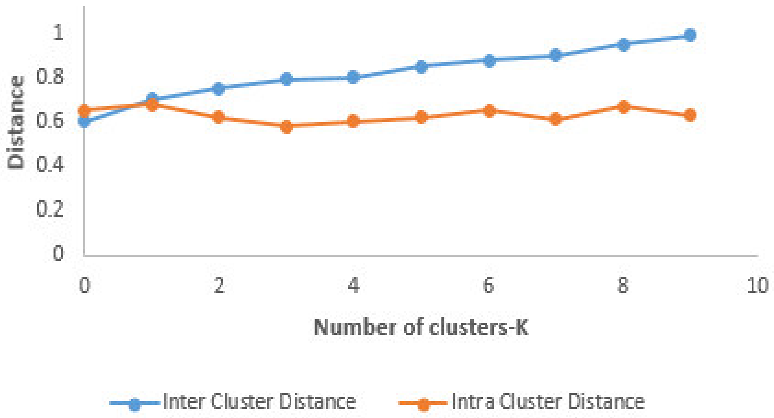

Figure 5. Evaluation of local clustering

To measure quality of clustering we classify 1500 tweets from our dataset manually. In order to manually classify tweets, clustering task is reduced to keyword clustering. The manually grouping of tweets use to evaluate the algorithm directly on tweets and to verify the conformity between the vector based K- Mean clustering and content of individual tweet. After manually classifying tweets quality is measured by using precision, recall and F1-score [18] as shown in Figure 6. Result shows we are able to achieve good precision, recall and F1-scores.

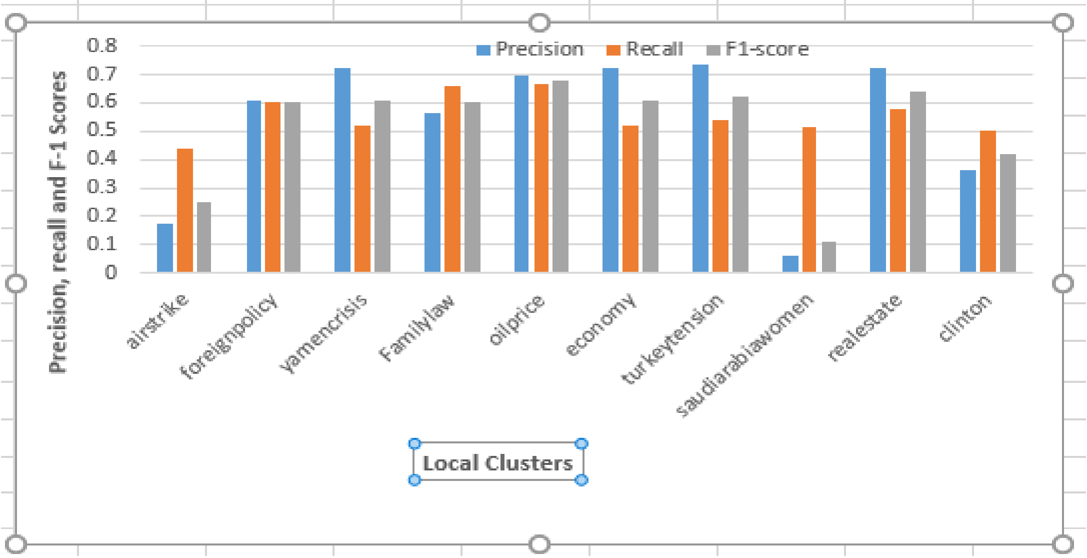

Figure 6. F1 score of local clusters

D. Global Clustering

After local clustering is complete, each agent sends its data to central site to achieve overall clustering. Instead of sending entire data a small set of representative data points such as centroid, number of points in cluster, average distance of each point from center etc. are sent to central site, which combines the local cluster descriptions to produce global clustering description. The merge function is implemented, on the union of the local representative points to form the global clustering utilizing DBSCAN algorithm. DBSCAN uses some similarity metric such as cosine similarity, in the form of a distance, to group data points together. If points lie outside specified distance the point is marked as noise. It need two input, minimum number of points (*Minpts*) in a dense cluster, and distance (*Eps*). DBSCAN visits every data point in the dataset and draws an *Eps* radius around the point. If there is at least *Minpts* number of points in *Eps* radius, the point is called dense point. Otherwise point is mark as noise and removed from clusters. By varying the Eps and MinPts optimum number of clusters are achieved. The entire multistage clustering process is shown in Figure 7. To visualize the clusters created by the DBSCAN algorithm, the results were mapped onto Google Maps using Google Map API 3.0.

Figures 8 shows global clusters for top 10 over all clusters of topics collected during middle of Sept. 2016 across Saudi Arabia.

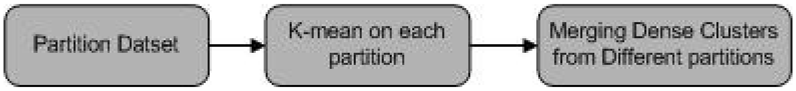

Figure 7. Multistage Clustering process

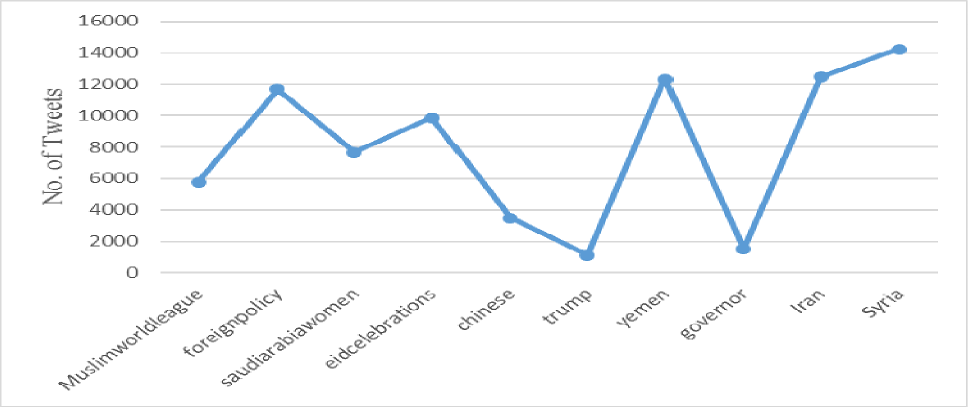

Figure 8. Figure 5. Top 10 cluster showing overall popularity of topics

V. Conclusion and Future Work

We have proposed a general scalable and efficient multistage general framework for analyzing large scale data. The efficiency and scalability is achieved by utilizing distributed clustering over multi agent based architecture. Through experimentation on a large corpus from Twitter dataset, we evaluated the effectiveness, efficiency, and scalability of the proposed framework. The distributed clustering approach has shown better result to automatically group tweets into sets of clusters, such that each cluster contains tweets pertaining to a specific topic in given region. Although we have evaluated Twitter dataset in context of common topic identification in given region, it can be equally applied for extraction of other concepts such as monitoring disease outbreak, product promotions on certain areas, sport events etc. Furthermore, clustering algorithms used at each level can be replaced by other clustering algorithms, depending on the nature of the desired clusters.

In future framework, will be adopted for analyzing data from real time applications such as sensor networks data. Furthermore, efficiency and accuracy of framework will be further evaluated by varying clustering algorithm according to requirement of application.

REFERENCES

[1] A. M. Kaplan and M. Haenlein, "Users of the world, unite! The challenges and opportunities of Social Media," *Business horizons,* vol. 53, pp. 59-68, 2010.

[2] A. Fahim, A.-E. Salem, F. Torkey, M. Ramadan, and G. Saake, "Scalable varied density clustering algorithm for large datasets," *Journal of Software Engineering and Applications,* vol. 3, p. 593, 2010.

[3] S. Chaimontree, K. Atkinson, and F. Coenen, "A framework for multi-agent based clustering," *Autonomous Agents and Multi-Agent Systems,* vol. 25, pp. 425-446, 2012.

[4] D. S. Dos Santos and A. L. Bazzan, "Distributed clustering for group formation and task allocation in multiagent systems: A swarm intelligence approach," *Applied Soft Computing,* vol. 12, pp. 2123-2131, 2012.

[5] Z. Wang, C. Chen, J. Zhou, J. Liao, W. Chen, and R. Maciejewski, "A novel visual analytics approach for clustering large-scale social data," presented at the Big Data, 2013 IEEE International Conference on, 2013.

[6] D. Dehideniya and A. Karunananda, "Dynamic partitional clustering using multi-agent technology," presented at the Advances in ICT for Emerging Regions (ICTer), 2013 International Conference on, 2013.

[7] M. Wang, W. Zuo, and Y. Wang, "An improved density peaks-based clustering method for social circle discovery in social networks," *Neurocomputing,* vol. 179, pp. 219-227, 2016.

[8] I. García-Magariño, C. Medrano, A. S. Lombas, and A. Barrasa, "A hybrid approach with agent-based simulation and clustering for sociograms," *Information Sciences,* vol. 345, pp. 81-95, 2016.

[9] Z. Li and J. Liu, "A multi-agent genetic algorithm for community detection in complex networks," *Physica A: Statistical Mechanics and its Applications,* vol. 449, pp. 336-347, 2016.

[10] N. B. Osman, "Extending the Technology Acceptance Model for Mobile Government Systems," *development,* vol. 5, p. 16, 2013.

[11] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing multi-agent systems with JADE* vol. 7: John Wiley & Sons, 2007.

[12] Y. Lv, T. Ma, M. Tang, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, "An efficient and scalable density-based clustering algorithm for datasets with complex structures," *Neurocomputing,* vol. 171, pp. 9-22, 2016.

[13] R. Bandari, S. Asur, and B. A. Huberman, "The pulse of news in social media: Forecasting popularity," presented at the ICWSM, 2012.

[14] D. Kahle and H. Wickham, "ggmap: A package for spatial visualization with Google Maps and OpenStreetMap. R package version 2.3," *See* http://CRAN. *Rproject. org/package= ggmap,* 2013.

[15] R. D. Johnson, "Gender Differences in E-Learning: Communication, Social Presence," *Innovative Strategies and Approaches for End-User Computing Advancements,* p. 175, 2012.

[16] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management,* vol. 24, pp. 513-523, 1988.

[17] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," presented at the Kdd, 1996.

[18] C. D. Manning, P. Raghavan, and H. Schütze, "Flat clustering," *Introduction to information retrieval,* pp. 350-374, 2008.