JMLR: Workshop and Conference Proceedings 11 (2010) 134–141 Workshop on Applications of Pattern Analysis

Visualization of Online Discussion Forums

Mitja Trampuš MITJA.TRAMPUS@IJS.SI
Marko Grobelnik MARKO.GROBELNIK@IJS.SI
Jozef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia

Editors: Tom Diethe, Nello Cristianini, John Shawe-Taylor

Abstract

This paper describes a set of visualization tools which aid the understanding of discussion topics and trends in online discussion forums. The tools integrate into the forum's web page, allowing for easy exploration of its contents. Three visualizations are presented: a visual browsing suggestions mechanism, a semantic "atlas" providing a thematic overview of larger forum segments, and a timeline displaying temporal evolution of forum topics. The underlying algorithms have very few language-dependent components. The software is operational and can be tested live on Slovene, Slovak and Hungarian pilot sites, containing up to 5 million forum posts.

Keywords: Visualizations, Text Mining, Politology, Dimensionality reduction

1. Introduction

eParticipation, eGovernance and most eAnything have become hot topics in the field of political science in recent years. This should not come as a surprise: with more and more people using the internet, it is an obvious way of reaching out to the voters as well as collecting their feedback.

One of the richest internet sources of information on public opinion are discussion forums. In giving everyone an opportunity to speak out, they come close to the ancient ideal of democracy[1]. However, despite the constantly growing number of users and posts, the structure of these forums is not very different from the Usenet service which has been around since mid-eighties. On one hand, this is natural as a single fine-grained topic is easiest to follow with the good old threaded display, regardless of the number of such topics. On the other hand, navigating *among* the topics is becoming increasingly difficult. New visitors to the forum are met with an overwhelming number of topics and posts. With threads getting excessively long, it can even become hard to grasp the general idea(s) expressed in a single thread in a reasonable amount of time. On the aforementioned political discussion forums, analysts and decision makers are eager to follow the public opinion but simply cannot read through hundreds of posts every day.

As a possible alleviation of these scalability problems, we have developed a tool which enables any visitor of a discussion forum to easy visualize its contents and thus gain an overview of its structure and discussion trends. We have deployed it on three politically-

1. With some obvious restrictions: not all people use the internet, and not all users are equally motivated to post their opinion, a notable extreme being party propagandists.

themed forums, although the software is not tuned to this domain and could be used on any forum.

2. Software Overview

As the paper describes a working system, let us begin by giving a brief overview of the software from the end user's perspective. The underlying algorithms are discussed in later sections.

Our tool, the *VIDI toolbar*, is a generic add-on to any online discussion forum. It provides forum visitors and moderators with three types of visualizations of (subsets of) forum content. It integrates itself directly into the forum's webpage and is therefore easy to access, yet it takes up very little valuable screen space.

When a user first visits a VIDI-supported forum, the toolbar is discreetly hidden in the left margin of the page (Figure 1(a)) unless the user has used the toolbar on her previous visit. If the user clicks on the blue handle, the toolbar expands (Figure 1(b)).

The upper half of the toolbar is dedicated to selecting forum topics and/or threads[2] to be visualized. As the forum is potentially huge and its tree-like structure consequently unwieldy, we decided to reuse the navigation offered by the page itself to let the user guide our tool to threads of interest. To this end, the software injects a small selection icon in the page next to each thread or topic title (see Figure 1(c)). Clicking the icon toggles between the corresponding thread being added to or removed from the selection list on the toolbar. Using cookies, the selection list is preserved through subsequent visits to other forum's pages. Beneath the selection list, the selected topics can be further subselected by specifying a date range.

The lower half of the toolbar provides access to visualizations themselves; three of them are offered to the user. The first visualization, *browsing suggestions*, is the simplest one. It helps users lost in the deluge of threads as well as data analysts searching for more threads pertaining to their subject of analysis. Given a set of preselected posts, the visualization marks all related thread titles with orange blobs (see Figure 1(b)), the size of the blob being proportional to the similarity of the corresponding thread with the user-selected threads.

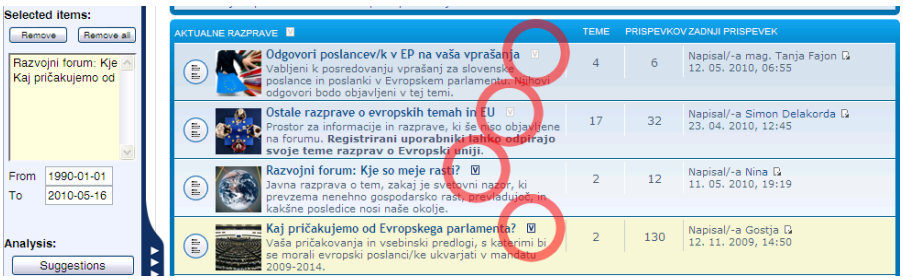
The second visualization, *topical atlas*, is intended to give users insight into forum's topical structure. Given a selection of threads, all the posts contained in those threads are displayed on a two-dimensional chart. Each post is represented by a point and the points are arranged in such a way that the distance between two points is roughly proportional to the similarity of the two corresponding posts. An example is given in Figure 2(a). Topically related posts naturally form visual clusters. To discover the topic associated with each cluster, the user can inspect the green keywords on the "atlas" or move the mouse around to get the keywords for the current location. Clicking on a point in the chart navigates the browser to the corresponding post.

The third visualization, *topical timeline*, helps understand how the "hot topics" evolved through time. Based on the posts from selected threads, a timeline from Figure 2(b) is

2. A forum is typically structured into broad *topics*, each consisting of *threads* which in turn consist of *posts*. Our software however handles topics and threads almost equivalently, so in this paper we use the word *thread* to mean "either thread or topic". Note that we do continue to use the word *topic* to denote latent semantic topics/themes.

(*a*) Initial, hidden state.
(Note the left margin.)

(*b*) Expanded and ready for use.
The "browsing suggestions"
visualization is visible.

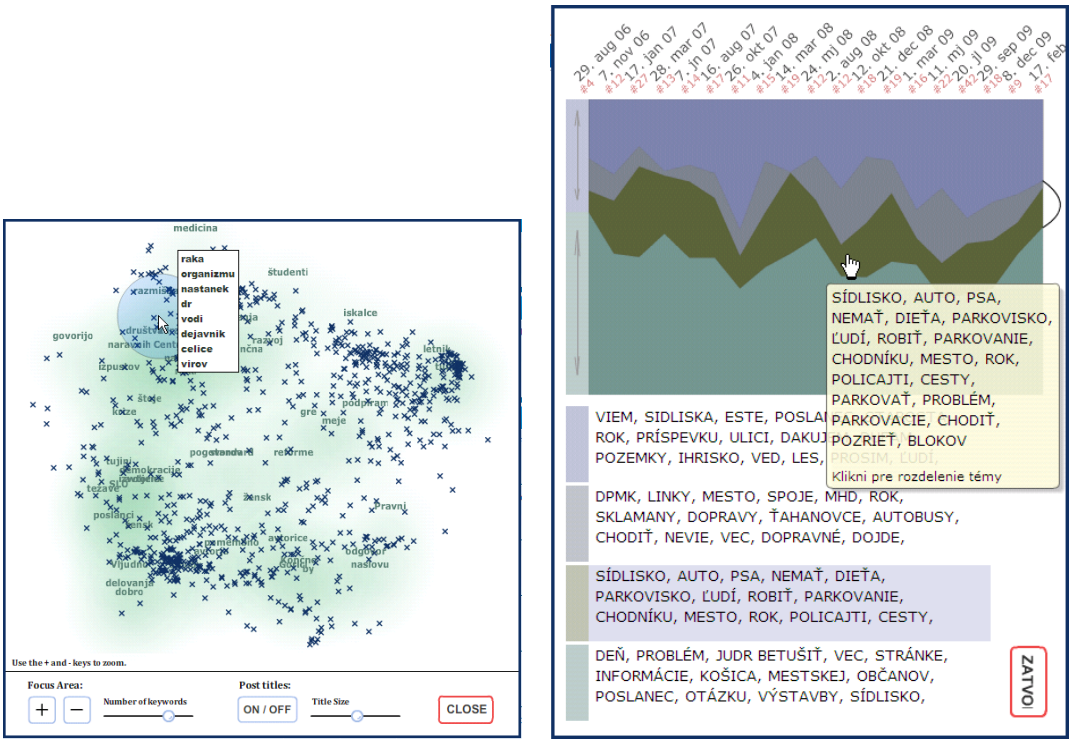


(*c*) The thread selection icons, here additionally marked with circles, embedded in the page by VIDI. The two threads with darkened icons appear in the selection list on the left.

Figure 1: A forum with integrated VIDI toolbar.

displayed. The horizontal axis represents time. Each colored strip represents a latent topic automatically identified by clustering; its keywords are given below the timeline and in the mouse tooltip. Thickness of each strip corresponds to the proportion of forum posts that relate to the strip's topic. Each strip can be split by the user into two, allowing to drill down on topics of interest.

The user interface is described in more detail in Trampuš et al. (2010), a deliverable of the VIDI Project from which the software described in this paper resulted. The project has



(*a*) The topical atlas. (*b*) The topical timeline.

Figure 2: Topical and temporal structure of the forum.

another data mining component (Sen et al., 2009); in this paper, we limit ourselves to the visualization components.

2.1. Adoption Effort

Effort has been made to make deployment of VIDI on a forum as simple as possible. For the system to start processing the data, a programmer only needs to provide a mapping from forum's data scheme to VIDI's and a regular expression for identifying thread titles on the web page. To finish the installation, the forum administrator includes a single line of javascript in the HTML code, making the toolbar immediately available and visible to all visitors. Alternatively, if a forum-wide installation is undesirable or impossible, each user can install VIDI individually simply by creating a bookmarklet in her browser.

3. Data Processing and Algorithms

3.1. Software Architecture

The software has a server and a client component. The server periodically fetches, preprocesses and caches all relevant data from the forums in a local database. When a visualization is requested on the client side, it performs all the heavy computation and provides

aggregated, display-ready data to the client. Its frontend is written in python and the computationally intensive parts in C++.

The client, i.e. the toolbar visible to the users, is written with the Google Web Toolkit (GWT); the visualizations are done in Flash. Almost no computation is performed on the client side. An interesting complication arises from the fact that the client is injected inside the forum page. This makes it technically a part of forum's domain, which complicates the communication with the server from both GWT and Flash – cross-domain scripting from web pages is intentionally made hard in the interest of security. Fortunately, workarounds such as JSON/P exist (and were used).

3.2. Data Acquisition and Preprocessing

All methods use solely data present in the local cache which is kept fresh by synchronizing with the forums approximately once an hour. This is easiest to do by directly accessing each forum's database via SQL and copying data changes into the local cache, adjusting for the data model differences between the two databases as we go along. If SQL access is unavailable (as it was with one of our use cases), web crawling and parsing is required. Note that this step unavoidably has to be performed manually for each forum; this means that VIDI cannot support any forum before some effort has been invested into data migration.

Some established preprocessing steps are performed on all forum posts once they are stored in the database: HTML/bbCode cleanup, tokenization, lemmatization and stopword removal. Additionally, we perform named entity extraction and consolidation. Finally, a sparse term frequency (TF) vector is created and stored for each post. To speed up processing of analytic modules, we also store TF vectors for all forum threads (and update them upon post insertions). We also cache some other basic statistics, e.g. document frequencies for all terms and average post length.

For the tokenization stage, we keep track of frequent n-grams in the corpus and, if encountered, mark them as a single token.

Lemmatization is performed with an adapted existing lemmatizer (Juršič et al., 2007); Slovene and Hungarian stemming rules were provided by the authors, Slovak ones were trained on the Slovak National Corpus (available at `http://korpus.juls.savba.sk`). We also track the most frequent surface form for each lemma to display to the user. Because all three languages are highly inflectional, lemmatization does worse than with english – accuracy is about 80%.

Stopwords are removed using stopword lists obtained from the three national linguistics institutions. Due to the informal vocabulary, heuristics have to be employed in addition to the lists to remove noise like "haha:)))" or "woooooo".

Named entity extraction is performed by a simple heuristic: each capitalized word that does not start a sentence is considered a name; a capitalized word at the beginning of the sentence is considered a name only if it already exists in the database of names. If capitalized words immediately follow each other, the whole sequence is considered a single name. This heuristic is quite error-prone, but produces mostly false positives which are in our case harmless as described below. In the disambiguation step, we use incremental min-linkage clustering with Levenshtein distance and a hand-tuned cutoff threshold. This means that whenever an unknown name n is encountered, it is checked against the most

similar (according to Levenshtein) name n^* in the database. If the difference is smaller than some threshold value T, n is assigned the same entity ID as n^*. Otherwise, a new entity is created with n as its only known name. If, in this process, the difference between two names from two distinct clusters falls below the threshold T, the clusters are merged; this was, the process is independent of the order in which the names are encountered. We do not consider different entities with equal names.

The extraction step obviously has low precision and high recall; however, the extracted capitalized words that are not really entities tend to have low frequency in the corpus, so hardly any consolidation occurs among them, leaving performance unaffected. For frequent entites (more than 10 occurences) however, the algorithm helps significantly, with the dominant cluster for each entity having about 90% precision and 80% recall.

3.3. Browsing Suggestions

Similarity between the query and target threads is calculated using the trusted cosine distance between averages of the corresponding TF-IDF vectors. The similarity is converted into blob size with a non-linear formula such that the differences between best-matching threads become more distinguishable.

3.4. Topical Atlas

To keep the amount of data manageable both for the user and the computation methods, we first randomly sample 3000 posts from the specified threads.[3] Computation and visualization is performed only on this sample.

We reduce the dimensionality of the space defined by the posts' TF vectors down to two dimensions required by the visualization in several steps. First, we prune the TF vectors to 2000 most prominent dimensions; then we project them onto 200 dimensions using LSI (latent semantic indexing) and from there onto two dimensions using MDS (multidimensional scaling). To speed up the process, an approximate result of MDS is first determined by clustering the posts into 200 clusters and performing MDS on their centroids.

The descriptive keywords for every point of the map are computed by taking the most prominent keywords (according to TF-IDF) from documents in the immediate neighbourhood of the point.

3.5. Topical Timeline

As with the topical atlas, a subsample of 3000 posts is first created.

To identify latent topics, posts are clustered using divisive hierarchical bisecting k-means with random restarts and TF-IDF cosine distance. Posts that are outliers with regard to their timestamp are discarded. The timespan described by the remaining posts is split into 10–20 equisized intervals (depending on number of posts and period length). The number of posts from a cluster in each of these intervals determines the thickness of the strip in the graph along the horizontal axis.

3. In fact, quite pseudorandomly – the sampling as well as all random decisions later on in the process must not change between two runs of the algorithm with equal input parameters, or else the users get very confused.

4. Testing and User Experience

The VIDI toolbar has so far been deployed in three languages on three forums and is publically available for testing; see `http://vidi.ijs.si` for the URLs. The use cases are quite diverse: a Slovene forum on EU-related issues, highly moderated, with about 1000 posts; a Slovak forum of a local municipality concentrating on local issues, lightly moderated, with about 10000 posts; and a Hungarian forum, dealing with everything political, unmoderated and very informal in tone and vocabulary, with about 5 million posts. The average post length descreases strongly with growing informality of the forum: 889, 585 and 283 words per post respectively for the listed forums.

A formal evaluation of users' satisfaction is currently being performed. Preliminary feedback, however, is positive. Moderators/analysts of all three forums have found the tool to be useful in assessing visitor opinion. In the Slovene use case, a report on an environmental issue was prepared for and as commissioned by the members of the European parliament; VIDI was also used in preparation of the report.

5. Related Work and Contributions

The visualizations proposed in this work have been used before. However, to the best of our knowledge, the data on which they were applied was different in nature, causing some differences in data processing as well as in interpretation of the images.

The general approach used for the topical atlas is first described in Fortuna et al. (2005). The topical timeline is traditional in structure; however, the topics to be shown on the chart typically need to be defined by hand. The approach of displaying latent topics identified with text clustering is first suggested in Shaparenko et al. (2005) and implemented e.g. in the IST-World project (Ferlež, 2006).

While the core data mining methods in our work are not greatly different, we believe this is the first time they have been applied to and tightly integrated with discussion forums to visualize public opinion. Forums as a data source also require changes in parts of the data processing pipeline: the informal language sparsifies and increases the vocabulary and the hierarchical structure needs to be handled. Our software handles multiple languages, efficiently handles constantly-incoming new data and scales to several millions of posts (and probably higher). From the applicative perspective, this is one of the rare text visualization tools to be used in political science and the accompanying decision-making process.

6. Conclusions and Future Work

We have presented a scalable, multilanguage text visualization toolbox which integrates with the forum tightly and with minimum effort regardless of its platform. Preliminary testing shows that while relatively simple, the methods produce useful results and provide added value by being packaged together into a convenient tool.

The tool is being promoted and tested by professionals from political and social sciences. As these are non-technical areas, the novelty and perceived usefulness of the software are particularly high. We plan to continue our cooperation with this sector as there seem to be many opportunities left for applications of data mining.

While improvements to the methods in the way of using recent linguistically advanced approaches are unattainable due to lack of linguistic tools for non-english languages, we hope to further improve the software in other areas, e.g. by expanding the set of visualizations with time difference analyses or sentiment analysis. Also, we plan to improve the presentation quality of present methods' results based on user feedback.

Acknowledgments

This work was supported by the IST Programme of the EC under VIDI (EP-08-01-014).

References

J. Ferlež. IST World–machine learning and data mining at work. In *Conference on Data Mining and Data Warehouses (SiKDD 2006), October 9th*, 2006.

B. Fortuna, M. Grobelnik, and D. Mladenić. Visualization of text document corpus. *Informatica, Special Issue: Hot Topics in European Agent Research*, 29:497–502, 2005.

M. Juršič, I. Mozetič, and N. Lavrač. Learning ripple down rules for efficient lemmatization. In *Conference on Data Mining and Data Warehouses (SiKDD 2006), October 9th*, pages 206–209, 2007.

S. Sen, N. Stojanovic, and R. Lin. A graphical editor for complex event pattern generation. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, page 41. ACM, 2009.

B. Shaparenko, R. Caruana, J. Gehrke, and T. Joachims. Identifying temporal patterns and key players in document collections. In *Proceedings of the IEEE ICDM Workshop on Temporal Data Mining: Algorithms, Theory and Applications (TDM-05)*, pages 165–174, 2005.

M. Trampuš, M. Grobelnik, and D. Mladenić. Integrated and evaluated vidi system and system manual. VIDI EU Project Deliverable D2.2, 2010.