Mining Social Networks of Other Languages/Cultures

Ahmed Sameh
Prince Sultan University
Riyadh 11586, Saudi Arabia
Sameh.aucegypt.edu@gmail.com

Abstract. *The universal nature of Social Networks and its wide spread in the whole world allowed users to post their contributions in their own languages. In this paper we are using a combination of language translation and text-mining techniques to understand other cultures in Social Networks that are using other languages of interest. In particular we are studying the case of Twitter and focus on two cultures: Hebrew-speaking and Persian-speaking in the domain of politics. We propose a new Twitter Client that collects live tweet streams on frequent time intervals from #tags of specific topics of the target language and compute a mix of qualitative and quantitative indicators to measure Opinion spread. The proposed Client provides three levels of analysis: Surface, Shallow, and Deep depending on analyst wish and the traffic rate of input streams for the purpose of providing real-time mining service. Hebrew, Persian and Arabic tweets are analyzed taking into consideration important context background information. A prototype of the proposed Client is implemented on-top of "Nodexl"; an open source template for Microsoft Excel that allows automated connection to a social network server and import (Using Twitter APIs) any data stream into the usual Excel environment. Tweet translation, coloring and Edge coloring algorithms are implemented as Excel Macros with selective setting to either surface, shallow, or deep analysis parameters. Visualization graphs are provided that allow dynamic filtering, vertex grouping, adjusted appearance (zoom into areas of interest), graph metric calculations, etc.*

Keywords. Twitter Client, Graph Coloring, Q-Word Net, Arabic Corpus, Nodexl

1 Introduction

Understanding other cultures through participating in and analyzing their Social Networks' posts is of great benefit. Digging down to the actual/raw people's opinions and posts will avoid governments' and media manipulation and hiding of issues. For example Social Networkers in the Arab world read and post in both Arabic and English. Arabic since it is their native tongue and English since it is a universal language. They have no way of participating in any other foreign language social network. In this paper we propose a new Tweeter Client that provides instant translate of posts of any other language/culture into Arabic and vice versa. This Client allows Arabic Social Networkers to participate in other language/culture Social Networks as if they are native members of the language/culture. Arab participant will be able to collect live tweet streams on frequent time intervals from #tags of specific topics of the target language/culture and compute a mix of qualitative and quantitative indicators to measure Opinion spread and opinion spectrum of the #tag of interest. The proposed Client provides three levels of analysis: Surface, Shallow, and Deep depending on analyst wish and the traffic rate of input streams for the purpose of providing real-time mining service. As case studies, Hebrew and Persian tweets are analyzed taking into consideration important culture context background information (since they are neighboring cultures to the Arab world). We use Google translate service to translate all Tweets to Arabic and then focus only on "Arabic" Tweet analysis. The proposed Client also introduces three "Tweet Coloring" algorithms that make use of previously developed Arabic NLP tools and graph algorithms such as "Arabic Wordnet", "Q-WordNet" ontology [1] for sentiments, "Arabic Lexicons", "Arabic Tweet Corpus", "Max Flow Minimum Cut" [1] in order to speculate Tweeters' inclination and impression about the #tag subject of interest. On the other hand, three "Edge Coloring" algorithms are also introduced to speculate on opinion, influence, and trust spread (cascading) through Twitter social network graphs. Each one of the three algorithms has a "binary" version (+ve/-ve, Yes/No) and a "continuous" version. The algorithms compute inferred value ratings through polling neighbors and weighted averaging. In case a user wants to reply or comment on a particular Tweet in the target language, the proposed twitter Client provides an automatic translation facility that posts reply tweets in the target language. Through the analysis component of the Client, the user can understand the #tag current posts from the other language/culture network. Then using the auto-translation component, the user can post replies and tweets that are deep in meaning and participation in the target network. Making use of web services such as the data mining, text mining and language translation within the social network has been studied in our previous work [1]. In that works we claim that future social networks will be focused and specialized mixture of both context and traditional web services to enhance social networks and provide richer and deeper environments. The analysis component of the proposed Client is based on our previous work in [2].

The structure of this paper is as follows: section 2 describes the architecture of the proposed Client, then section 3 details the "Analysis" component showing the both the tweet, edge coloring, and translation algorithms. In section 4, the "Synthesis" component is described. Section 5 describes the implementation details of the proposed Client, then section 6 shows the experimentations of the "Hebrew" and "Persian" testing of the system. In section 7, we present the conclusion and future work.

2 The proposed Client Architecture

Figure 1 shows the architecture of the proposed Twitter Client and its interaction with both the Twitter server and the Google server. The Client has two components: an analytic component and a synthesizing component. The "Analytic Engine" communicates with the Communication Module to call the Twitter Server Backend using Twitter APIs to retrieve qualified Tweets for analysis. The retrieved Tweets are passed by the Engine to appropriate analysis module according to the End user needs. Built-in analysis algorithms are called upon to process these Tweets and display their results either through the Visualization Module or the Graph Analysis Algorithms. As shown in the figure, there is interaction among the analysis algorithms and they call each others for intermediate processing. Trend Analysis is a module that accumulates analysis results over a period of time and performs trend analysis for future prediction. The Synthesizing Component allows participating in the target culture/language through automatic translation.

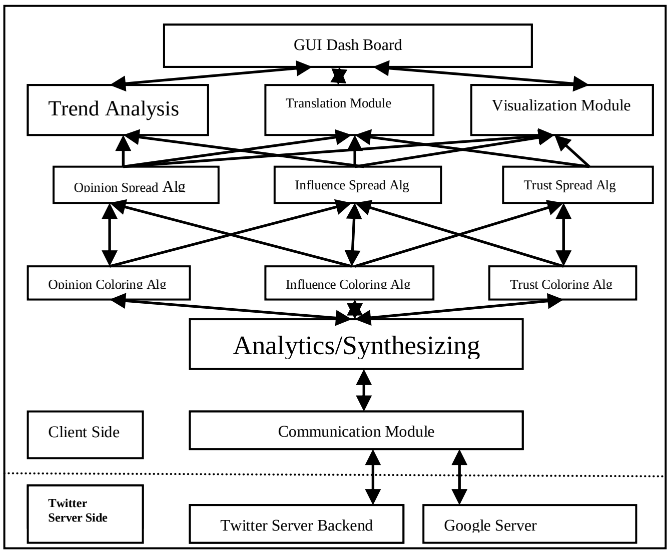

Figure 1: System Architecture –Analytics / Synthesizing Components

3 Client Analysis Component: Tweets, Edges Coloring, Translation, and Text Mining

3.1 Tweet Coloring and Text Mining

Tweets are text of no more than 140 characters. We use text mining algorithms to discover and extract knowledge implicitly embedded in the body of a Tweet or during an exchange of Tweets of unstructured text. Text mining is based on "Natural Language Processing- NLP" and "Information Retrieval-IR". On the other hand, an opinion (or regular opinion) is simply a positive or negative sentiment, view, attitude, emotion, or appraisal about an entity or aspect of the entity. Opinion orientation is positive, negative or neutral. The goal of an opinion is either speculative, persuasive, impression, and/or inclination. Opinions matter a great deal in understanding other cultures. In this work we focus on understanding what target participants are thinking, opinions' attacks, support, etc. Tweet coloring expands on the idea of opinion orientation, and borrows from Web page ranking. We apply similar ideas using text mining algorithms to color tweets according to the degrees of support or non-support for a certain issue. We collect live tweet streams on frequent time intervals from hash tags and apply these tweet coloring algorithms to color tweets to measure opinion support or non-support for the issue under investigation. We provide three levels of analysis: Surface, Shallow, and Deep depending on analyst wish and the traffic rate of input streams for the purpose of providing real-time mining service. English, French and Arabic tweets are analyzed taking into consideration important context background information (e.g. culture of the country, political background, nature of the election process, etc.). We focus more on "Arabic", and present three "Tweet Coloring" algorithms that make use of previously developed NLP tools and graph algorithms such as "Arabic Wordnet", "Q-WordNet" ontology for sentiments, "Arabic Lexicons", "Arabic Tweet Corpus", in order to speculate participants' inclination and impression about the target issue.

3.2 Surface Analysis

At this level of analysis we apply word-level text mining algorithm called "Word Bag". In this algorithm we have training and testing phases. In the training phase we analyze a collection of Tweets from the Tweet stream and build two word bags "Positive Words Bag" and a "Negative Words Bag" (sentiment words such as great, excellent, horrible, worst, are used to identify which bag to color the Tweet, etc.). This is kind of training for building an "Opinion" corpus. The need for this step is evident since Tweeters' language is completely different from standard everyday life text language. Tweeters use their own "slang" language and their own special words and symbols. That is why it is important to build a special corpus for this Tweets language, we called it "opinion" corpus. Figure 2 shows the details of the Tweet surface coloring bags. In the first step of the coloring algorithm, Tweet clustering is used to reduce the dimensionality and categorization of the training tweets according to the target domain. Unsupervised machine learning is used at this step. Tweet pre-processing is the next step, where text cleanup is performed (un-important stop words are deleted), followed by "Tokenization" where roots (stems) of words are produced, followed by POS (Part of Speech Tagging) where syntactic tagging is applied, and finally "word sense disambiguation" is applied. In the training phase a teacher (supervised training) is needed to classify the produced words into one of the two bags: positive and negative. This is called "opinion lexicon generation" stage. Then in the testing phase, tweets are colored positive or negative depending on the number and POS of the +ve/-ve words in the tweets identifies from the two bags.

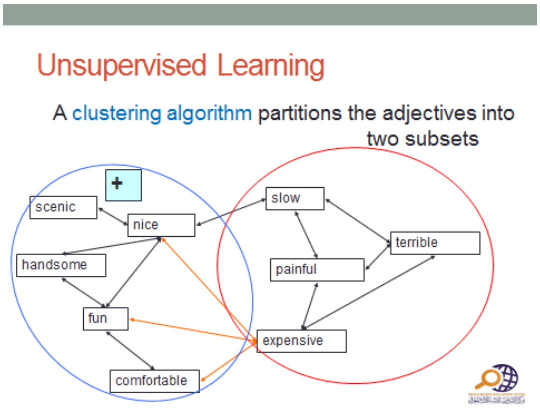

Figure 2 The surface analysis Tweet coloring algorithm's two bags.

3.3 Shallow Analysis

This level of analysis extends the previous level of analysis by replacing the "bag of words" with a "feature vector", where the words produced from the training tweets are inserted into a feature vector in the vector space. In the target domain we choose the best features that best characterize the domain.

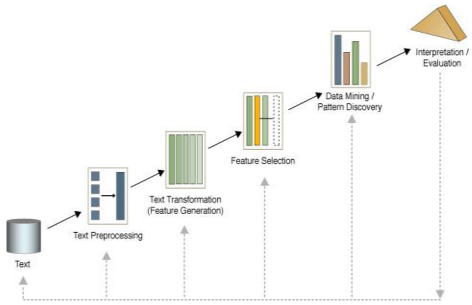

Figure 3 The shallow analysis Tweeter coloring algorithm.

These labels are thought of as column heads of a database table. This is a tweet level syntactic analysis where phrases along with words are also considered in the feature vectors. This is a continuous version of the surface analysis algorithm above. Figure 3 shows the shallow analysis Tweeter coloring algorithm. Text transformation is a preprocessing stage that prepares for the feature selection process. The prediction model is built in the pattern discovery stage. Then finally the testing mode is performed at the interpretation/evaluation stage.

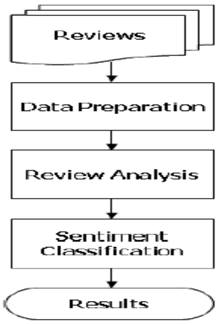

Figure 4: The Building Database in the "Review Analysis"

3.4 Deep Analysis

This level uses the attributes identified in the previous algorithm to create a corresponding database. Now the training set of tweets has been converted into a structured database. At this stage we can apply data mining algorithms such as discovering patterns, answer queries. Figure 4 shows the building of the database in the "review analysis" stage. At this level traditional data and visualization tools are used to analyze and visualize the results. Semantic analysis is considered at this level where ontologies such as "Wordnet" and "Q-Woednet" are used to understand hidden meanings.

Also background context knowledge about the target domain is consulted during the semantic analysis. This is another continuous version of the surface analysis algorithm above. Semantic analysis is considered at this algorithm where ontologies such as "Arabic Wordnet" and "Q-Woednet" are used to understand hidden meanings in the Tweets. Arabic Wordnet is classified into nouns, verbs, and adjectives. Once a POST (Parts of Speech) analysis of a Tweet is done, nouns, verbs and adjectives can be identified and used as input into their corresponding Arabic Wordnet ontology and consequently deep understanding of the Tweet becomes possible. Background and context can also be used as in our previous work [1] to help understand the Tweets.

Figure 5 shows the deep analysis Tweet coloring algorithm.

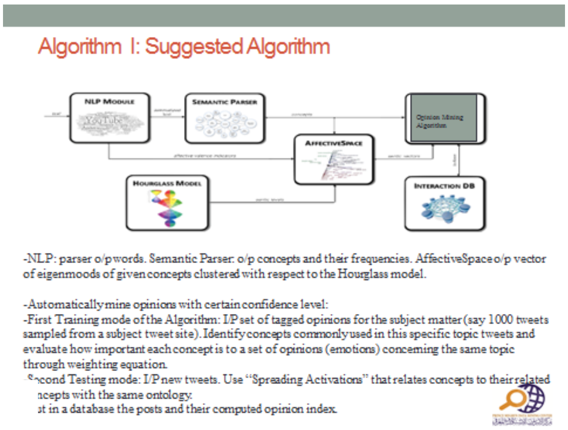

Figure 5: The Deep Analysis Tweet Coloring Algorithm

A multi-dimensional space built by blending Concept Net (a semantic net of common sense knowledge) and WordNet-Affect (a linguistic resource for the lexical representation of emotions)- See figure 6.

-This alignment build a matrix of: rows are concepts and columns are emotions, and whose values indicate truth values of assertions. +ve values for +ve emotions and –ve values for –ve emotions, 0 when nothing is known about the assertion. Dot product is high when two vector concepts are close in the AffectiveSpace above (eigenmoods form the axes of the above effective Space)

-Similar one for Arabic and target language tweets, threads, and text

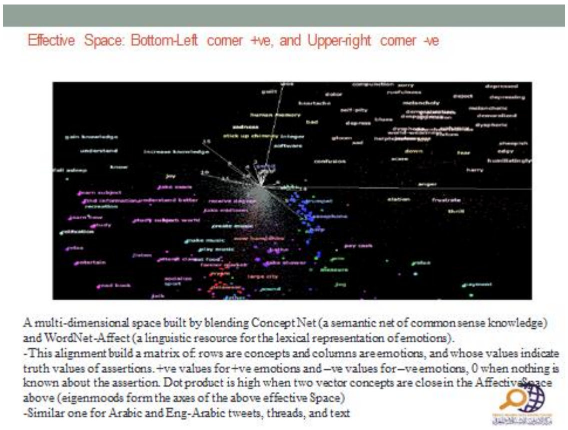

Figure 6: The Multi-dimensional Space of Concept Net and WordNet Affect

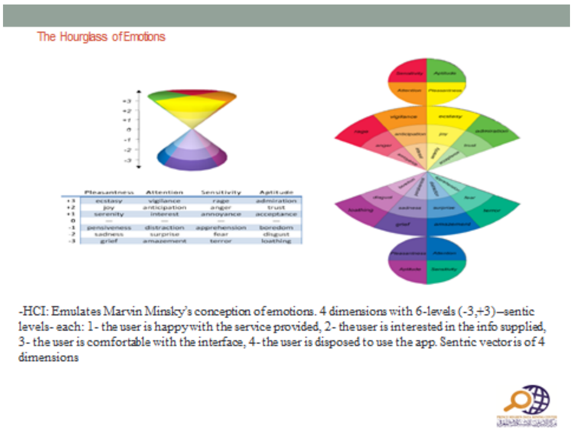

Figure 7: Marvin Minsky's Conception of Emotions

This diagram emulates Marvin Minsky's conception of emotions (See figure 7). 4 dimensions with 6-levels (-3,+3) –sentric levels- each: 1- the user is happy with the service provided, 2- the user is interested in the info supplied, 3- the user is comfortable with the interface, 4- the user is disposed to use the app. Sentric vector is of 4 dimensions.

3.5 Edge Coloring and Data Mining

Twitter followers and following create twitter graphs that can be manipulated using traditional graph theory algorithms to measure the spread (cascading) of opinions, influence and trust in a particular social network graph. We use the following "Edge Coloring" algorithms to discover the spread (cascading) of opinion, influence and trust. They are based on the traditional "Maximum Flow Minimum Cut" family of algorithms. The idea is that we would like to compute the flow of "Opinion", "Influence", and "Trust" from one follower to another by analyzing a sequence of Tweet exchange between them.

There is considerable literature on graph theory, network optimization, and the minimum cut set problem. The accomplishments of the work reported here are a) to find and implement a practical way of solving large networks for minimum cut sets and b) to discover that the minimum cut set for a large Twitter network was much smaller than what might have been expected given the number of edges in the network.

3.6 Opinion Spread

Figure 8 shows the details of the opinion spread algorithm.

After using any of the Tweet coloring algorithms above, an edge between two node i and j is either incremented or decremented according to the tweet color ij or ji. At the end of this stage an edge is colored according to the accumulated count over the edge. Large positive corresponds to heavy dark color, large negative corresponds to light blurred color. We use these counts as edge capacities in the well known "Maximum Flow Minimum Cut" problem.

Solving for a minimum cut set is a well-known network problem, and existing map data and network solvers could have been used to answer the question posed above. We have used a modified network solver algorithm. The first step in this algorithm is to determine where, in a general sense, the barrier is to be located. In our case, because we wanted a barrier surrounding the center of the graph, we defined the barrier's general location in terms of its distance from the center of the graph. After determining the general area in which the barrier will be located, the second step is to apply the methodology to find a minimum cut set in that general location.

To determine where a minimum cut set is to be located, we drew two concentric circles around a central point in the area in question. The source of "Opinion" is assumed to start at an unknown location outside the outer circle, with the intention of reaching an unknown location inside the inner circle. The actual path network considered consists of those edges that have at least one node (i.e., endpoint) between these two concentric circles. The second step in our algorithm is to find a minimum cut set for the graph whose segments have one or both nodes between these boundary lines. This is a well-known problem in graph theory, and it might be expected that there would be many solvers that could be used to obtain a solution to it. However, all but one of the solvers we considered could not find minimum cut sets in networks as large as the Twitter network of a real popular issue such as (مستوطنات). One, the GNET solver, could do so. Several solvers were also Excel-based, and so were restricted by the size limitations in capacity. Others could only accept the problem in the form of a general linear program and their LP-interfaces were unable to handle the problem. As a result, it took significant effort for us to identify just one solver that could find minimum cut sets in networks as large as the (مستوطنات) network. That one was the GNET solver [3]. GNET is designed to handle very large networks, and our experience so far is that it can handle networks of over one million arcs.

To let the max-flow algorithm find a minimal cut set, the following structure was used. Each of the segments with both nodes inside the ring was given a capacity of counts as described before. An artificial "super-source" node was added outside the outer ring, and an artificial "super-sink" node was added inside the inner ring. The outer node of each segment crossing the outer ring was changed to this super-source node and these segments were given an infinite capacity. Similarly, the inner node of each segment crossing the inner ring was changed to this super-sink node, and these segments were also given an infinite capacity. Finally, an artificial road segment going from the super-sink to the super-source was added, also with an infinite capacity.

We then solved this network for its maximum flow and, hence, for a minimum cut set. This minimum cut set gives the smallest number of users that must be there in order to ensure that any “opinion” attempting to penetrate the inner circle, starting on any node from outside of the outer circle, will necessarily encounter the super node. It should be noted that there could be more than one minimum cut set, and if there is more than one, the cut sets may or may not include some of the same nodes.

```
algorithm double_scaling;
begin
  set x := 0; v := 0;
  set Δ := 2^⌊log U⌋;
  set ε := 1/(n+1);
  while Δ ≥ 1 do
    augment_flow_double;
    set Δ := Δ/2;
  end while
  if (x, π) is not ε-optimal with ε < 1/n then
    reoptimize_flow_double;
  end if
  set P := find_admissible_path(G(x), ε, s, t);
  augment δ := (D − cx)/⌊π(s) − π(t)⌋ units of flow along P and along
  update x, cx, v;
end
```

Figure 8: Opinion Spread Algorithm

3.7 Influence Spread

Social Networks tools such as Klout [4] measure one’s influence on a network of followers based on his/her ability to drive action within a certain exchange. Kred [5] analyze tweets over the last 1000 days of an exchange. It analyzes what others do because of you. Influence increases when others take action because of one’s content. Peer Index [6] measure how active, receptive audience within an exchange. Tweepar [7] computes the ratio of number of followers to the number following.

```
Algorithm 1: Hybrid maximum flow algorithm
  (Apply preprocessing operations)
  Label vertices
  repeat
    Depth-restricted flow augmentation
    Update vertex labels after r augmentations
  until no augmenting path with prescribed length between s and t is found
  Switch to ‘double tree’ or ‘push/relabel’ strategy
  (Undo preprocessing operations (reroute flow))
```

Figure 9: Influence Spread Algorithm

Figure 9 shows the influence algorithm that computes a numeric score for a candidate. It measures the volume of waves (cascading of information flow) that his/her posts generate over the Social Network. The algorithm uses statistical indicators to measure "Influence" of a node in Twitter. We compute for each Candidate node its influence using the following indices: 1- No of Tweets he/she sends a day, 2- No of mentions he receives a day, 3- No of Re-tweets his followers make a day, 4- No of people he/she follows, 5- Eigen-Factor: Weight of "highest-count-of-followers" person who ever Re-Tweeted one of his Tweets, 6- Rate of Followers growth, 7- Cascade measure: Breadth and Depth of spread of his/her Tweets (through Re-Tweets), 8- Diffusion measure: Breadth and depth of spread of his/her name (through mentions), 9- Degree Centrality: no of neighbors in his/her graph of followers up to 3-levels deep, 10- Betweeness Centrality: Exist on the shortest paths of all other node-pairs of his/her 3-levels deep neighbors graph, 11- Closeness Centrality: Has the shortest paths to every other node in his/her 3-levels deep neighbors graph, 12- Engaged Audience measure: Summation of "Direct Messaging"+“Mentions"+"Re-Tweets", 13- How Active measure: Time since he/she joined Twitter & Time he/she spent on Twitter a week, 14- Eigenvector Centrality: Higher scoring nodes contributes more to his/her score, 15- Hub measure: Arabic- Hebrew-Persian hub, 16-Visibility measure: Best visibility of what is happening in the network, 17- Persuasion measure: His/Her convincing power to internalize the argument into his/her audience and make them adapt it (on exchange of Tweets), 18- Impression measure: His/Her process to influence audience by regulating and controlling the exchange of Tweets (reputation), 19- Gallop measure: His/Her efforts in marketing himself/herself and ideas, 20- Bit-ly URL interestingness: How much interesting his/her URL links

3.8 Trust Spread

By comparing the opinion spread graph and the influence spread graph we can produce a trust spread graph. Figure 10 shows the “Trust Spread” algorithm. This algorithm has two versions: Binary and Continuous. Its basic structure is: Source polls neighbors for trust value of sink, then the source computes the weighted average of these values to come up with an inferred trust rating; when polled, neighbors return either their direct rating for the sink, or they apply the algorithm themselves to compute a value and return it. Figure 10 show how to infer trust: The source node A and the sink node C- and recommending to the source how much to trust the sink.

```
CAPACITY—SCALING
x ← 0
Δ ← 2^⌊log₂ U⌋
while Δ > 0
        do while in G_x exists path s ⇝ t with the capacity of at least Δ
                do find an augmenting path P with the capacity of at least Δ
                    δ ← min{r_ij : (i, j) ∈ P}
                    augment δ units of flow along P
                    update G_x
        Δ ← Δ/2          ▷ Integer division
return x
```

3.9 Tweets Translation

The proposed Client calls Google translate web service to perform the Persian/Arabic and the Hebrew/Arabic translation service and vice versa. This idea of integrating social networks and sematic web services has been introduce in our previous research [2]. In that works we claim that future social networks will be a focused and specialized one; mixing both context and traditional web services to enhance social networks and provide richer environments. Performing accurate translation depends so much on disambiguation and context understanding. Context aware Social networks have been the subject of our paper [1].

```
procedure successive shortest path
  begin
    X = min{H(X)|X ∈ Z^E}
    π = 0
    while there is a node s with g_s(X) > 0
      compute the reduced costs c^π(X)
      determine the shortest path distances d(i) from s to all other nodes
        with respect to the reduced costs
      choose a node t with g_t(X) < 0
      let w(s, t) denote the shortest path vector from s to t in D
      X = X + w(s, t)
      π = π − d
    end
  end
```

Figure 10: Trust Spread Algorithm

4 Client Synthesizing Component: Tweets Translation

The proposed Client calls Google Translate web service to translate both ways from Arabic to the target language and from the target language to Arabic.

Figure 11 shows the system architecture of the proposed synthesizing component with metrics dashboard. The Synthesizing Component allows participating in the target culture/language through automatic translation of "Arabic" posts into the target language and automatic posting on the Social network of the target audience.

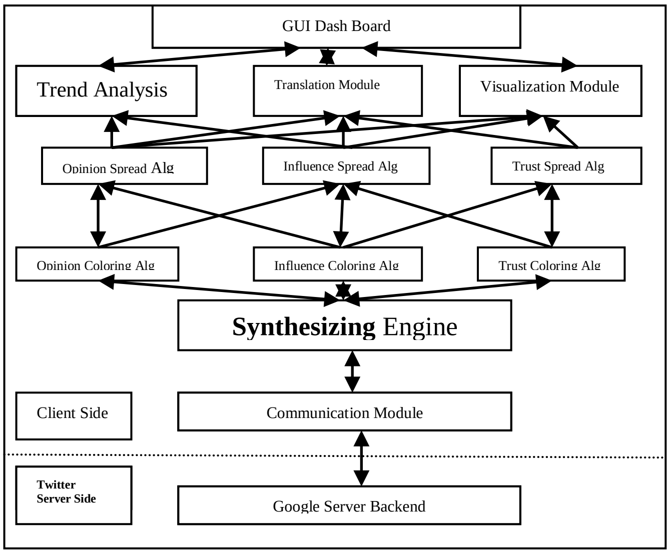

Figure 11: **Synthesizing Component** Architecture

The "**Synthesizing** Engine" communicates with the Communication Module to call the Google Translate Server Backend using Google APIs to send and retrieve translations. The retrieved translations are passed by the Engine to appropriate analysis module according to the End user needs. Built-in analysis algorithms (as described in the previous section) are called upon to process these translated tweets and display their results either through the Visualization Module or the Graph Analysis Algorithms. As shown in the figure, there is interaction among the analysis algorithms and they call each others for intermediate processing. Trend Analysis is a module that accumulates analysis results over a period of time and performs trend analysis for future prediction. The synthesize module can work concurrent with the analysis module.

5 Prototype Implementation

A prototype of the proposed Client with metrics dashboard is implemented on-top of "Nodexl" [8]; an open source template for Microsoft Excel that allows automated connection to a social network server and import (Using Twitter APIs) any data stream into the usual Excel environment. Tweet coloring and Edge coloring algorithms are implemented as Excel Macros with selective setting to either surface, shallow, or deep analysis parameters. Influence is measured through some twenty graph related terms such as: Degree centrality, Betweenness centrality, closeness centrality, Bit-ly interestingness, visibility measure, cascade measure, etc. Visualization graphs are provided that allow dynamic filtering, vertex grouping, adjusted appearance (zoom into areas of interest), graph metric calculations, etc. In order to provide real-time mining service even in ‘Deep’ analysis setup, a parallel cluster farm of duplicate servers is provided in order to help with parallel processing capabilities.

Each analytic algorithm provides sentiment analysis results with human classification of the sentiments expressed in tweets."By illustrating instances when unprompted, natural conversation deviates from responses to specific survey questions, the algorithm helps capture the nuances of public opinion.The Indices are updated five times daily. The synthesize module calls the Google translate service upon need. The synthesize module can work concurrent with the analysis module. Figure 12 shows two screenshots of the implemented prototype.

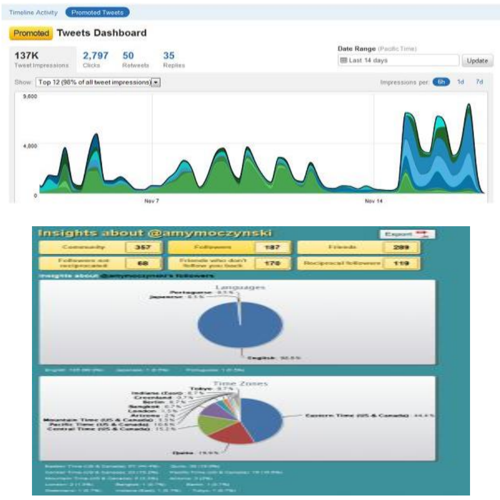

Figure 12: Two Screenshots of the Analytic Tool Dashboard

Code for the shallow Tweet coloring Macro, the surface Tweet coloring, Macro, the Deep Tweet coloring Macro, the Influence Spread Edge coloring Macro, the trust Spread Edge coloring Macro, and the Translation Macro can be found on our web portal at [10]

Visualization graphs are provided including ones that allow dynamic filtering, vertex grouping, adjusted appearance (zoom into areas of interest), graph metric calculations, etc. In order to provide real-time mining service even in ‘Deep’ analysis setup, a parallel cluster farm of duplicate servers is provided in order to help with parallel processing capabilities. Elaboration on both the visualization and the multi-server parallel processing can be found in [9]. Nodexl provide several graph visualization capabilities. Various styles of node, edge, flow information can be selected and various viewing criteria can be set. Figure 13 shows one of the possible “Clustered” visualization of nodes. Color codes can also be customized to help in the visualization of the graph.

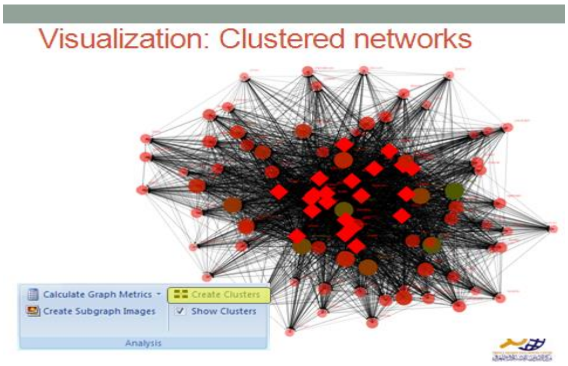

Figure 13: The Client Visualization Capability-Clustering Nodes

6 Experimental System

6.1 Experiment I- Hebrew

We started our first experiment by creating a new Twitter account with a Hebrew name. We then searched #مستوطنات using its Hebrew translation # התנחלויות. After getting 50 Tweets into our Client and translating them to Arabic. We choose a “shallow” analysis option. Four main threads of discussion were detected: The recent court decision, building on Plastinine lands, Europ boycuts products of settlements, and favoring settlements using religious coercion and fanatisim. We then used tweet and edge coloring macros to detect opinions, influence and trust among the four groups of discussions. We then replied to the most influencial trustworthy participant of each group. Then we got involved with the four groups in their discussions. The four groups seemed to accept our views and take them seriously. That was the objective of the experiment. See figure 14 for sample of these tweets, and figure 15 for the postings we have made in “Hebrew”.

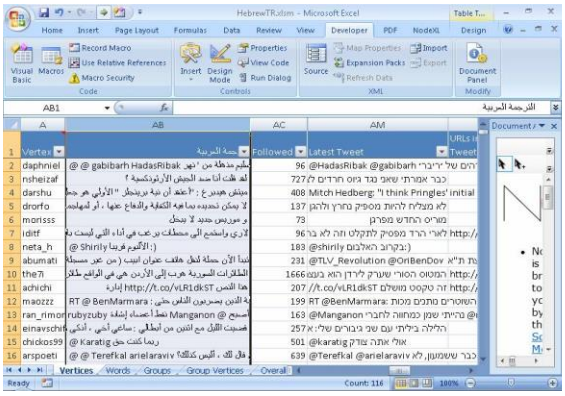

Figure 14: Hebrew Tweets Analysis

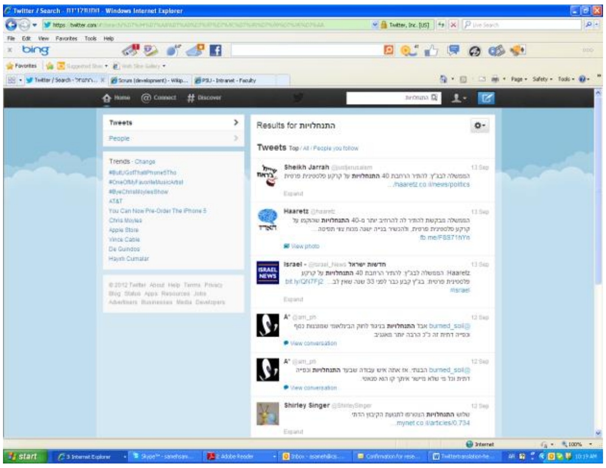

Figure 15: Hebrew Posting

6.2 Experiment II- Persian

We started our second experiment by creating a new Twitter account with a Persian name. We then searched #الدولارusing its Persian translation # دلار. After getting 30 Tweets into our Client and translating them to Arabic. We selected a "Deep" analysis choice. Two main threads of discussion were detected: The recent Saudi soccer match, and religious coercion and fanatisim. We then used tweet and edge coloring macros to detect opinions, influence and trust among the two groups of discussions. We then replied to the most influencial trustworthy participant of each group. Then we got involved with the two groups in their discussions. The two groups seemed to accept our views and take them seriously. That was the objective of the experiment.

Figures 16 and 17 show the "Persian" network and our participation through the proposed client in a "Persian" speaking Twitter network.

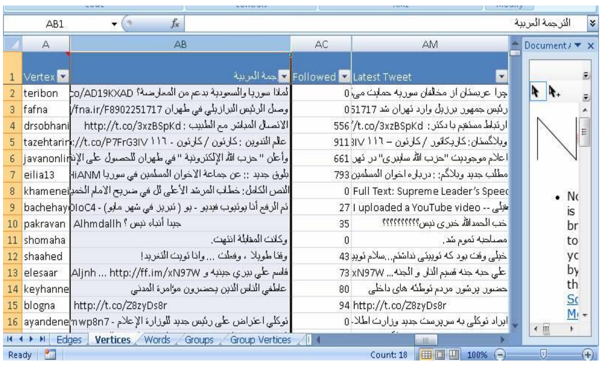

Figure 16: Persian Tweets Analysis

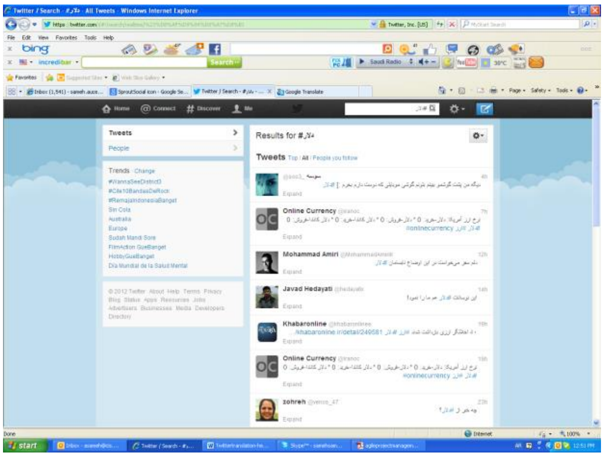

Figure 17: Persian Tweets Posting

7 Conclusion and Future Work

We have described an implementation of a new Twitter Client that has the capability of plugging on two Social cultures: Hebrew-speaking and Persian-speaking. We propose a Client that collects live tweet streams on frequent time intervals from #tags of specific topics of the target language and compute a mix of qualitative and quantitative indicators to measure Opinion spread. The proposed Client provides three levels of analysis: Surface, Shallow, and Deep depending on analyst wish and the traffic rate of input streams for the purpose of providing real-time mining service. Hebrew, Persian and Arabic tweets are analyzed taking into consideration important context background information. The proposed Client uses Google translate service to translate all Tweets to Arabic and then focus only on "Arabic" analysis (assuming users are Arabic speakers).

References

[1] Ahmed Sameh, “Future Social Networks: Integrating Context into Semantic Social Networks with Web Services”, Working Paper

[2] Ahmed Sameh, “A Twitter Analytic Tool to Measure Opinion, Influence and Trusrt”, ICDDM-2013

[3] Kodama Y., Kudoh T., Takano R., “GNET gigbit Ethernet Network Testbed”, Proceedings of the 2004 IEEE International Conference on Cluster Computing. 2004

[4] www.Klout.com/

[5] www.Kred.com/

[6] www.peerindex.com

[7] www.tweepar.com

[8] www.nodexl.codeplex.com

[9] Ahmed Sameh, “Parallel Processing with Twitter Analytics”, Working Paper

[10] Author Web Portal: http://www.psu.edu.sa/psu/cis/~sameh