Extracting Topics of Debate between Users on Web Discussion Boards

1T. Georgiou, 1M. Karvounis, 1Y. Ioannidis

1Department of Informatics and Telecommunications, University of Athens
{t.georgiou, manosk, yannis}@di.uoa.gr

Abstract
An increasing research interest is observed on how thread extraction from discussions, in online web discussion boards, can be accomplished using linguistic methods, text similarity algorithms and graph-propagation techniques. However, there are very few publications on this field that utilize the forum structure and the available knowledge. In this paper we describe a novel method for the extraction of all the topics a user has discussed about. Our work focuses on certain structure features that can be found on any modern web forum, such as the quotation system and the tree-like form of the conversations, in order to extract all the threads of a discussion. Temporal information such as the posting timestamps is also used in order to discover implicit connections between postings (posts) in a discussion (thread). This work is combined with an algorithm that discovers agreement and disagreement relationships between users, and results in the creation of groups with possible “friends” or “debaters”.

Keywords: Discussion boards, web forums, thread segmentation, topic extraction, social networks, experimentation

1. Introduction

Also known as internet forums, message boards and bulletin boards, web discussion boards are online discussion sites, users can join anonymously and participate in the ongoing conversations. People registering in an internet forum can create social bonds with other users, while discussing topics of their interest.

As internet population grows, more and more discussion boards are showing up in the cyber-space. Numerous web sites and companies install such boards on their pages in order to offer to their online visitors a way to discuss topics regarding the site, the company, their products, their services etc. Moreover, online discussion forums are very commonly used to offer technical support or to distribute the knowledge of users on certain issues (e.g. www.ubuntuforums.org offers a place where everyone can get or offer help regarding the Ubuntu Linux operating system). Finally, forums are also used as entertainment places, where users can discuss about various topics of their interest, as for example music, movies, politics, religion, current events etc. As a result, these online places accumulate an overwhelming quantity of knowledge; the
extraction of which can lead in various interesting conclusions and observations concerning the opinion and behavior of people on the internet.

Popular web forums have millions of registered users from whom usually at least a thousand is active (they post on a daily basis). Every day thousands of postings and new discussion threads show up, something that makes it difficult for someone to keep track of every new post, even about his topics of interest and only. It is very common for active users to spend many hours a day in order to read every new post and write their opinion for the corresponding topics. Moreover, new users, who face a new forum with millions of posts and unknown users, and have no previous knowledge of the ongoing topics discussed, find it difficult to fit in and adapt to the forum’s environment. Commonly, new users ask questions already answered, which results to old users getting annoyed and new users not having replies to their inquires.

Although there is a lot of work regarding the extraction of user opinion, until now the extracted knowledge is targeted and used only to estimate people’s preferences on products, movies, political views etc. What we want to succeed through our work is to use this knowledge in a user-oriented way: the implementation of a web service, as part of the forum engine, that would offer to its users “follow suggestions” for users that discuss relevant topics. Similar to the friend suggestion system that Facebook has, every social media site could use a service that suggests to its members, unknown users that have possibly common interests and opinions. Something like that could also improve the user experience in Twitter or other (micro-)blogging systems in general.

In this paper, we focus on the efficient extraction of the topics discussed by a user in a forum thread. The problem consists of two parts:

- The thread detection part. In order to estimate the specific topic the user discusses we need to cluster all the posts that refer to this topic and apply a topic extraction algorithm on this set of posts. Getting the topic from the title only or the first post or from the whole thread, has shown that it may lead in erroneous results, especially in large sized threads where topic is frequently changing with the lapse of time.

- The topic extraction part. After collecting all the posts referring to a specific discussion topic, a topic extraction algorithm must be applied. It is important to take into consideration the special characteristics of the text a post contains. Topic extraction algorithms that process whole text documents can again give wrong results when processing postings with small size and casual writing style.
We propose two methods in order to efficiently accomplish each of the above challenges. The novelty of our algorithm for the thread extraction lies in the utilization of the special structure-features offered from a web forum (figure 1), while most related methods use only the linguistic information contained in postings.

2. Related Work

Related work can be found in both fields of thread extraction and topic extraction, on web discussion boards.

A technique that utilizes the special features offered from the structure of web forums in order to cluster the posts of a discussion that have the same topic, is proposed by [Kim et. Al. (2005)]. They propose an algorithm that uses temporal information as the time and date of posts, the post authors etc. and they create posting chains using a topic similarity algorithm enhanced with the utilization of the quoting system. Topic keywords are computed in a weighted basis depending on the distance of the quoted postings from the analyzed post. Next, they segment these chains using segment boundaries that are identified through topic specialization and generalization. This process is repeated on a top-down fashion in order to segment whole message hierarchies. An interesting part of this research is the computation of the effect that the quotations have on their algorithm. The same algorithm was tested without taking into account the post connections that quotes offer. The experiment showed that there was an increase in the erroneous results by 11%. This led us to further consider the importance of the quotation system.

[Adams et. Al. (2008)] utilizes the structure of online chat in order to solve the thread extraction problem in an existing conversation. They use the time distance between chats and the nicknames to discover relationships between them that could possible correspond in topic similarity.
Utilization of the specific structure of a web forum can also be found in work that has nothing to do with the thread extraction or topic extraction problems: [Huang et. Al. (2007)] propose a method to extract chatbot knowledge from forum threads. The idea is to create pairs of thread titles and posts that contain relevant information. In order to classify a post, whether it is relevant to the topic of the thread or not, they use the quotation system, the post’s author knowledge and the time distance between posts.

[Wang et. Al. (2008)] studied cases where the implicit thread structure of an online conversation is missing and published an algorithm that recovers this knowledge using message similarity methods. [Wu et. Al. (2007)] proposed an approach to discover topics in online discussions that takes into primary consideration the user participation frequency. Finally, an interesting method to track the topic change in a discussion is proposed from [Labadié et. Al. (2008)]: They use linguistic methods (lexical similarity and thematic distance) to detect the topic boundaries in a conversation and split it in topic related clusters.

Regarding topic extraction, the usage of TF-IDF (term frequency) and text similarity methods is very common: [Adams et. Al. (2008)], [Harada et. Al. (2003), [Sehn et. Al. (2006)] and [Shi et. Al. (2009)]. In our work we also use a variant of the same method to discover the topics discussed in a forum thread. Finally, [Nagao (2007)] indicates how to achieve summary extraction from online discussion content. However, her work focuses on Audio/Video and not on text media.

3. Our Approach

In our approach we use only temporal information and forum features in order to create clusters of posts that are related to the same topic. Our algorithm computes, in a greedy way, for every post of a thread, its neighboring posts and then extracts the topic from this cluster. We observed that when a post quotes or replies to another, then it refers at the same topic with a very high proximity. As so, we define as neighboring posts those that are connected with quotes or replies (and not just by having small distance between them!).

3.0 Analysis Observations on Forums

Discussions in a web forum are similar to those of a chat box. Common features are:

- Many users (more than two) may participate in a conversation.
- Users may leave and come randomly in time.
- Unstructured and casual language is used.
- Posts appear in a linear way based on the time and date they were posted.

However, there are also some characteristics that make those two types of online discussion critically different; at least regarding the issues we face in our work:
- Posts usually contain a (relatively) large number of words in contrast with chats that contain one or two sentences at most; they even sometimes contain only a couple of words.
- The time between successive posts can vary between minutes and hours or even days. This has as a result for a discussion to last many days or even months.
- In order to help conversation participants understand to whom a user may reply to, web forums offer the quotation system, a way to include parts of the quoted post in the reply (figure 1).
- The topics discussed in a thread can be more than one, especially in large threads.
- However, every thread comes with a title that gives the general thematic direction of the topic.

Given the above observations, it is clear that the first problem we had to encounter is the existence of several topics in a thread, which is something very usual in popular discussions. Moreover, even if the topic discussed in a thread is unique, different features and aspects of it may be analyzed in the conversation. This led us to create an algorithm that isolates parts of the discussion and then extracts the topic using only these parts and not the whole thread.

### 3.1 Pre-processing

In our project we implemented a program (web crawler) to collect entire discussions from a web discussion board (in our tests we used the forum “4forums Political Polls and Debates”). These discussions are stored as individual posts along with important information as the author and the date they were posted. Posts are then cleaned from unnecessary html tags and finally are linked with other posts that they may quote. All this data is stored in a database for quick access and easy manipulation.

### 3.2 Extraction of the Posting Graph

In the first step we need to recover the posting graph structure. When posts are added in a discussion they form a linear structure where every post follows another, sorted by publication date. In this step we leave behind this structure and recover connections between posts, both explicit and implicit. Quotations are considered explicit connections: when a post quotes another, then there is a clear relationship between them. Implicit connections are considered between posts where a user replies to another by posting exactly below, in a short period of time. It is indeed very common for users not to use the quotation system when they reply immediately to the post they want, before anyone else posts something new. Implicit connection extraction can be further extended by also discovering posts where a user replies to another and uses his name in his post rather than quoting him (e.g. a user replies to the user “junkie_poster” like this: “@junkie_poster I think you should not post in
every thread!”). However this introduces us to further problems: nickname mismatches or the usage of only a word of the nickname for simplicity (e.g. only “junkie” rather than “junkie_poster”). We do not take into account these connections in our current work.

Gathering all these post connections results in the creation of a graph (figure 2) that has for nodes the posts of the discussion and as vertices the connections between them (either quotes or direct replies). We also added a weight to each vertex that represents the distance of the two posts. This distance is a function of two very important metrics: the number of posts interpolating between the two posts (post distance) and the time passed between the publication of the two posts (time distance). The weight is proportional of these two values.

![Figure 2. Posting graph and post/time distance examples (each node in the graph represents a post in the thread)](image)

timestamp(i): May 15, 19:37
post distance(i, i+3): 3

timestamp(i+3): May 15, 23:41
time distance(i, i+3): 4h 5m

Regarding the post distance distribution (figure 3), we discovered that most posts replying to another, are usually posted either immediately after the latter, either after two posts or three. Besides, it is very common in web forum societies when new users join a conversation, not to even read the previous pages and therefore completely ignore the flow of the discussion till then. This is also proved from the second chart, the time distance distribution (figure 4), were it is clear that when a user replies to a post, he will do it immediately or at least in a period of four or five hours. When more time passes, it is unlike for the user to reply at all at the specific post.

![Figure 3. Post Distance Distribution](image)
3.3 Topic Term Extraction

For every post we extract its topic terms, if this is possible, depending on the number of words it contains. In this step we used the online topic term extraction service offered by “AlchemyAPI”. We also tested the “Yahoo! Term Extraction” service but found its algorithm less suitable for overall small documents. With “AlchemyAPI” we got a percentage of 89% of posts that offer at least one term.

In order not to account quoted posts multiple times (a “popular” post could be quoted by many users) and therefore bias the result of the term extraction, we exclude all internal quotations (quoting posts of the same thread) and only keep external ones (e.g. when a user quotes an article from a blog or a news-site).

3.4 Post Neighbors

In this step we utilize the posting graph in order to discover the close neighbors of every post. For every post \(i\) from the thread, we compute a coherent sub-graph \(G_i\) of \(G\) (where \(G\) is the whole posting graph) with the following concept. Starting from the post \(i\) and using a BFS technique, we collect all posts whose distance from post \(i\) is no greater than a given value \(w\). This value can vary depending on the nature of the forum. As stated above, the distance is computed considering both the post-distance and the time-distance values, as they were described in section 3.2. Distances are computed accumulatively in order to get the distances of posts that are not connected in the graph in an immediate way. For example in figure 2, the post distance between
posts $i$ and $i-2$ is computed as the post distance of $i$ and $i-1$ plus the distance of $i-1$ and $i-3$ plus the distance of $i-3$ and $i-2$. Same goes for time distance.

Running the above algorithm, we get a set of posts for each post $i$ of the discussion. We consider this set as the thematic context of the specific post, and as so, topic extraction is applied on this set only.

### 3.5 Authors and Topic Clusters

In the final step of the algorithm, we convert the “post-oriented” extracted information to “author-oriented”! We process every author that participates in the discussion and cluster the topics he mentions in his posts (as this has been found using the method described in 3.4). The number of these topics cannot exceed the number of posts of the user (may be equal at the most). However, posts of power users that post very frequently will be neighbouring in many cases and as so, will result in having the same actual topic and are clustered together.

In order to improve the clustering of topics discussed, and discover related topics between different authors, we apply hypernym augmentation using “WordNet”. Therefore, even if we have authors that appear to discuss different topics, after applying the augmentation we may receive a common topic basis and cluster them together!

### 4. Using the Extracted Knowledge and Conclusions

This research is conducted in the framework of a project with two parts. In the other part of the project [Karvounis et. Al. (2010)], we have developed a method that inserts users in two sets, depending on their opinion homogeneity. More specifically, if two users are in the same set, they are assumed to have homogenous opinions, whereas two users in different sets are assumed to have heterogeneous opinions. These results can be combined with the method described in this paper to identify agreement-disagreement between pairs of users: if two users are in the same set (meaning that they have homogenous opinions) and in the same cluster (they talk about the same topic) they are assumed to agree, whereas if they are in different sets (heterogeneous opinions) and in the same cluster they are assumed to disagree.

### 5. Evaluation and Conclusions

First experiments show that the proposed algorithm can give promising results. We evaluated 1000 posts on more than 10 threads and we roughly estimated that the success ratio reaches the percent of 76%.

What has to be underlined here is the advantage of our method against linguistic approaches: regarding the part of the thread extraction, the algorithm is independent
of the language used in the web forum, as only temporal information is utilized! This is quite important given the international nature of the internet.

5. Future Work

Primarily we need to work on the evaluation of the method. Given the subjective nature of the problem (some times it is not clear even for a human to recognize the topic correctly), several diverse evaluators are needed, in order to have a realistic and sound evaluation.

Moreover, we plan to continue the research on this field. Certain parts of the algorithm can be improved. Weights can be added in the topic term clustering according to the position of posts in the thread, as [Kim et. Al. (2005)] did in their work, or even considering the quality of posts from their author (e.g. if his/she is a power user). On the creation of the posting graph, the further recognition of implicit post connections (as mentioned in 3.2) can be added. The topic term extraction procedure can be improved by adding recognition of domain words in cases of posts where no any terms have been extracted using the mentioned tools.

Finally, the algorithm can be implemented and adopted by forum engines in order to offer the described suggestion system to its users. It would also be very interesting to extend this research on known and largely used social media, beside discussion boards, like Twitter, Google Buzz and Facebook.

6. Acknowledgments

We would like to thank our colleagues at the University of Athens that evaluated some results and helped us make an estimation of the success of the proposed algorithm.

References


Harada, S., Khandelwal, S. (2003), Automatic Topic Extraction and Classification of Usenet Threads, Natural Language Processing Course Project, Stanford University, Stanford, California, USA.

Karvounis, M., Georgiou, T., Ioannidis, Y. (2010), Utilizing the Quoting System to Estimate User Agreement on Web Forums, in SIGMOD Undergraduate Research Poster Competition, SIGMOD/PODS Conference 2010, Indianapolis, Indiana, USA.


Shi, L., Sun, B., Kong, L., Zhang, Y. (2009), Web Forum Sentiment Analysis based on Topics, in Proc. CIT09: 9th International Conference on Computer and Information Technology, Xiamen, China.

Wang, Y. C., Joshi, M., Cohen, W. W., Rosé, C. P. (2008), Recovering Implicit Thread Structure in Newsgroup Style Conversations, in Proc. ICWSM II: 2nd International Conference on Weblogs and Social Media, Seattle, USA.

Wu, Z.L., Li, C.H. (2007), Topic Detection in Online Discussion using Non-Negative Matrix Factorization, in Proc. WI-IATW07: International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops, Silicon Valley, California, USA.