SmartNews: Bringing Order into Comments Chaos

Marina Litvak and Leon Matz
Department of Software Engineering, Sami Shamoon College of Engineering, Beer Sheva, Israel

Keywords: Topic Sensitive Page Rank, Query-based Ranking, Comments Retrieval.

Abstract: Various news sites exist today where internet audience can read the most recent news and see what other people think about. Most sites do not organize comments well and do not filter irrelevant content. Due to this limitation, readers who are interested to know other people's opinion regarding any specific topic, have to manually follow relevant comments, reading and filtering a lot of irrelevant text. In this work, we introduce a new approach for retrieving and ranking the relevant comments for a given paragraph of news article and vice versa. We use Topic-Sensitive PageRank for ranking comments/paragraphs relevant for a user-specified paragraph/comment. The browser extension implementing our approach (called SmartNews) for Yahoo! News is publicly available.

1 INTRODUCTION

Most of modern news sites allow people to share their opinions by commenting some issues in a read article and to read what other people write about. However, usually comments are not organized well and appear under one (and sometimes very long) thread in chronological order. Some commenting systems include a rating component, but it is usually based on explicit feedback of users, where comments with the highest average grade (usually measured by the fraction of "thumbs up") or the most popular ones (having the biggest number of references) are displayed on top. Since a comment's rank does not relate to any specific content, and all comments are presented in a non-structured way, it is quite difficult for a reader to follow peoples' opinion about some specific aspect mentioned in the article. The only way he/she can do it, it is to scan manually a huge amount of comments.

In this paper we introduce an approach for ranking comments in news websites relative to a given content (here we refer to a paragraph as an independent text unit describing one of the article's aspects). Our method can be generalized for all comments systems where people refer different aspects in their comments disregarding of domain or language of articles. Since the method includes only very basic linguistic analysis (see section 3.2), it can be applied to websites in multiple languages.

Formally speaking, in this paper we:

- Define an interesting problem of ranking comments relative to a given content;
- Formulate the introduced problem as a query-based ranking and reduce it to the calculating of eigenvector centrality;
- Solve this problem by adapting Topic Sensitive PageRank algorithm;

Since the eigenvector centrality can be computed in a linear (in number of vertices in a graph) time, the computational complexity of our approach depends on graph construction time, that is quadratic in number of comments/paragraphs in a given article.

This paper is organized as follows: section 2 depicts related work, section 3 describes problem setting and our approach, last section contains our future work and conclusions.

2 BACKGROUND

Information retrieval from comments attracted much attention in IR community in recent years. Comments and ratings form a key component of the social web, and its understanding contributes a lot to retrieving important content, ranking and recommending it to the end user. The most known challenge in retrieving comments is managing the doubtful quality of a user-contributed content: many comments are too short, some of them are hardly refer the source content, big portion of comments are written in a poor language.

Litvak M. and Matz L..
SmartNews: Bringing Order into Comments Chaos.
DOI: 10.5220/0004618301910196
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval and the International Conference on Knowledge Management and Information Sharing* (KDIR-2013), pages 191-196
ISBN: 978-989-8565-75-4

Nevertheless, there is a significant volume of recent works have begun steps in the following related directions: comments-oriented summarization (Hu et al., 2008), spam detection (Mishne, 2005; Jindal and Liu, 2008), finding high-quality content (Agichtein et al., 2008), recommending a relevant content (Szabo and Huberman, 2010; Agarwal et al., 2011), improving blog retrieval (Mishne, 2007), and many others. One of the central directions is the ranking comments on the web (Dalal et al., 2012; Hsu et al., 2009), however, none of the works focused on the topic-sensitive ranking of comments. Since in many web domains like news different comments may refer to different aspects of the same article, resolving this problem is very important for structuring and better retrieval of user-contributed content.

In this work, we propose a novel approach to the ranking comments relative to the content they refer to. We provide ranked comments to the user-specified paragraph of a news item and, vice versa, ranked paragraphs that are relevant to a given comment. Our approach is unsupervised and does not require training on an annotated data.

3 SMART NEWS

3.1 Problem Setting

We are given a set of comments $C_1,...,C_m$ referring to an article describing some event and speaking on several related subjects. An article consists of a set of paragraphs $P_1,...,P_n$ speaking on different related subjects. Meaningful words (terms) in all article's paragraphs and comments are entirely described by terms $T_1,...,T_k$. Our goal is, given paragraph P_i, to find a subset $C_{i_1},...,C_{i_r}$ of comments such that[1]

1. These are the most relevant to P_i comments that refer to topics described in P_i itself or comments about it.
2. The comments are ordered by the "relevancy" rank.
3. There are at most M comments.

Our method is based on enhanced eigenvector centrality principle (Topic-Sensitive PageRank, as its variant), that already has been successfully applied to lexical networks for passage retrieval (Otterbacher et al., 2009), question-focused sentence extraction (Otterbacher et al., 2005), and word sense disambiguation (Mihalcea et al., 2004). The intuition behind PageRank utilization on comments (and text in general) is based on its main benefit–node's score is propogated through edges recursively, and as such relevant comments with non-similar content (that is a natural situation in discussion) may be easily discovered. Our approach consists of two main stages: (1) graph constructing and (2) computing the eigenvector centrality. The next two subsections describe both stages, respectively.

3.2 Vector Space Representation Model

According to the VSM (Salton et al., 1975), we represent each paragraph P_i by a real vector $\vec{v}_i = (v_{ij})$ of size k, where k is a vocabulary size and v_{ij} stands for *tf-ipf* (term frequency inverse paragraph frequency) of a term T_j in P_i. Formally speaking, the term frequency is obtained by dividing term's occurence in the paragraph by the total term count in that paragraph, according to the formula

$$tf(t,p) = \frac{tc(t,p)}{|p|}$$

where t is term and p is paragraph. Inverse paragraph frequency is calculated as

$$ipf(t,D) = \log \frac{N}{|p \in D : t \in p|}$$

where N is the number of paragraphs in a document D. In the similar manner, each comment C_i is represented by a real vector $\vec{w}_i = (w_{ij})$ of size k, where w_{ij} stands for *tf-icf* (term frequency inverse comment frequency) of T_j in C_i.

A standard text preprocessing includes HTML parsing, paragraphs segmentation, tokenization, stopwords removal, stemming, and synonyms resolving[2] for articles and their comments. Additionally, to filter "spam" nodes, we remove all comments that have no common terms (considering synonyms) with the related article.

3.3 From Vector Space to Graph Representation Model

In order to represent our textual data as a graph, we relay on the following known factors influencing PageRank and described in (Sobek, 2003):

[1] Here and further, we focus on comments ranking problem, while, generally, our method can be applied to the inverse problem – ranking paragraphs given a comment. Our plugin implements both directions.

[2] With Synonym Map http://lucene.apache.org/core/old_versioned_docs/versions/2_9_1/api/all/org/apache/lucene/index/memory/SynonymMap.html

1. An additional inbound link for a web page always increases that page's PageRank;
2. By weighting links, it is possible to diminish the influence of links between thematically unrelated pages;
3. An additional outbound link for a web page causes the loss of that page's PageRank;[3]
4. There is known effect of "dead-ends"–dangling pages, or cycles around groups of interconnected pages (Strongly Connected Components)–that absorb the total PageRank mass (Avrachenkov et al., 2007).

We start from organizing comments to be ranked as nodes in a graph (denoted by a **comments graph**), linked by edges weighted with text similarity score calculated between nodes.[4] Formally speaking, we build a graph $G(E,V)$, where $N_i \in V$ stands for a comment C_i, and $e_k \in E$ between two nodes C_i and C_j stands for similarity relationship between texts of the two comments.[5] We measure the cosine similarity (Salton et al., 1975) between real vectors of length k $\vec{v} = (v_i)$ and $\vec{w} = (w_i)$ representing two text units[6] as follows.

$$sim(\vec{v},\vec{w}) == \frac{\sum_{i=1}^{k} v_i \times w_i}{\sqrt{\sum_{i=1}^{k} v_i^2} \times \sqrt{\sum_{i=1}^{k} w_i^2}}$$

Each edge e_l is labeled by a weight w_l equal to the similarity score between the linked text units. Edges with a weight lower then a pre-defined threshold are removed. According to the rule 2, by weighing links we diminish the influence of links between thematically unrelated text units and, conversely, increase the influence of links between strongly related ones. An example of resulted comments graph is demonstrated in Figure 1(a).

We treat a paragraph as a query that must to influent the resulted ranks of comments. We add an additonal node (denoted by a **query node**) for the paragraph with respect to which the comments should be ranked. The query node is also linked to the comments nodes by similarity relations, with weighted edges directed from a query node to comment nodes.

[3]Rules 1 and 3 are considered independently.

[4]Since mumber of comments may vary from tens to thousands, we limit their amount by configurable number (60 in the current version).

[5]For the inverse problem, we represent a document as a graph of paragraphs (aka **paragraphs graph**) linked by a similarity relationship (Salton et al., 1997).

[6]The cosine similarity is measured between each pair of comments and comments with a query paragraph in the **extended graph**.

According to rule 1 and rule 2, adding weighed inbound links from the query node to thematically related comment nodes must increase their PageRank relative to other nodes. Here and further, we call the resulted graph **extended graph**. This stage is demonstrated in Figure 1(b).

According to rule 4, applying PageRank on the resulted extended graph might have undesirable side effect in the following situation. Consider comments graph with a group of strongly connected nodes (denoted as SCC in graph theory), mostly thematically irrelevant to a query node (see Figure 2(a)). This situation is created when we have comments "talking" to each other and deviate from the main (query) topic. It is enough that only one node from a group will be linked to a query node for "grabbing" a query's rank to a group and, at each iteration, enlarging the PageRank of strongly connected nodes. In order to avoid (1) PageRank increasing in unrelated nodes linked with related ones in a closed system and (2) "leakage" of PageRank in a query node, we add outbound links from comment nodes to a query node, according to the rule 3. For uniform impact on all comment nodes, we give all edges the same weights of 1. Comment nodes that are strongly related to a query, will gain their PageRank back in each iteration due to a high weight assigned to inbound links from a query node, while irrelevant nodes will "loose" their PageRank irretrievably. The final graph is demonstrated in Figure 2(b). The same update applied to a graph from Figure 1(c) will result in a new structure depicted in Figure 1(d).

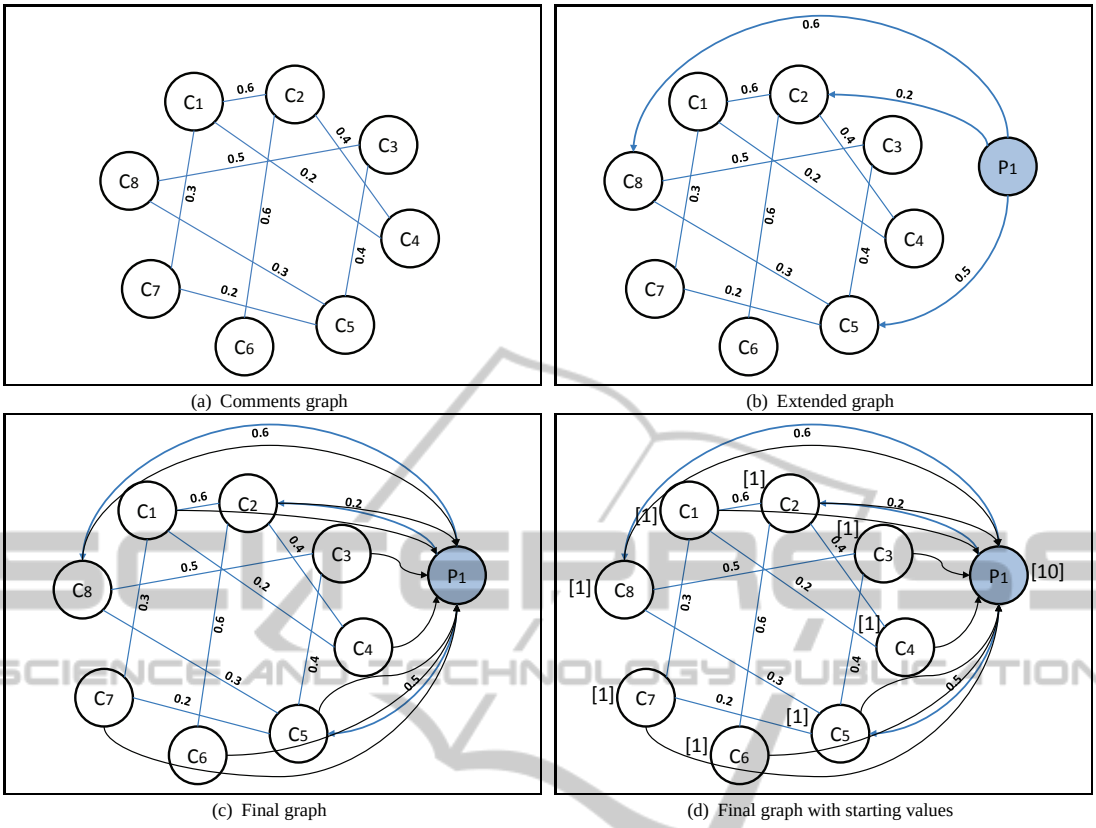
3.4 Computing the Eigenvector Centrality

In order to rank and retrieve comments, we apply PageRank algorithm (Brin and Page, 1998) to an extended graph. PageRank $PR(A)$ of page A is given by

$$PR(A) = (1-d) + d\sum_{i=1}^{n} \frac{PR(T_i)}{C(T_i)}$$

where $PR(T_i)$ is the PageRank of pages T_i which link to page A, $C(T_i)$ is the number of outbound links on page T_i, and d is a damping factor which can be set between 0 and 1. So, PageRank is determined for each page individually. Further, the PageRank of page A is recursively defined by the PageRank of those pages which link to page A.

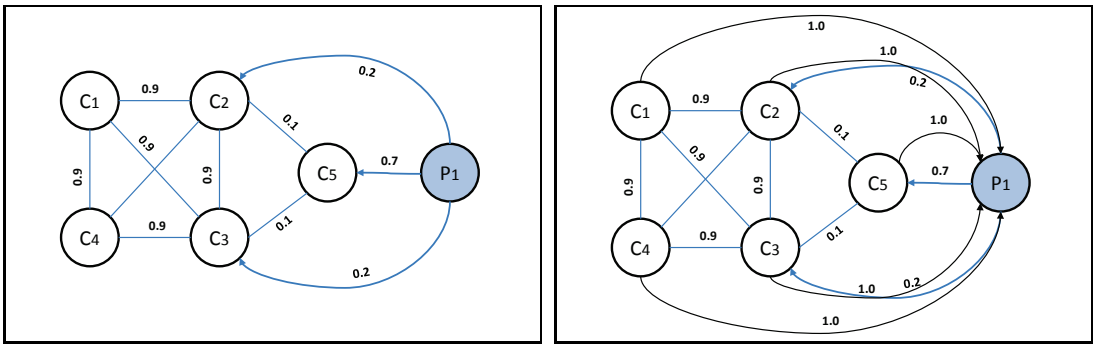
In this setting, our goal can be reformulated as the problem of finding subset $N_1,...,N_k$ of nodes standing for comments $C_1,...,C_k$ in an extended graph G, so that the comments represented by these nodes are



(a) Comments graph (b) Extended graph

(c) Final graph (d) Final graph with starting values

Figure 1: Graph representation: four steps.



(a) Strongly connected component (b) Updated graph

Figure 2: Strongly connected component: problem and its solution.

most relevant for the given paragraph represented by a query node. In order to influence nodes' rank by a query node, we apply several modifications to a PageRank algorithm, according to the known factors influenting PageRank score which are enumerated below and described in (Sobek, 2003).

1. If the computation is performed with only few iterations, the higher starting values assigned to certain websites before the iterative computation of PageRank begins would influence that pages' PageRank;
2. Assigning the different damping factors for webpages increases PageRank for pages with higher factor values and decreases PageRank for those with lower values (known as Yahoo bonus or Topic Sensitive PageRank).

According to the rule 1, we give a high starting value to a query node before the iterative computation of PageRank begins. Adding outbound links from comment nodes to a query node (described above) helps to keep high PageRank in the query node through successive iterations. The final graph structure including

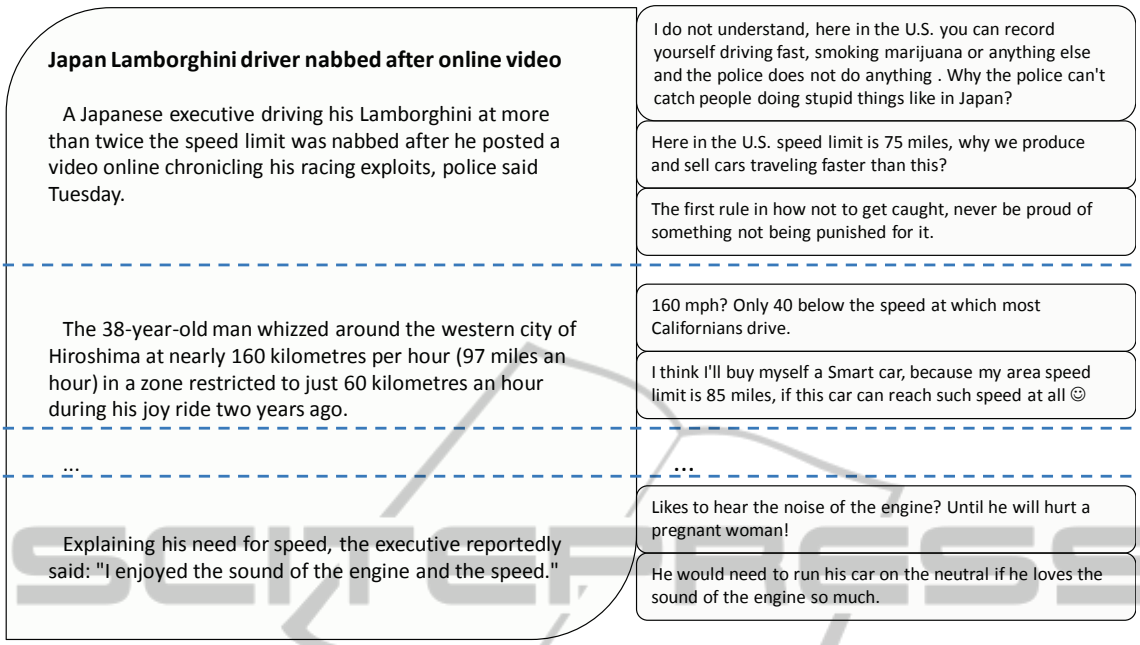


Figure 3: Textual example: article and its comments.

initial starting values is shown in Figure 1(d).

In order to implement a theme-based retrieval, we adapt the idea of Yahoo Bonus or Topic-Sensitive PageRank (see rule 2), where the thematically relevant comments get higher damping factor. In our approach, the damping factor is set proportionally to the text similarity E between a query and a comment nodes.[7]

$$PR(A) = E(A)(1-d) + d\sum_{i=1}^{n}\frac{PR(T_i)}{C(T_i)}$$

For example, if a user is interested in retrieving the comments relevant to the paragraph talking about *victims* in Tohoku earthquake[8], all comments semantically related to this topic will receive a relatively higher value of E and recursively "pass" this value as a PageRank to the pages which are linked to. Of course, if we assume that the related comments tend to link to other comments within victims topic, comments on that topic generally will receive a higher score.

Again, the motivation of applying the Topic-Sensitive PageRank in our setting, is avoiding high ranking for the groups of less relevant inter-connected comments, and comments with many similar comments, while increasing the influence of the theme relevance (comment-paragraph similarity).

The Topic-Sensitive PageRank can be used in our setting, since we retrieve comments *with respect to* a given paragraph representing a topic an actual user is interested in. The actual paragraph a user is interested in is identified by sending the position of the user's mouse (upon user's click) to the server.

We treat a PageRank score as a final rank of items. In a greedy manner, we extract and present at most M most ranked comments ordered by their rank to the end user. In our settings, $M = 5$.

4 CONCLUSIONS AND FUTURE WORK

In this paper we present an application based on a new approach for the topic-sensitive ranking of comments helping the end user to better understand and analyse the content contributed by other users on the web. Our approach is based on computing the eigenvector centrality and the factors influencing the centrality score. The introduced approach is unsupervised and does not require the annotated data. The example of article text and the most ranked comments, per paragraph, can be seen in Figure 3. More examples are provided in http://goo.gl/7idNw. It can be seen that the comments

[7] We normalize the E values so that the average over all pages is 1, and the PageRank average continue to converge to 1.

[8] We suppose, that an article giving overview of such event, will consist of several paragraphs on different topics like earthquake characteristics, location, repercussion, victims, humanitarian help provided by different countries, etc.

are very related to the paragraphs content and, moreover, they relates the **subject** of a paragraph as well as a **discussion** and **opinions** it arises, beyond the text overlapping. Such performance is provided by a recursive nature of PageRank, where the relationships between comments are iteratively elaborated. Unlike this approach, ranking comments by a (text) similarity to a given paragraph would not retrieve related comments with a different vocabulary.

The plugin implementing our approach is publicly available from http://goo.gl/To4Rd.[9] In future, we intend to evaluate our system by comparing it to the other state-of-the-art ranking techniques.[10]

ACKNOWLEDGEMENTS

Authors thank project students: Maxim Magaziner, Anatoly Shpilgerman and Sergey Pinsky for implementing the introduced approach as a Chrome Extension for Yahoo! News[11] website, and Igor Vinokur for a technical support of the software. Especial thanks to Dr. Amin Mantrach from Yahoo! Labs, Barcelona, for very constructive and helpful comments.

REFERENCES

Agarwal, D., Chen, B.-C., and Pang, B. (2011). Personalized recommendation of user comments via factor models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 571–582.

Agichtein, E., Castillo, C., Donato, D., Gionis, A., and Mishne, G. (2008). Finding high-quality content in social media. In *Proceedings of the international conference on Web search and web data mining*, WSDM '08, pages 183–194.

Avrachenkov, K., Litvak, N., and Pham, K. S. (2007). Distribution of pagerank mass among principle components of the web.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Dalal, O., Sengemedu, S. H., and Sanyal, S. (2012). Multi-objective ranking of comments on web. In *Proceedings of the 21st international conference on World Wide Web*, pages 419–428.

Hsu, C.-F., Khabiri, E., and Caverlee, J. (2009). Ranking comments on the social web. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04*, pages 90–97.

Hu, M., Sun, A., and peng Lim, E. (2008). Comments-oriented document summarization: Understanding documents with readers feedback. In *In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR 08. ACM*.

Jindal, N. and Liu, B. (2008). Opinion spam and analysis. In *Proceedings of the international conference on Web search and web data mining*, WSDM '08, pages 219–230.

Mihalcea, R., Tarau, P., and Figa, E. (2004). Pagerank on semantic networks, with application to word sense disambiguation. In *In Proceedings of The 20st International Conference on Computational Linguistics (COLING 2004)*.

Mishne, G. (2005). Blocking blog spam with language model disagreement. In *In Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*.

Mishne, G. (2007). Using blog properties to improve retrieval. In *In Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*.

Otterbacher, J., Erkan, G., and Radev, D. R. (2005). Using random walks for question-focused sentence retrieval. In *In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP*, pages 915–922.

Otterbacher, J., Erkan, G., and Radev, D. R. (2009). Biased lexrank: Passage retrieval using random walks with question-based priors. *Inf. Process. Manage.*, 45(1):42–54.

Salton, G., Singhal, A., Mitra, M., and Buckley, C. (1997). Automatic text structuring and summarization. *Information Processing and Management*, 33(2):193–207.

Salton, G., Yang, C., and Wong, A. (1975). A vector-space model for information retrieval. *Communications of the ACM*, 18.

Sobek, M. (2003). A Survey of Google's PageRank. http://pr.efactory.de/.

Szabo, G. and Huberman, B. A. (2010). Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88.

[9]Unzip the archive, press "Load unpacked extension" in "Developer mode" of chrome "Extensions" tool, and choose the unzipped plugin folder.

[10]Currently, we are performing an experiment aimed at creating the Gold Standard collection of ranked comments. Since it is a very time/labor/budget-consuming process, we are expecting to be able to run evaluations only in several months.

[11]http://news.yahoo.com/