Annette Greiner and Sarah Van Wart
12/10/2009

Summarizing Public Opinion on a Topic

1 Abstract

We present SPOT (Summarizing Public Opinion on a Topic), a new blog browsing web application that combines clustering with summarization to present an organized, annotated selection of blog articles related to a user query. SPOT addresses the diversity of subtopics and opinions in the blogosphere that relate to any single topic. It retrieves relevant hits from the Google blog search by using the Google API, clusters them to create logical groupings, and extracts sentences to build a summary of each grouping. We have succeeded in implementing an initial version of the system. While this early prototype does yield useful results, it would benefit from further optimization and experimentation with alternative algorithms.

2 Introduction

Blogs provide a broad range of opinions on issues and events, which are often more personal, informal, and emotional than other text-centric mediums. Arguably, blogs can be seen as a cross section of popular opinion, and can offer different insights into various topics that are not possible to glean from traditional news sources. Blog text has the potential to provide a wealth of valuable information that can be applied to a number of arenas, including marketing, public policy, election campaigns, and education. Thus, providing an easy way to parse the range of opinions on a particular topic in the blogosphere could be an extremely valuable sense-making tool.

Summarizing Public Opinion on a Topic, SPOT, is one attempt to distill the blogosphere's sentiment on a particular topic into groupings of easily understandable sub-categories. By (1) providing a way to dynamically query the blogosphere, (2) grouping resulting blogs into related subgroups or themes, and (3) summarizing each theme, SPOT aims to provide the user with an understanding of the range of events and opinions for a given topic.

This paper describes some of the methods we explored in trying to use unsupervised learning to dynamically catalog blog sentiment and summarize the results. We discuss our use of document vector dimensionality reduction techniques, TF/IDF, and the K-Means clustering algorithm in the development of several models and suggest some future directions for this work.

3 Related Work

As the blogosphere has grown over the past several years, so has the research community's interest in exploring methods for mining, organizing, and clustering blog text. Though research in this area is becoming more prevalent, blog text analysis is still a relatively new field, and there are numerous open questions.

Regarding blog clustering, Li, Shuting, and Zhang, in their 2007 work, downloaded blog pages from a website corpus and performed K-Means clustering. By analyzing both the distribution of documents around cluster centroids (entropy), and the extent to which the documents from each cluster belonged to a single class (from a predetermined set of classes), they found that by assigning larger weight values to blog comments, they were able to produce better clusters (Li et al. 2007). Hayes and Avesani, in their 2007 exploration of topic-relevant blogs, used Spherical K-Means clustering[1] to determine topic clusters and then used individual blog tag sets to find the most topic-relevant blog in a cluster. Though they used intra- and inter-cluster distance to evaluate the clusters themselves, they used Google's page rank results to evaluate the system's ability to determine the most topic-relevant blog (Hayes and Avesani, 2007). In another 2007 project, Elmer et al. did a study to determine the political influence of bloggers in the Canadian election campaign. The project used Google's Blog Search API[2] to retrieve blog posts, and treated the top results returned as a proxy indicator for blog post prevalence and popularity (Elmer et. al. 2007).

Regarding text summarization, there has already been quite a bit of research done in this area. Radev et al. (2000, 2001) used a test of centrality based on distance from a cluster centroid to construct text summaries. The multiple-document summarizer used in those studies, MEAD, uses the centroids computed in the clustering step to summarize sentences. The 2001 work by Radev et al. also incorporated scoring for the positioning of sentences within their documents and their similarity to the first sentence in the document. LexRank (Erkan and Radev, 2004) uses a different take on centrality. It is a graph-based algorithm where nodes represent sentences and edges represent similarities between them. Based on the notion of prestige in social networks, it scores sentences by the number of other sentences in the document with high similarity to it.

SPOT is similar to previous work on a number of fronts. For clustering, SPOT uses TD/IDF techniques with the K-Means clustering algorithm to group similar blogs together, and intra- versus inter-cluster distance to evaluate the effectiveness of the clustering technique. For blog summarization, SPOT also uses a test of centrality from a centroid to extract the best sentences, but it calculates a sentence-based centroid from the sentences in the cluster rather than taking the document-based centroid used in the initial clustering.

SPOT also differs, from existing projects in that it downloads, analyzes, and clusters blog texts on-the-fly from Google's search engine search results. In addition, SPOT provides a way to analyze not only the entire blog texts of returned blog entries, but also the blog titles and blog snippets that are returned from Google.

4 Data and Features

The data that SPOT analyzes consists of the top N results returned from Google's Blog Search API, based on a user-generated free-text query. We made this data choice because we wanted to test a more pragmatic implementation of blog clustering and summarization techniques that could be computed on-the-fly, in order to accommodate the blogosphere's constantly changing content. Since Google is the market leading search engine[3], and the Google API provides a simple way in which to retrieve results, we

[1] A variation of the K-Means algorithm that scales well for large document collections

[2] http://blogsearch.google.com

[3] It is estimated, as of August 2009, that Google commands 64.6% of the search engine market share (http://news.cnet.com/8301-10805_3-10354394-75.html)

felt that this choice would mirror the way that many people retrieve blog posts on the internet.In addition to allowing users to query dynamic data, SPOT provides a number of other useful features: (1) a web-based interface, (2) the various themes that exist within a particular blog topic in the form of clustered blog entries, and (3) and summaries of each of these blog topic themes. From the web interface, a user can type in a search query, submit his/her search, and be presented with grouped lists of URLs, as well as a summary of each grouping. These group summaries are essentially descriptions of the clusters themselves, and can be seen as statistically generated themes for a given topic within the blogosphere.

SPOT provides the user with the ability to analyze blog search results by either parsing the entire contents of each blog (more computationally expensive), or looking at the structured data returned from Google's Blog Search API (faster). Regarding Google's structured data, since the algorithms that determine the description field for each blog are not publicly disclosed, it is difficult to say precisely how this field is populated; however from empirical analysis, Google seems to use a combination of the <description> tag, sentences from the first or last paragraph of the blog text, and sometimes even blog comments.

By logically grouping blog posts on a given topic into themes and summarizing these themes, SPOT aids the user in making sense of the range of events, themes, and opinions.

5 Models and Results

5.1 Clustering

5.1.1 Methods

Sorting blog articles into logical groupings is inherently a clustering task. Since we cannot know what types of groups would be appropriate for a given query in advance, we cannot use classification. We also have no labeled data for training SPOT, so we needed an approach that allowed us to use unlabeled data. Noting the cluster hypothesis—the idea that documents that cluster together meet similar information needs—we feel this is a reasonable approach to finding useful groupings of blog articles and presenting them to a browsing user. Because SPOT must retrieve data while the user waits for a response to her query, it was important to choose a clustering algorithm with good performance. We felt that hierarchical clustering would be too slow for such an application. Thus, SPOT determines the groupings of blog articles by using flat clustering.

We use the most widely used algorithm for flat clustering, K-Means. With this approach, a set of random points are initially chosen as cluster centers, articles are assigned to the cluster with the nearest center, the centroid for each cluster is calculated, and the process repeats with the new set of centroids used as cluster centers. We use 50 repetitions in our K-Means clustering.

In order to create vectors for use by the K-Means algorithm, we treat each blog article as a "bag of words". We remove stop words by checking against a list of stop words, and we do stemming with WordNet's lemmatizer. The resulting words are used to calculate TF-IDF values used in the vectors.

Unfortunately, server configuration issues have thus far prevented the online version of SPOT from being able to use stemming.

Since any clustering technique requires some definition of how many clusters to make, this was an issue we needed to address. We had several options for setting the number of clusters. Our initial version of SPOT allows the user to select the number. We chose this option in development so that we could try different numbers and get a feel for the effects of changing it on system performance as well as clustering results. Ultimately, we would like to implement a version of SPOT that tries a series of cluster numbers and chooses the best based on internal evaluation criteria.

Related to the number of clusters is the number of blog articles gathered for clustering. This number has a strong effect on the speed of returning a result page to the user. Here, again, we chose to allow the user to select the number, in order to get a sense for how many articles we could expect to return in a reasonable amount of processing time. When analyzing entire HTML files retrieved from the Google API, even a small number of articles is handled very slowly. When switching to analyzing only snippets, the number of articles can be larger but still begins to slow with more than 30.

5.1.2 Evaluation

To evaluate the accuracies of our clusters, we compared within-cluster and between-cluster variability. Within-cluster variation was calculated by averaging the Euclidean distances of all the clustered articles from their respective cluster centers. The between-cluster variation was computed by finding a mean centroid vector (a centroid of all the centroids) and calculating the average distance of each centroid from that mean. Our within-cluster variations tended to be larger than our between-cluster variations. (A typical result for two clusters gives an average variation within clusters of 4.42 and an average variation among cluster centroids of 1.75, for a ratio of 0.39.) As we increased the number of clusters, the within-cluster variations decreased and the between-cluster variations increased. We believe that the large variation within clusters results from having widely dispersed cluster members. As the number of clusters increases, the number of items in each cluster decreases. Thus, each cluster is more tightly grouped and its members are less widely dispersed.

We also compared the numbers of articles assigned to each centroid in order to assess the balance of groupings between clusters. We considered more even balances to be good, though with relatively small numbers of articles pulled and a corpus that changes with public whim, it is not unlikely for certain views of a topic to be less well represented in the corpus than others. In addition, we assessed the clustering results qualitatively by inspection, noting whether the cluster groups made sense to a browsing user. We found that some searches were more readily grouped into recognizable clusters than others. It was typical to see at least one grouping that did not make intuitive sense and at least one that did.

5.2 Summarization

5.2.1 Methods

For the summarization of our clusters, we used a centrality-based approach. We chose to find the three most similar sentences to a centroid for the collection of sentences in each cluster. That is, we calculated a centroid from the vectors for all sentences in all the articles in each cluster. We then sorted the sentences by distance from the centroid, taking the three sentences with the smallest distances as our summary. We again used a "bag of words" model to create vectors, but for this task the bags of words were assembled at the level of the sentence. We also removed stop words and used WordNet's lemmatizer for stemming (except that the online version does not do stemming, as noted above).

We tried two approaches for calculating distances between vectors for summarization. Initially, we used Euclidean distance. Later, we switched to using the cosine distance. This change appeared to improve the summary sentences, yielding noticeably fewer of the occasional sentence fragments that our segmenter had parsed incorrectly. The average summary length increased significantly with this change as well.

5.2.2 Evaluation

In order to evaluate the summarization, we looked at the Euclidean distances and cosine distances returned by the summarizer. Since the two methods for measuring distance do not yield numbers that can be directly compared (Euclidean distance being a point-to-point measure and cosine distance being an angular one), we cannot compare the two numerically. The Euclidean distances do provide a sense of the similarity, since a difference of 90 degrees would indicate an orthogonal text. A typical set of the top three sentences, retrieved by SPOT, ranges from 65 to 68 degrees in cosine distance from the sentence centroid.

Qualitative analysis of summarization indicates that our sentence segmenter is failing to correctly handle ellipses and frequently generates sentence fragments. We use the Punkt sentence segmenter as implemented in NLTK. Improving this piece of our pipeline is likely to yield much better clusters as well as better summaries.

Another measure of summarization is compression. Our compression level depends on the number of articles in a given cluster, which in turn is dependent on the number of articles selected and the number of clusters selected. We consistently deliver three sentences, thus delivering greater compression the larger the selection values.

5.3 Overall Interface

As can be seen in Appendix 1, SPOT’s overall interface – a web-based search tool with adjustable numbers of clusters to be formed and URLs to be analyzed – allows for users to view clustering scores, as well as to perform their own empirical analysis of the summarizer as a whole. For example, when we type the word “Obama” into the search form, and asked SPOT to analyze the top 25 URLs returned from the Google Blog Search API and cluster the results into 5 clusters, we see that all of the articles related to the “Nobel Prize” group together.

6 Conclusions

SPOT offers a new way of browsing the blogosphere and getting a sense of what different groups of people are saying about a given query topic. By combining clustering with summarization, we have been able to build a functional, though not yet optimal, tool for online blog browsing.

Our choice of clustering method is constrained by the need for a system that returns results in a reasonable amount of time. Performance issues also affect the selection of the number of blogs to cluster and the number of clusters. Given more development time, we would like to build a version of SPOT that optimizes the selections based on the blog content, yielding a number of clusters that suits the natural grouping of blog articles.

We would have liked to see a higher ratio of between-cluster to within-cluster variation. Experimentation with ways of programmatically finding an optimal number of clusters and an optimal number of blog articles could improve that mark. We could also try using different clustering algorithms, such as hierarchical clustering or variations of K-Means. Using additional features, such as weighting of proper names or words in titles, recognizing elements of document structure in the raw blog text, and removing advertising are all likely to yield improved results.

We would also like to see smaller cosine distances between summary sentences and the centroid of sentences within a cluster. Improved sentence segmentation could yield better results here, as could weighting for sentence position or similarity to initial sentences. Switching the summarizer to a graph-based measure of centrality would be of interest as well.

Finally, we recognize that performance tuning of our code would be of great value, especially since it could make more compute-intensive clustering options (such as hierarchical clustering) viable. We feel that being able to analyze a larger number of blog texts more efficiently would greatly improve the effectiveness of SPOT.

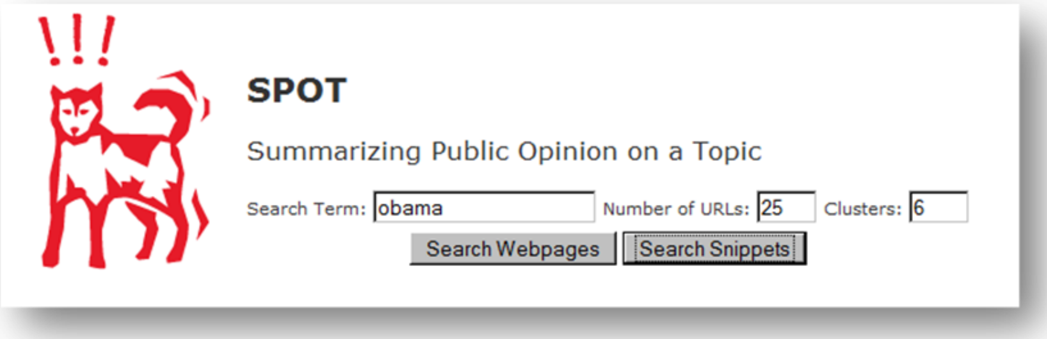
7 References

1. Beibei Li, Shuting Xu, and Jun Zhang, “Enhancing clustering blog documents by utilizing author/reader comments,” in Proceedings of the 45th annual southeast regional conference (Winston-Salem, North Carolina: ACM, 2007), 94-99, http://portal.acm.org/citation.cfm?id=1233359.
2. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, An Introduction to Information Retrieval, Cambridge University Press, online edition, 2009.
3. Dhillon, J. Fan, and Y. Guan. Efficient clustering of very large document collections. In G. K. R. Grossman and R. Naburu, editors, Data Mining for Scientific and Engineering Applications. Kluwer Academic Publishers, 2001.
4. G. Elmer et al., “Election bloggers: Methods for determining political influence,” First Monday 12, no. 4-2 (2007).
5. Günes Erkan, Dragomir R. Radev, "LexRank: Graph-Based Lexical Centrality as Salience in Text Summariztion." Journal of Artificial Intelligence Research, Vol. 22 (2004), pp. 457-479.
6. C. Hayes and P. Avesani, “Using tags and clustering to identify topic-relevant blogs,” in International Conference on Weblogs and Social Media, 2007.
7. D. Radev, S. Blair-Goldensohn, and Z. Zhang, "Experiments in Single and Multi-Document Summarization using MEAD," In First Document Understanding Conference (New Orleans, LA, 2001).
8. D. Radev, H. Jing, and M. Budzikowska, "Centroid-Based Summarization of Multiple Documents: Sentence Extraction, Utility-Based Evaluation, and User Studies." In ANLP/NAACL Workshop on Summarization (Seattle, WA, April 2000).

8 Appendices

8.1 Appendix 1: SPOT in Action

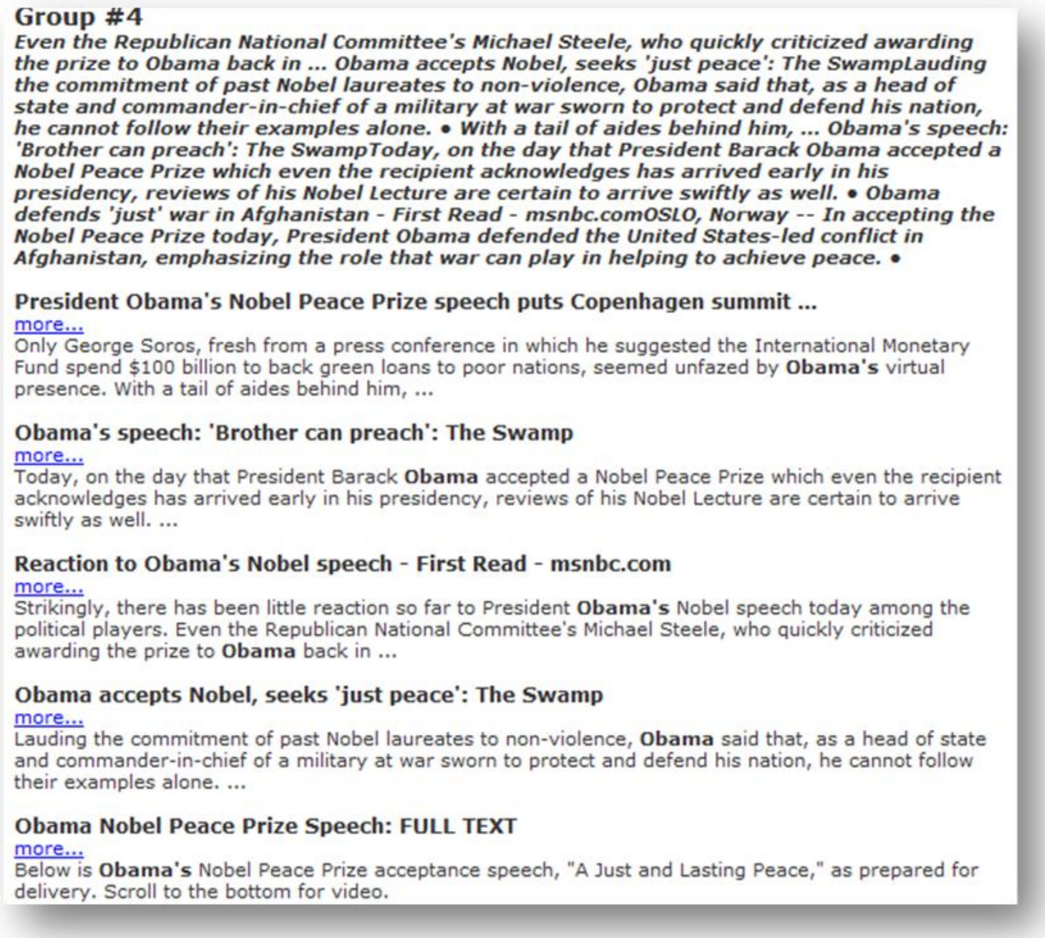
http://groups.ischool.berkeley.edu/spot/spot.html



SPOT

Summarizing Public Opinion on a Topic

Search Term: obama Number of URLs: 25 Clusters: 6



Group #4

Even the Republican National Committee's Michael Steele, who quickly criticized awarding the prize to Obama back in ... Obama accepts Nobel, seeks 'just peace': The SwampLauding the commitment of past Nobel laureates to non-violence, Obama said that, as a head of state and commander-in-chief of a military at war sworn to protect and defend his nation, he cannot follow their examples alone. • With a tail of aides behind him, ... Obama's speech: 'Brother can preach': The SwampToday, on the day that President Barack Obama accepted a Nobel Peace Prize which even the recipient acknowledges has arrived early in his presidency, reviews of his Nobel Lecture are certain to arrive swiftly as well. • Obama defends 'just' war in Afghanistan - First Read - msnbc.comOSLO, Norway -- In accepting the Nobel Peace Prize today, President Obama defended the United States-led conflict in Afghanistan, emphasizing the role that war can play in helping to achieve peace. •

President Obama's Nobel Peace Prize speech puts Copenhagen summit ...
more...
Only George Soros, fresh from a press conference in which he suggested the International Monetary Fund spend $100 billion to back green loans to poor nations, seemed unfazed by **Obama's** virtual presence. With a tail of aides behind him, ...

Obama's speech: 'Brother can preach': The Swamp
more...
Today, on the day that President Barack **Obama** accepted a Nobel Peace Prize which even the recipient acknowledges has arrived early in his presidency, reviews of his Nobel Lecture are certain to arrive swiftly as well. ...

Reaction to Obama's Nobel speech - First Read - msnbc.com
more...
Strikingly, there has been little reaction so far to President **Obama's** Nobel speech today among the political players. Even the Republican National Committee's Michael Steele, who quickly criticized awarding the prize to **Obama** back in ...

Obama accepts Nobel, seeks 'just peace': The Swamp
more...
Lauding the commitment of past Nobel laureates to non-violence, **Obama** said that, as a head of state and commander-in-chief of a military at war sworn to protect and defend his nation, he cannot follow their examples alone. ...

Obama Nobel Peace Prize Speech: FULL TEXT
more...
Below is **Obama's** Nobel Peace Prize acceptance speech, "A Just and Lasting Peace," as prepared for delivery. Scroll to the bottom for video.