What the Masses Want: A Case Study in Knowledge Discovery from Politically Oriented Data

Samhaa R. El-Beltagy, Moustafa Ghanem, Heba Ezzat, Sourya Ezzat, Mohmmed Aboelhouda, Ahmed Gamal, Mohamed Elkalioby, and Shady Alaa Issa
Nile University, Smart Village - B2 - Km 28, Cairo-Alex Desert Rd
Cairo 12677, Egypt

ABSTRACT
This paper describes an approach taken to analyze and categorize a sizable dataset of politically oriented posts that were submitted to a popular idea bank, Egypt 2.0, created following the Egyptian revolution. The aim of the analysis was to organize and present the data in a simple way that allows the voice of the people to be heard by decision makers and activists in a critical 6 week period in February and March 2011. The constraints faced when developing the approach included the absence of a classification scheme, the unavailability of training data, the need to assign more than one category, or label, to individual posts and the need to complete the task in a short period of time. The goal of this paper is twofold. Firstly, to present and evaluate the rapid development framework and algorithms used to organize the data. Secondly, to document the challenges encountered when both developing the system itself and analyzing the data, and to present our experience to the research community with the aim of identifying potentially new interesting research topics.

General Terms
Knowledge discovery, Taxonomy building, Classification

Keywords
Text Mining; Text Analysis; Topic Categorization; Multi-Labeling.

1. INTRODUCTION
The wide use of social media, online forums, and other online means of communication, is encouraging people to become more vocal with their political ideas, points of view, and aspirations. Any government that is serious about hearing the voices of its people must be attuned to their ideas and suggestions and must take these into consideration in its decision making and policy setting process.

The work detailed in this paper describes an approach that was taken to do precisely that after the Egyptian revolution. The approach was developed to address a real-life problem that emerged immediately following the revolution when a prominent political activist created a Google moderator series (Egypt 2.0) to allow Egyptians to express their dreams and ideas for what is to become the new Egypt (Google Moderator is a tool that allows distributed communities to submit ideas, events, presentations, etc, as well as vote on them). In just one week, Egyptians, at that point eager to participate in shaping their country, submitted over fifty thousand postings with more than one million votes. A single person would typically enter a single posting expressing all of his/her ideas in fields that s/he deemed important. So a single posting could contain ideas covering topics ranging from political reform, restructuring of the police force, improving education, to making the streets cleaner. While valuable information could be extracted from this resource such as areas that people feel need the most improvement, concrete ideas for making changes in a given area, etc, the size of this submitted data became a primary obstacle in its immediate utilization and made it necessary to carry out further analysis and processing in order to make it useful for decision makers, researchers, or an average user interested in learning more about what others have suggested.

The goal of the presented work is twofold. Firstly, to present the framework developed for the rapid development of a system capable of organizing this data by categorizing it and presenting it in a way that would allow the voice of the people to be heard even in the absence of a classification scheme. Secondly, to document the challenges encountered when developing the system and present them to the community with the aim of identifying new research topics.

The presented system has been deployed and one of its outputs was a report highlighting the main demands/ideas of the people using a category tree that was derived from the data itself. The report was included as part of the "national dialogue" congress initiative organized by the interim Egyptian government in late March 2011. The aim of the congress was to bring together activists and politicians to discuss the future of Egypt.

The rest of this paper is organized as follows: section 2, presents a brief problem description, section 3 provides an overview of the proposed system, section 4 presents the results of evaluating the proposed system, section 5 describes an enhancement that was brought about as a result of analyzing evaluation outcomes, section 6, overviews related work, section 7, briefly describes some of the insights gained by classifying user posting, section 8, summarizes the lessons learned and identifies future research challenges, and finally section 9, concludes this paper.

2. PROBLEM DESCRIPTION
The input to the proposed system is a large set of postings, written in both Arabic and English, collected from the idea bank - each of which can span multiple topics. The majority of the postings were relatively short with people trying to express their opinions in as little words as possible. The main problem that this work had to address is that there was a need to classify or annotate these posts in the absence of a classification scheme. The other problem is that classification had to take place without any training data. The first problem requires an understanding of the existing data, and the creation of a classification taxonomy to capture it. Manual inspection of the data and the consequent manual development of a classification taxonomy is not only a time consuming activity, but also an error prone one as many categories or subcategories may be missed in the presence of a large amount of data. To address both problems the following steps were taken:

1. Devise a semi-supervised data driven approach to create a classification taxonomy
2. Assign to each post all topic labels addressed by that post.
3. Create a web based system to facilitate reaching, browsing, searching and understanding this information

These steps were taken through the development of a system that is described in the next section.

3. SYSTEM OVERVIEW

The developed system consists of three main components: the first takes in as input postings and outputs a classification taxonomy. The operation of this component is semi-supervised. The second component takes in as input the classification taxonomy and any number of postings that need classification and assigns labels to those. The third component is the user interface which takes in as input labeled posts and allows the user the facility to browse, search or obtain various statistics on these posts. Each of these components is described in more details in the following sub-sections.

3.1 Semi-Supervised Taxonomy Building

As stated before, in order to build a classification taxonomy, a semi-supervised approached was followed. The main idea behind the followed approach was to group related posts together and then extract keyphrases from each cluster so as to capture the main concepts covered by each of these clusters. A keyphrase in this context is defined as a term consisting of one or more words that can act as a descriptor for the cluster from which it was extracted. By looking at the collective set of phrases, extracted from some given cluster, a human annotator can easily understand the main topic being discussed by this cluster.

So, the first step in building the taxonomy had to be the application of a clustering algorithm to group related postings together. Because of the fact that many postings address a multitude of topics, clustering accuracy was not expected nor was it a priority. So, we have used a simple centroid-based clustering algorithm to carry out this task. Rather than use all of the postings, a random sample of those comprising 10% of all the data was used (approximately five thousand posts). For each cluster, key phrases were extracted using the KP-miner system which is described in details in [5]. Each group of keyphrases was then used by a human annotator to determine a category and eventually build the entire taxonomy. For example, a group of keyphrases that has the phrases "education strategy, illiteracy, curriculum, teachers, curriculum development, exams" will most likely signal the taxonomy builder to create a category called **Education**.

Rather than discard the keyphrases associated with each identified category, it's also the duty of the annotator to refine these by removing any noise and adding any other keyphrases that s/he feels is relevant. After the completion of this process, the keyphrases can be thought of as important concepts related to each of the identified categories and subcategories and keeping them serves an important role in the later task of categorization. The terms "keyphrases" and "key concepts" will hence forth be used interchangeably throughout the paper.

The following algorithm summarizes the process followed to build the taxonomy.

1. *Cluster input posts.*
2. *For each cluster c_i obtained in step 1*
   a. *Extract keyphrases*
   b. *Present keyphrases to user so that s/he can:*
      i. *Create a category or subcategory based on extracted keyphrase*
      ii. *Edit keyphrases by adding, deleting or modifying what has been extracted.*
   c. *Associate keyphrases with created category or subcategory*
3. *Save Classification tree(the taxonomy) along with its associated keyphrases*

Table 1 presents a list of the obtained top level categories while Figure 1 depicts a sample of one category in the taxonomy (Politics) and its subcategories. The depth level for each category varies from zero to three children.

Table 1. A list of obtained top level categories

•**Politics**	•Housing and Construction	•Justice	•Religion and Ethics
•Education	•Anti-Corruption laws & policies	•Food and Agriculture	•Communication and IT
•Economy	•Social	•Health	•Aviation
•Media	•Scientific Research	•Industry	•Infra structure
•Work Force	•Transportation	•Human Rights	•Culture and Arts
•Energy	•Environment	•Defense	•Tourism

3.2 Categorization of Postings

In order to assign each posting with labels or topics that best describe its content, we propose an approach that uses language and discourse information; we explore the fact that some words and phrases are highly correlated with specific topics. Since we have no training data, we've decided to capitalize on the keyphrases associated with each category and subcategory to derive a relation between the postings and these categories. The main idea of the categorization algorithm is to extract key concepts from each posting and calculate its proximity to entries in the taxonomy. To model taxonomical categories as well as posts, the vector space model is used. To achieve the final goal of assigning multiple labels to postings, the following steps are carried out:

1. Preprocessing of taxonomy entries.
2. Preprocessing of postings
3. Building weight vectors for categories and subcategories.
4. Building weight vectors for postings.
5. Calculating the similarity between each posting and all category vectors.

6. Assigning labels to a posting based on computing similarity values.

Each of these steps is explained briefly in the following subsections..

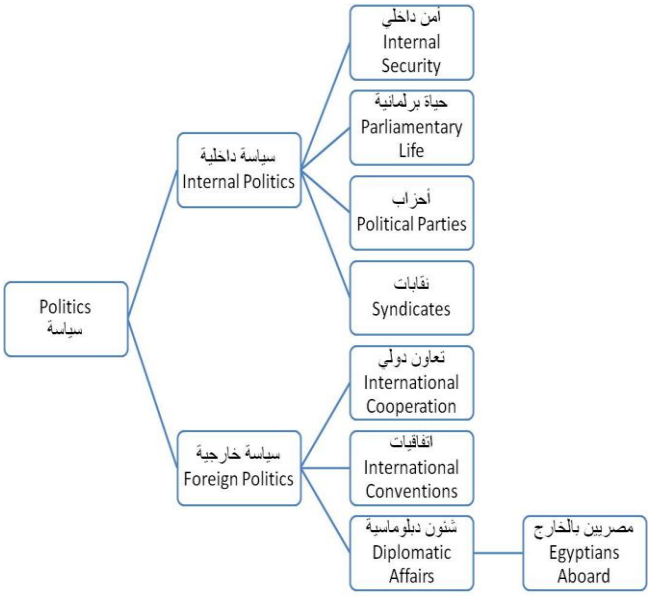

Fig 1: An expanded view of category "Politics"

3.2.1 Preprocessing of taxonomy entries

As stated before, each taxonomy entry (which is either a category of sub-category) is associated with a set of keyphrases. Preprocessing of taxonomy entries simply entails stemming both the category name and its associated keyphrases. Since in our particular case, all entries were in Arabic, the stemmer described in [4] was the one used. All keyphrases are then stored in a table along with the frequency of their occurrence in various categories (the taxonomical keyphrase table).

3.2.2 Preprocessing of postings

Preprocessing of postings involves a number of steps which can be summarized as follows:

1. **Language unification**: even though most of the postings were entered in Arabic, there were cases when an entry was made in English or where a user would enter his/her ideas used transliterated text. So the first preprocessing step was to identify such postings and to convert them to Arabic. The Google translation and transliteration APIs were used to translate English posts to Arabic and to convert transliterated posts to their Arabic respectively.
2. **Duplicate post removal:** It was observed that some of the users made the same posting several times, so the next preprocessing step involved the detection and removal of duplicate postings.
3. **Elongation Detection**: Since the language used in making the postings was informal, some of the users tried to emphasize certain terms in their post by elongating them. An example of elongation in English would be to write "yesssssssssss" instead of just "yes". An "Elongation filter" was thus developed to detect and remove elongated words.
4. **Stemming:** the stemmer described in [4] was the one used
5. **Key concept detection**: A key concept in this context is defined as any phrase that is associated with at least one category entry in the taxonomical tree, i.e. it appears in its list of keyphrases or as its name. To extract these, an efficient algorithm provided as part of the KP-Miner API[5] and that computes all possible n-grams for a given sentence is used. The n-grams are then matched against entries in the taxonomical keyphrase table and only n-grams that are found, are kept. At this point each posting is simply represented by its id and key concepts.
6. **Build an IDF table for key concepts:** IDF (inverse document frequency) is a numerical statistic which reflects how common a given term is in a collection and hence its importance or ability to discriminate [13]. An IDF table, in the context of this work, is a table containing key concepts each of which may be one word or longer, along with their IDF values. An IDF value for a key concept c is computed using the following equation:

$$idf(c,P) = \frac{|P|}{|\{p \in P : c \in p\}|} \quad (1)$$

Where P denotes all postings, | P | is the total number of postings and |{p∈P:c∈p}| is the total number of postings where the key concept c appears.

3.2.3 Building weight vectors for categories and subcategories

The next step after preprocessing both taxonomical entries and postings, is to represent each using a weighted feature vector. A feature vector of a category includes the stemmed name of the category as well as the stemmed representation of all key phrases associated with this category and sub-categories. Initially, each keyphrase is assigned its idf value which is obtained from the IDF table of key concepts. However, this value is further modified based on the following observations:

- Keyphrases consisting of more than one word are more likely to discriminate and should thus be given a higher weight than single word key terms. This is done by introducing a boosting factor β1.
- The name of a category is probably the most indicative feature in the feature vector of terms and should thus carry more weight. This is done by introducing a boosting factor β2.
- Any term that appears as a keyphrase in multiple categories should be penalized. A factor Θ has been introduced to penalize any such terms.

To summarize, given a category feature vector C_i consisting of key phrases k_1 to k_n, the calculation of the weight of any k_i is given using the following formula:

$$\textbf{weight}(k_i) = idf(k_i, P) * \max\{1, (\text{multipleWord}(k_i) * \beta 1)\} * \max\{1, (\text{catName}(k_i) * \beta 2)\} * 1/\text{catCount}(k_i) \quad (2)$$

Where multipleWord(k_i), and (catName(k_i) are boolean functions that test whether k_i consists of multiple words and whether k_i is a category name respectively, and where

catCount(k_i) is another function that returns the total number of categories in which k_i has appeared as a keyphrase.

3.2.4 Building weight vectors for postings

For a posting p_i, represented by a list of key concepts or keyphrase c_1 to c_n, the weight of c_i is simply calculated using the following formula:

$$\textbf{weight}(c_i) = \textbf{idf}(c_i, P) * \textbf{tf}(c_i) \quad (3)$$

where tf(c_i) is the number of times term c_i has appeared in posting p_i. Since the postings are usually quite short, this value is usually simply equal to 1.

3.2.5 Calculating similarity between a posting and all category vectors

To calculate the similarity between various categories and postings, the cosine similarity metric is used. So given any posting p_j, and a category c_k the similarity between both is calculated using the following formula:

$$Sim(p_j, C_k) = \frac{\sum_{i=1}^{N} w_{i,\,pj} \times w_{i,\,ck}}{\sqrt{\sum_{i=1}^{N} w_{i,pj}^2}\sqrt{\sum_{i=1}^{N} w_{i,ck}^2}} \quad (4)$$

where $w_{i,\,s}$ is the weight of keyphrase w_i in source ***s*** and N is length of the used feature vector.

In this step, given a posting p_j, the similarity between this posting and all categories C_1 to C_n is calculated and stored in a list *matchingCategories* that is used in the next step.

3.2.6 Assigning labels to a posting

In this final step, categories or labels are assigned to a posting. To assign a category name as a label to a posting, first the score of the category with the maximum similarity value is obtained. If this value is smaller than some threshold Ω, then the posting is returned un-labled. Otherwise, this score is used to normalize all other scores so that their values would range from zero to one. Only categories with a normalized score that is greater than some given threshold γ are used to label the posting. The rationale behind this normalization process, is to prevent categories whose scores have a wide gap with the similarity score of the best matching category, to be used as labels. This has been found to positively influence precision, while slightly diminishing recall. The process is summarized by the following algorithm.

1. *max = getMaxScore(*matchingCategories)
2. *if (max < Ω) return*
3. *for each category C_i in matchingCategories*
4. *score(C_i)= score(C_i)/max*
5. *for each category C_i in matchingCategories*
6. *if(score(C_i)= > γ) then*
7. *assign (pj, category(C_i))*

3.3 The Back-end System and User Interface

To implement the system based on our algorithms we developed an integrated system that starts by downloading the complete set of posts automatically from Google Moderator. Using Python scripts to coordinate the complete workflow the content of posts is then pre-processed and categorized using our algorithms to derive the category labels automatically. The posts, automatically assigned labels as well as the original meta-data of the posts (author, date, votes for and votes against) are stored in a MySQL database hosted on an Apache server. A web-based front end based on Java script, JQuery and ProtoViz charting tool, was also developed for browsing the posts and analyzing their statistics. The complete system allows the user to:

1. Browse the ideas using the developed taxonomy
2. Discover which subjects the Egyptians regard as most important by presenting a statistical breakdown of the ideas with respect to the developed taxonomy.
3. Search within entered ideas using a search interface.
4. Understand relations between ideas through a relation co-matrix.
5. Easily view ideas that reference public figures and key entities (a static dictionary was used for this purpose).

The complete system is available online at: http://www.nubios.nileu.edu.eg/7ewarat/v1/

Figure 2**Error! Reference source not found.** shows various snapshots of the user interface, which is presented entirely in Arabic. These include the interface for browsing the automatically-annotated posts, the heat-map used for representing the category co-relation matrix, the interface for displayingcategory statistics as well as time-based analysis of the categories. In the heat-map (bottom right screen), category lables appear on the rows as well as on the columns of the matrix, but might be too small to see. Different colors in this matrix, denote different co-relation degrees with burgendy for example, denoting a high corelation.

4. EVALUATION

To determine whether the categorization performance of the system is acceptable or not an experiment using 500 randomly selected posts, was conducted. In this experiment, the chosen posts were manually annotated using category labels by a person who was not involved in the system development process. A comparison between the system's assigned labels and the list of manually assigned ones was then carried out in order to calculate the average precision, recall and F-measure metrics (MicroF1). The results of the experiment are shown in Table 2.

Table 2: Evaluation of the devloped categorization algorithm over 500 postings

Precision	86.2 %
Recall	83.53%
F-score	84.84 %

When analyzing these results, it was observed that two main factors affected the performance of the system:

1. Manual comparison between system annotated posts and the same manually annotated ones showed that some of the labels assigned by the human annotator are derived from an implicit reading of the post which

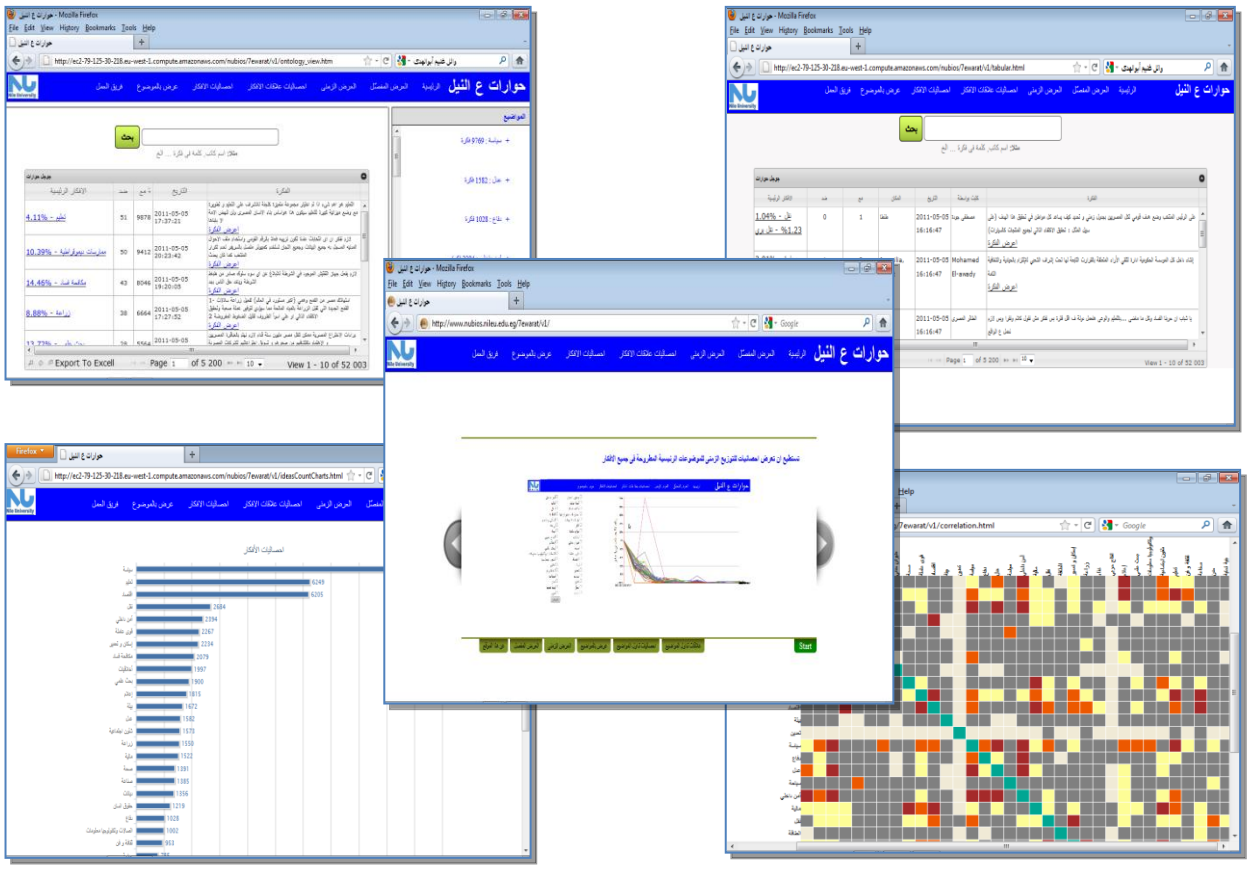

Fig 2: Various snap shots from the user interface

can be very hard if not impossible to duplicate automatically

2. A keyword that is strongly associated with a certain category may have other meanings in a different context thus leading to the assignment of an incorrect label, which directly affects precision.

The fact of the matter is that the performance of the system will always be related to the quality of keyphrases associated with each category. This has led us to introduce a refinement that is described in the next section.

In order to validate the proposed approach even further, we've compared the results of categorizing the Alj-Mgz[1] dataset using the developed algorithm, with a system that categorizes the same dataset using a supervised approach. Specifically, we compare our results with the best results obtained during a comparative study that aims to maximize classification accuracy of Arabic text [12]. The classifier used by [12] is a Support Vector Machine (SVM). The Alj-Mgz data set consists of 4,462 articles, categories of which are already known. The documents in this dataset are distributed unevenly among eight categories. We've manually mapped the names of these categories to ones in our classification scheme, However some of the categories in Alj-Mgz dataset such as locals, internationals and society, did not map to any of the categories in our taxonomy. As a result, these categories were not taken into consideration when carrying out the evaluation. So, the experiment was conducted using only 2,050 articles. Precision and recall were evaluated on top-level categories only because Alj-Mgz has a flat hierarchy. For example, a post that is labeled by our system as "Internal Politics", would automatically be mapped to its root parent which is "Politics". When applying our algorithm on the we got an average precision value of 83.26, recall of 85.53 and an f-Score of 85.8. The result reported by [12] on the same dataset, is given only in terms of the f-score the value of which is 86%. It is important to note here that is result is given in terms of all 4,462 of which we've only used a subset. Nevertheless, having achieved a result that is so close to the one reported, even on a smaller subset, is an indication of the validity of the followed approach.

5. SYSTEM ENHANCMENTS BASED ON EVALUTION RESULTS

Manual selection of keyphrases that best abstract a given category may sometime results in giving too much strength to a term that can be used in different contexts and in different categories. To adjust this possible undesirable side effect, a simple algorithm was introduced to refine or replace manually selected keyphrases to associate with categories. In order for this algorithm to work, we need to have a reasonable amount of postings each of which belongs belong to a single distinct category. Rather than resort to manual classification for a subset of postings, we've opted to use the already devised algorithm to obtain these. To do so, categorization using the previously described algorithm was performed only on very short postings where a short posting is defined to have ***n*** or less words. In our experiment we set ***n*** to 25. The rationale

[1] Available online at http://filebox.vt.edu/users/dsaid/AlgMgz.tar.gz

behind using only very short postings, is that these are almost always going to map to just one category. Filtering all postings to obtain only very short ones, resulted in retaining 10386 postings. After all posting are categorized, postings with similar categories are concatenated. Then, for each set of concatenated postings belonging to category C_i keyphrases are extracted. These keyphrases will overlap with some of the originally obtained keyphrases, but might not include all of them and will almost always include new ones. After that, the chi square statistic [ref] is calculated for each keyphrase. Keyphrases whose score exceed a certain threshold are said to be the most discriminative words representing category C_i.

Table 3: Comparison between results using the original keyphrases and using the new keyphrases in 2 settings

	Precision	Recall	F-Score
Original keyphrases	86.2	83.53	84.84
Original keyphrases + new keyphrases	87.45	85.65	86.54
New keyphrase only	94	87.85	90.85

The experiment presented in the evaluation section was then replicated under two different settings. In the first setting, newly obtained keyphrases were used to augment existing keyphrases, and in the second, they were used to replace existing keyphrases. The results are shown in Table 3.

As can be seen from the results, the best outcome is achieved when the entire list of keyphrases associated with each category is replaced with the one derived used the presented approach.

6. RELATED WORK

The work presented in this paper is related to a number of research areas, the most obvious of which is text categorization. Classical categorization methods are based on machine learning and probabilistic approaches. A good review of current categorization methods is presented in [14]. Yet, almost all automatic categorization techniques rely on training data in the form of pre-classified documents to help in the classifiers' learning process and hence, require a labeled dataset [10].

As training data is not always readily available, such as in the case of the work presented in this paper, some have suggested the use of an ontology in order to classify unstructured documents into meaningful categories without the need for training [8] [9]. In [8] and [9] the authors present a novel ontology-based approach to apply automatic text categorization on an English CNN news dataset. The paper derived its ontology from Wikipedia[17] and WordNet [7]. In many situations a semantic hierarchy may exist between categories, documents belonging to a "lower-level" category must also belong to another "higher-level" category. For example, documents belonging to a tennis category must also belong to the sports category. Developing such a hierarchy is discussed in [2][16].

Multi-label categorization is another area addressed by this work. Since multi-labeling is a more challenging task than single class categorization, most of the work conducted on a multi-label collection with ***m*** classes, runs ***m*** independent binary experiments, thus simulating a local labeling method, which is artificial and unrealistic. In [6], the authors use MLTC-multi-label topic classification- which is usually accomplished by generating ***m*** independent binary classifiers, one for each class and each entrusted with deciding whether a document belongs to class or not. More sophisticated work was presented by Naonori and Kazumi [15] where they proposed a novel probabilistic generative model, called parametric mixture models (PMMs) for multiclass, multi-labeled text categorization problem and derived efficient learning and prediction algorithms for PMMs. There is also a more general probabilistic model for multi-latent-topics text called Latent Dirichlet Allocation (LDA), which was proposed in [1]. LDA is formulated in an unsupervised manner.

7. INSIGHTS GAINED BY CLASSIFYING USER POSTINGS

After all postings have been classified, some analysis was carried out to understand what people are mostly interested in. Fig 2illustrates the number of posts related to or talking about a specific topic.

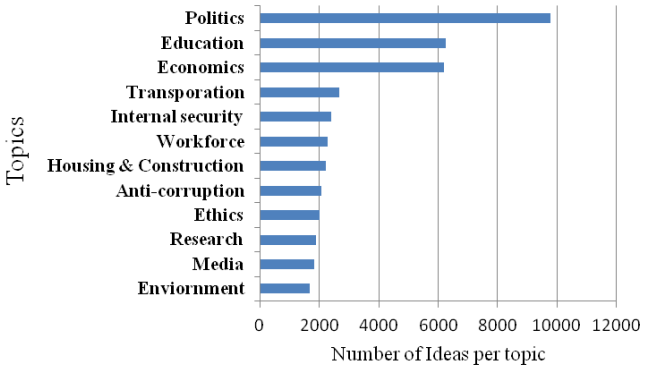

Fig 2: Breakdown of ideas using top level categories (only the top twelve categories are shown)

As an observation, the figure shows that the main topic of people's concern is Politics. Indeed, following the 25th of January revolution, the main topic of conversation everywhere has become politics. The next main topics of concern are those of Education and Economics. As a growing country, with an emerging and developing economy, people want to see a visible improvement in economy and in the education of their children. Transportation also seems to rank highly amongst peoples' concerns. Given the widespread occurrence of traffic jams; on average, citizens waste as many as 3 hours daily just to go to work and back, which can explain why this topic ranks so highly. Internal Security and the reform of the police force are another two major topics of interests.

An in depth analysis of these results is outside the scope of this paper. A report highlighting the main demands/ideas of the people under each category derived from the data itself was submitted as report distributed as part of the "national dialogue" congress initiative organized by the Egyptian government in March 2011. In the report the key demands were grouped into a deeper taxonomical tree.

8. LESSONS LEARNED AND OPEN RESEARCH CHALLENGES

Further mining ideas such as those addressed by this work to co-relate users (by classifying them) with their ideas can go a long way in understanding the needs of separate segments in the society. The motivation of the original Egypt 2.0 site on Google Moderator was to create an idea bank, where people

submit proposals for a new Egypt and to create a forum allowing them to express their opinions about such ideas, discuss them and vote on them. Labeling the posts into categories (Politics, Economics, Education….) as we have done in this work is a first important step towards organizing the data, which is extremely rich and was never intended for automatic processing. The data is very diverse in nature: it includes a large number of posts that either simply mention a key problem that is well worth prioritizing or requires immediate attention (e.g. need for spending more money on education or restoring trust between people and the police), or go as far as detailing how to implement certain proposed solutions. Moreover, many of the posts make references to cultural or social contexts. In fact, the data contains posts from more than 40,000 people with different writing styles, different cultural, social, ideological and educational backgrounds that can be inferred by a human reader, but clearly not by an automatic system. Many ideas and views were clearly effected by the background of the writer, examples of which are related to issues pertaining to taxes, personal liberties, income discrepancies and free trade. A human reader viewing such posts can tell immediately whether the author of the post has a liberal, leftist, or religious background. Understanding such context is clearly needed when attempting to explain or analyze why people have conflicting views on any one topic.

Clearly, attempting to conduct any form of deep analysis on this data set or other ones derived from similar ideas banks leads to various interesting challenges. Three of the key challenges that we encountered and that need to be considered before contemplating the automatic analysis of such rich data include:

- The development of ontologies for describing what the user's post represents: an "idea", a "problem" description, a high-level "solution", a "concrete solution", a "view" on a previous post, etc. This task is a complex one, especially when posts cover compound concepts.
- For analyzing ideas with the goal of understanding complex ideological references, there is a need to develop more detailed and specialized ontologies and knowledge bases that would help categorizing the data based on a more subtle basis such as political or religious orientation.
- Detecting how ideas influence other ideas. "Influence" is a direct outcome of the social network/discussion forum nature of an idea bank. Not only do people comment on posts by other users and vote on them, but their own ideas are influenced by them. Although in some cases a certain thread could be identified, automatically tracing the provenance of idea/solution progression is very difficult. If what is needed is to encourage people to propose new ideas, or to trace their progression, then the issue needs to be addressed in a structured way.

Text Mining Challenges:

The posts available in the data set themselves were a mix of Arabic language, English Language and Transliterated Arabic (Arabic written in English characters) and covering a wide range of topics. The key challenges include:

- The development of methods that do not require any human intervention, or at least minimize it, for learning a taxonomy of topics on the fly from large amounts of data, especially when the posts cover a wide range of topics, as well as to support developing deeper taxonomies than the one developed in this work.
- The development of tools for mining colloquial Arabic text as this can simplify the task of mining this dataset and any colloquial Arabic text in general. Colloquial Arabic text maps to spoken Arabic dialects, and there is no standard way for representing many of the used terms. This results in many variations in spelling the same term and clearly complicates any matching effort. Secondly, the use of informal structure and grammar means that most available part of speech taggers or morphological analyzers are not likely to produce any accurate results for such text. Finally, punctuation marks are rarely used when writing colloquial Arabic, making it very difficult to determine where the borders of a sentence lay. Many of these problems are not necessarily specific to Arabic but are, more generally, a feature of the informal nature of postings on social media forums and informal idea banks in many languages.

9. CONCLUSION AND FUTURE WORK

The work presented in this paper described our efforts for developing a system for rapidly categorizing short text segments representing popular demands and ideas for a new Egypt. Despite the simplicity of the presented algorithm, it resulted in the rapid development of a complete usable system and submission of a report, summarizing the most important and re-current ideas in individual categories to politicians in Egypt two months after the revolution started. By conducting the work itself we have identified and documented a number of key interesting challenges that we believe may deserve future research efforts by the KDD community at large.

Our future work includes experimenting with more methods for selecting keyphrases associated with categories and experimenting with other algorithms for matching between categories and labels. We will also develop an automatic summarization feature for generating reports that highlight distinct ideas for each available topic.

Since it is expected that more and more idea banks as well as complaint banks will start emerging, we expect that the developed system will be utilized across a number of government agencies.

10. ACKNOWLEDGMENTS

We would like to thank Wael Ghonim for providing us with the used dataset. This work was partially funded by Microsoft's Advanced Technology Lab in Cairo, grant number CIS-001-1011

11. REFERENCES

[1] Blei, D.M. Ng, A.Y. and Jordan, M.I. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, pp. 993-1022.

[2] Boyapati, V. 2000. *Towards a comprehensive topic hierarchy for news*. Master Thesis. The Australian National University.

[3] Chang, Y. and Huang, H. 2008. An automatic document classifier system based on nave Bayes classifier and ontology. *In Proceedings of 7th International Conference on Machine Learning and Cybernetics*, Kunming, China.

[4] El-Beltagy, S. R. and Rafea, A. 2011. An accuracy enhanced light stemmer for Arabic text. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(1).

[5] El-Beltagy, S. R. and Rafea, A. 2009. KP-Miner: A keyphrase extraction system for English and Arabic documents. *Information Systems*, 34(1), 132–144.

[6] Esuli, A. and Sebastiani, F. 2009. Active learning strategies for multi-label text classification," *In Proceedings of the 31st European Conference on Information Retrieval* (ECIR'09). Toulouse, France, pp. 102–113

[7] Fellbaum, C. (Ed) (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

[8] Janik, M. and Kochut, K. 2008. Training-less ontology-based text categorization. In *Proceedings of Workshop on Exploiting Semantic Annotations in Information Retrieval* (ESAIR 2008) at the 30th European Conference on Information Retrieval (ECIR'08), Glasgow, Scotland,.

[9] Janik, M. and Kochut, K.. 2007. *Wikipedia in action: ontological knowledge in text categorization*. Technical. Report No. UGA-CS-TR-07-001. University of Georgia.

[10] Joher, A. , Al-hajar, Z. and Kassem, F. 2008. Automatic Arabic text categorization with Bayesian learning. Damascus University - Department of Artificial Intelligence, 2008.

[11] Mendenhall, W. Beaver, R.J., and. Beaver, B. M. (2003). *Introduction to Probability and Statistics*. Brooks/Cole, a division of Thomson Learning.

[12] Said, D., Wanas, N., Darwish, N., and Hegazy, N.2009. A study of text preprocessing tools for Arabic text categorization. In *Proceedings of the 2nd International conference on Arabic Language Resources and Tools*. Cairo, Egypt, 2009

[13] Salton, G. and M. J. McGill (1983). Introduction to modern information retrieval. McGraw-Hill. ISBN 0070544840.

[14] Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys (CSUR*), pp.1–47.

[15] Ueda, N. and Saito, K. 2003. Parametric mixture models for multi-labeled text. *Advances in neural information processing system*s, 15, 721–728.

[16] Wang, B.B., McKay, R.I., Abbass, H.A., and Barlow, M. 2002. Learning text classifier using the domain concept hierarchy. In *Proceedings of the IEEE International Conference on Communication*s, New York, USA.

[17] Wikipedia. (2012). http://www.wikipedia.org/