Blog Distillation
via Sentiment-Sensitive Link Analysis

Giacomo Berardi, Andrea Esuli, Fabrizio Sebastiani, and Fabrizio Silvestri
Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
56124 Pisa, Italy
E-mail: {firstname.lastname}@isti.cnr.it

Abstract

In this paper we report a new approach to blog distillation, defined as the task in which, given a user query, the system ranks the blogs in descending order of relevance to the query topic. Our approach is based on the idea of adding a link analysis phase to the standard retrieval-by-topicality phase. However, differently from other link analysis methods, we try to analyse whether a given hyperlink is a citation with a positive or a negative nature, i.e., if it expresses approval or disapproval of the linked page by the linking page. We report the results of testing our method on the Blogs08 collection used in the 2008 and 2009 editions of the TREC Blog Track.

1 Introduction

Blog distillation is a subtask of the blog search task. It is defined as the task of ranking in decreasing order of relevance the set of blogs in which the topic expressed by the query q is a recurring topic of interest. Blog distillation has mainly been tackled within the TREC Blog track [14, 15], where participants have experimented with various combinations of (i) methods for retrieval by topicality and (ii) sentiment analysis methods. Retrieval by topicality is needed since topicality is a key aspect in blog distillation, while sentiment analysis is needed since blogs tend to contain strongly opinionated content, which makes the analysis of opinions (an aspect orthogonal to topic) necessary.

In this paper we propose a strategy in which, on top of a standard method for retrieval by topicality, we add a link analysis phase, which is meant to account for the reputation of the blog. Link analysis has been extremely popular in Web search in the late '90s and early '00s [4], due to the success of the PageRank algorithm and the Google search engine, which had PageRank at the center of

1

its ranking strategy. However, link analysis has witnessed a somehow decreased interest in the late '00s, mainly due to the fact that the "link-as-endorsement" hypothesis (i.e., the hypothesis that the presence of a hyperlink denotes an endorsement of the linked page on the part of the linking page) is less and less justified in the Web at large, due to the fact that Web pages are now frequently generated by automated Web authoring software, rather than by people.

This work is based on the assumption that, unlike in the Web at large, the "link-as-endorsement" hypothesis can still be assumed true in the blogosphere, largely due to the fact that blogs and blog posts are authored by humans. However, due to the highly opinionated nature of blog contents, we must consider that many hyperlinks express *disapproval*, and not approval, of the linked post on the part of the linking post; as a result, a simplistic link analysis which takes all hyperlinks as expressing approval might actually lead to unintuitive results. We attempt to solve this problem by performing *sentiment-sensitive link analysis*, i.e., a random-walk process in which links are seen as representing either endorsements (positive) or rebuttals (negative) and are thus weighed according to their sentimental valence (or "polarity"). We establish whether a given hyperlink transmits positive or negative endorsement by performing sentiment analysis on a text window around the hyperlink.

This paper is organized as follows. In Section 2 we present our approach to blog distillation. Section 3 presents the results of experiments we have performed on the Blogs08 collection used in the 2008 and 2009 editions of the TREC Blog Track. Finally, Section 4 presents related work.

2 Sentiment-Sensitive Link Analysis

Let $B = \{b_1, \ldots, b_n\}$ be a collection of blogs, let each blog b_i be a set of blog posts $b_i = \{p_{i1}, \ldots, p_{im_i}\}$, and let q be a query (which will typically be expressed as a sequence of words). The relevance of a blog to a query q is typically computed in terms of the relevance of its posts to q. In this paper we propose to compute blog relevance via the combination of a classic method for computing topical relevance (e.g., BM25) and a link analysis method. We first describe our link analysis method, after which we will describe the combination of the scores resulting from the two methods.

Let $R_q = \langle \beta_1, \ldots, \beta_k \rangle$ be the list of the k top-ranked blogs from a blog collection B as returned by a topicality-based retrieval system for query q . Each retrieved blog $\beta_i \in B$ is then associated with **retrieval**$_i$, a score returned by the topicality-based retrieval system. In our method, as we shall see in the remaining part of this section, associated with each blog $\beta_i \in B$ and with each blog post $p_{ij} \in \beta_i$ are two *link-based scores* $\mathbf{r}_i^{\text{blog}}$ and $\mathbf{r}_{ij}^{\text{post}}$, respectively. They are computed on the graph formed by considering blogs (or posts) as vertexes and hyperlinks as edges. Link-based scores are obtained by applying a particular implementation of the *random walk with restart* (RWR) procedure [21], which mimics the behaviour of a generic blog surfer that reads a post and decides to follow a link

only if she is convinced to do so by the text surrounding the link (the *anchor text*) ; in this case we define a link to be *positive*. If the anchor text discourages the surfer from following the link, then we define the link to be *negative*; this is the case, e.g., when the anchor text contains a sharp critique of the linked post. When the anchor text surrounding the link contains neither a clear positive nor a clear negative appreciation of the linked page, then we defined the link as *neutral*; in this case, the blog surfer has a behaviour typical of the random surfer of "classical" link analysis.

To model the behaviour of this blog surfer we have modelled the post (resp., blog) graph $G = (V, E)$ as the union of two graphs, namely $G^+ = (V^+, E^+)$ and $G^- = (V^-, E^-)$, where V is the set of nodes representing all the posts (resp., blogs) analysed, and $(v_1, v_2) \in E$ if there exists a link interconnecting the posts (resp., blogs) represented by v_1 and v_2. The edges in G are subdivided in two subsets: (i) E^+ (resp., E^-,) is the set of edges with a polarity greater (resp., less or equal) than a given threshold λ, with V^+ (resp., V^-) the set of nodes that are endpoints of at least one edge in E^+ (resp., in E^-). Each edge $(v_i, v_j) \in E$, thus also in E^+ and E^-, is weighed by the sentiment associated to that link. By $p(v_i, v_j)$ we denote the probability of transition in G^+; we set $p(v_i, v_j) = 0$ if $(v_i, v_j) \notin E^+$. We discuss in Section 2.1 the details of how weights are computed according to a sentiment analysis method. Each polarity score s_{ij} is turned into a transition probability $p(v_i, v_j)$ in G^+, while each weight in G^- is equal to the absolute value of s_{ij} if $(v_i, v_j) \in E^-$, 0 otherwise.

On the nodes of G^+ we compute a $r^+[v]$ value obtained by performing a RWR from the nodes in V^+. Restart probabilities are uniformly distributed among the nodes corresponding to posts in the result set R obtained by the baseline retrieval method (as described above). Actually, in order to compute the random walk we apply a refined version of the popular "power method": (1) we build the matrix M^+ associated with G^+ by naturally considering $m_{ij}^+ = p(v_i, v_j)$, and (2) we partition M^+ into $K + 1$ blocks as described by [11] (the "+1" block contains all the dangling nodes, that are treated separately). See [11] for more details.

The final vector of link-based scores $\mathbf{r}$ is obtained from $\mathbf{r}^+$ by applying the transformation $\mathbf{r} = \mathbf{r}^+ \cdot (1 - \theta(M^-)^T \mathbf{r}^+)$ through the matrix M^- associated with G^-. In the transformation, θ is a smoothing factor that is used to mitigate the effect of negatively weighed anchor texts.

Finally, the overall score of a blog with respect to a query q, is computed according to the following procedure:

1. Let $\mathbf{r}^{\text{post}}$ and $\mathbf{r}^{\text{blog}}$ be the stationary distributions of the post and blog graphs;

2. Retrieve the top-n post set R ranked according to a BM25 score on the query q;

3. Compute a relevance score for each retrieved post j as

$$\mathbf{score}_j^{\text{post}} = \mathbf{retrieval}_j + w\mathbf{r}^{\text{post}}[j]$$

3

where w is the weight assigned to the link analysis score;

4. Finally, compute the relevance score for the a blog b_i as

$$\textbf{score}_i^{\text{blog}} = \left(\sum_{j \in B_i} \frac{\textbf{score}_j^{\text{post}}}{|B_i|} \right) + \left(w\textbf{r}^{\text{blog}}[i] \frac{|R \cap B_i|}{|B_i|} \right)$$

where w is again the weight assigned to the link analysis score, and B_i is the set of blog posts of blog i.

The top-r scoring blogs are returned for the query q.

2.1   Sentiment-based Edge Weights

The RWR methods we propose is based on the assumption that any edge in the graph G is assigned a weight s_{ij} that identifies the sentiment-related properties of the links that the edge represents: a positive (or negative) value for s_{ij} indicates a positive (or negative) attitude toward the linked document, and the absolute value of s_{ij} indicates the intensity of that attitude.

In order to automatically assign sentiment scores to links we have built a simple sentiment analysis system that is based on recognizing *sentiment patterns* in the text surrounding the link (including the text that defines the link). The sentiment pattern recognition component is based on POS tagging the text[1] and selecting as candidate patterns all the sequences of words matching the `(RB|JJ)+` and `NN+` patterns. A sentiment score is assigned to each pattern after asking SentiWordNet [3] to assign a sentiment score to each term in the pattern. The pattern is then checked for the presence of *sentiment modifiers*, e.g.:

- negators (polarity inversion) (e.g., "no", "not");

- intensifiers and diminishers (e.g., "very", "strongly", "barely", "almost").

As the resource for sentiment modifiers we have used the appraisal lexicon defined in [2]. When a modifier is found, the score of the word that follows it is modified accordingly (e.g., "very good" is assigned a doubly positive score than "good"). The sentiment scores for all the words in a pattern p are summed up, taking into account modifiers, thus resulting in a sentiment score s_p for the pattern.

All the sentiment scores for patterns that appear in the same sentence of the link are summed up to determine the sentiment-based weight of the link. The sum function takes into account the distance of the pattern from the link, modelling the hypothesis that the closer a sentiment-pattern is to the link, the more it is related to it. Distance is proportional to the discourse function of the words between pattern and link (e.g., "than" divides two subjects in a comparison, it increases the distance), or proportional to their part of speech.

---

[1]For this we have used the Natural Language Toolkit available at `http://www.nltk.org`

The sentiment weight assigned to edge s_{ij} is equal to the sentiment score assigned to the relative link. If an edge is related to more than one link in the text, its weight is equal to the sum of the sentiment scores of all the links (if both positive links and negative links are present, they compensate each other).

3 Experiments

3.1 Experimental setting

3.1.1 The dataset

We have tested our method on the Blogs08 collection used in the 2008 and 2009 editions of the TREC Blog Track [15]. Blogs08 consists of a crawl of 1,303,520 blogs, crawled from 14 Jan 2008 to 10 Feb 2009. The crawl identified a total of 28,488,766 blog posts, each identified by a unique URL. Each blog post has been effectively downloaded two weeks after its first identification from the crawler, in order to include also a number of comments about the post from the readers of the blog. For our experiments we have followed the protocol of the 2009 Blog Track, using the 50 queries of 2009 and their relevance judgments [14].

3.1.2 Links graphs

We have processed the links in the collection in order to produce two graphs:

- a graph of blog posts, in which the nodes of the graph are the blog posts and a directed edge from post x to post y exists iff x contains a link to y;

- a graph of blogs, in which the nodes of the graph are the blogs and a directed edge from blog X to blog Y exists iff a post x from blog X contains a link to a *homepage* of the blog Y.

The graph of blog posts contains 4,697,700 nodes (which means that about 60% of the posts are not linked in any way to other posts) and 12,633,788 edges. Considering an undirected version of the graph of blog posts, it is composed of 257,227 connected components, with the largest one composed of 3,985,132 nodes. The graph of blogs is composed of 634,313 nodes (with the largest connected component consisting of 628,670 nodes) and 5,533,981 edges.

3.1.3 Link polarity

All links relative to the edges composing the two graphs have been processed by our link polarity detector (see Section 2). The processing of links relative to the graph of blog posts resulted in the identification of 8,947,325 neutral links (i.e., polarity weight equal to zero), 1,906,182 positive links, and 1,780,281 negative links. In case an edge between two nodes is determined by more than one link, the polarity value for the edge is determined as the average of the polarities of
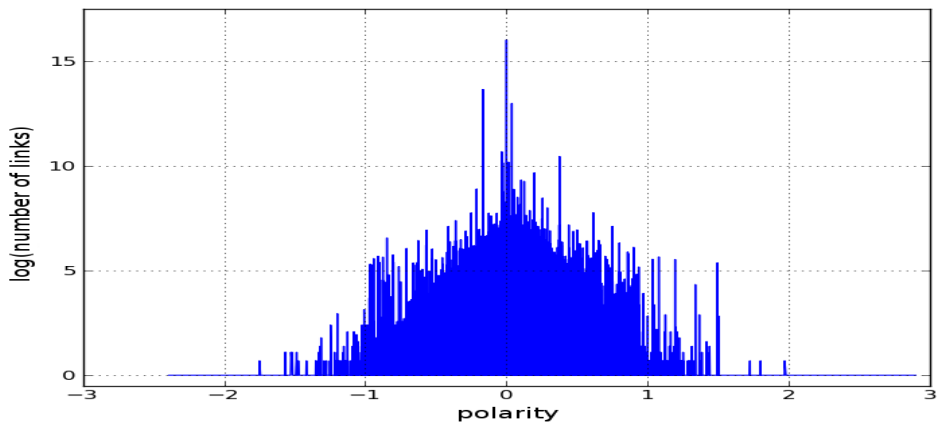
Figure 1: Distribution of polarity scores for the edges in the blog posts graph

the links. Figure 1 shows the distribution of polarity scores for the edges in the blog posts graph.

3.1.4 Evaluation measures

The experimental results have been evaluated by using the two evaluation measure that are typically adopted in the Blog Track, the *mean average precision* (MAP) and the *binary preference* (bPref) [1]. MAP is defined as the weighted average of the precision scores obtained considering the first k results, for all the relevant values of k, i.e., $MAP = \frac{1}{|R|} \sum_{1 \leq k \leq |C|} r_k \cdot precision(k)$, where R is the set of documents that are relevant with respect to the query, C is the whole collection, and $r_i = 1$ if the document in position i in the results is relevant with respect to the query, $r_i = 0$ otherwise. Precision is defined as $precision(k) = \frac{1}{k} \sum_{1 \leq i \leq k} r_i$. We will use the notation P@k as a shortcut for $precision(k)$.

The bPref measure [1] is proportional to the average number of time in which a non-relevant document appears in the ranking after a relevant document, i.e., $bPref = \frac{1}{|RN|} \sum_{1 \leq k \leq |C|} r_k \cdot |\{j \mid r_j = 0, j > k\}|$, where N is the set of non-relevant documents in the collection with respect to the query. The highest possible value both for MAP and bPref is 1, and it is returned when all the relevant documents are places on top of the non-relevant ones in the ranking.

3.1.5 Experimental protocol

As the indexing system for the Blogs08 collection we have used Terrier [18]. With Terrier we have built a baseline for the comparison of our results, based on ranking the documents in the collection by their BM25 relevance with respect to each query. This rankings have been used also as the input to the RWR-based reranking phase of our method.

As the damping factor for the RWR method we have used the typical value $\alpha = 0.85$ [5], we leave to future work the optimization of this parameter. The epsilon value used to stop the iterative process is set to $\epsilon = 10^{-9}$.

We have tested two sets of values for the λ parameters that determine the thresholds for separation of the graphs into the two graphs G^+ and G^- used by our method:

- Λ_1, $\lambda = -0.01$ for the blog posts graph, and $\lambda = -0.005$ for the blogs graph;

- Λ_2, $\lambda = -0.1$ for the blog posts graph, and $\lambda = -0.02$ for the blogs graph.

The Λ_1 pairs of value have been set in order to have the 87% of G edges in the G^+ graphs. The Λ_2 values produce a split in which the 91% of G edges belong to the G^+ graphs.

For each query, in the collection we retrieve the first million results, ranked by TF-IDF. This million posts are set as the nodes with non-null restart probabilities given as input to the RWR method.

We have tested various combinations for the θ, w, and the n parameters:

- θ is the smoothing factor that is used to mitigate the effect of negatively weighed anchor texts, we have testes the values 0.1 and 1.0. The first value is chosen to obtain a minimal effect while the second to obtain the maximum effect.

- w is the relative weight assigned to the RWR score with respect to the BM25 score assigned by the baseline system. We have tested $w = 2, 3, 4, 5$. These values have been chosen empirically, we have not obtained better reranking with different weights.

- n is the number of posts in the baseline BM25-based ranking that are reranked by using the RWR scores. We have tested $n = 1000, 2000, 3000, 4000$. These values have been chosen empirically. Adding more not relevant posts can introduce noise in the reranking.

3.2   Results

Evaluations with different combination of parameter values does not report big differences. This can been understood in table 2, looking at the differences among best and mean evaluations.

The importance of negative scores denoted by θ does not seem relevant to the final ranking; changing the two different values for θ does not bring improvements. We can claim that the contribution of G^- is minimal, and the random walk ranking is solid enough to obtain definitive results. Also the graph cut with λ, and the weight parameter w, do not have a sensible effect on results.

n	1,000	2,000	3,000	4,000
MAP	0.1775	0.1914	**0.1958**	0.1949
P@5	**0.3347**	0.3143	0.2980	0.2939
P@10	**0.2878**	0.2796	0.2653	0.2592
bPref	0.2039	0.2203	**0.2226**	0.2210

Table 1: Evaluation of the BM25 baseline. Boldface indicates best results for each evaluation measure.

n	1,000		2,000		3,000		4,000	
	$w = 5, \Lambda_2$	mean	$w = 3, \Lambda_2$	mean	$w = 4, \Lambda_1$	mean	$w = 3, \Lambda_1$	mean
MAP	0.1802	0.1801	0.1943	0.1942	0.1986	0.1983	0.1977	0.1976
P@5	0.3347	0.3347	0.3184	0.3173	0.3020	0.2980	0.2980	0.2959
P@10	0.2898	0.2880	0.2837	0.2819	0.2735	0.2719	0.2673	0.2650
bPref	0.2056	0.2050	0.2222	0.2220	0.2247	0.2242	0.2230	0.2224

Table 2: Best (in terms of MAP plus P@10) and mean evaluations of the Random Walk rankings. Best evaluations are showed with relative parameters; mean evaluations are computed on the different values of w and division of graphs by Λ_1 and Λ_2.

Results show an improvement over retrieval baseline. This improvement however is not heavily significant, this is due to the sparsity of graphs. Retrieved posts are often not present in the posts graph, so their scores affect a little part of results. Using only blog scores follows better evaluations in some cases.

Rising the n parameter increase improvement over baseline (i.e. P@10), because more posts are covered by Random Walk scoring, MAP increases too. Rising n too much introduces less relevant posts in results which decrease improvements.

In order to understand the benefit of using Sentiment Analysis, we have examined a comparison of our Random Walk with a PageRank implementation. PageRank is computed on the whole graph; edges are weighted with uniform probability of transition: $1/outdegree(node)$. Differences in the evaluations are insignificant, we show an example in table 3. The improvement of PageRank is probably due to the graph coverage of posts, in fact PageRank is calculated on a bigger graph in terms of nodes. Our algorithm however is faster than PageRank, because it works on the reduced transition matrix given by G^+.

w	2	3	4	5	6
RWR	**0.1981**	0.1983	**0.1986**	0.1983	0.1983
PR	0.1978	0.1983	0.1985	**0.1988**	**0.1987**

Table 3: Comparison of MAP evaluations between our Random Walk with Restart (RWR) and PageRank (PR), with $\theta = 1.0$, Λ_1 and $n = 3,000$. Boldface indicates best results for each weight.

	MAP	bPref	P@10
buptpris [13]	0.2756	0.2767	0.3821
ICTNET [23]	0.2399	0.2384	0.3513
USI [9]	0.2326	0.2409	0.3308
RWR	0.2032	0.2349	0.3103
FEUP [20]	0.1752	0.1986	0.3282
uogTr [16]	0.1317	0.1531	0.2333
UAms [22]	0.0803	0.0966	0.1590
knowcenter [12]	0.0624	0.0742	0.1410

Table 4: Evaluations of participants at TREC Blog Track 2009, ordered by MAP. RWR is our algorithm with $n = 3,000$, $w = 4$, Λ_1, $\theta = 1$.

	λ	iterations	time (secs)
posts graph	-0.01	107.8	145.7
	-0.1	107.6	153.9
PageRank on posts graph		107.9	160.5
blogs graph	-0.005	94.5	38.3
	-0.02	94.1	41.8
PageRank on blogs graph		94.2	47.8

Table 5: Average number of iterations and average time required by our RWR method and PageRank to converge.

We finally show a comparison with participants' approaches at TREC blog track 2009 (table 4). Evaluations are made on the official subset of queries, it consists of 40 topics. For each participant the run with the highest MAP is selected, we have chosen algorithm parameters according to this TREC rule. Blog Track participants have an approach similar to ours in the retrieving phase. They index post documents and the often obtain a baseline ranking, which is reordered with their algorithms. No one takes advantage of link analysis; they use statistical techniques on terms, improve retrieval with *query expansion* methods, exploit page features like temporal informations.

Table 5 reports the average number of iterations and the average time required by our RWR method and PageRank to converge.

4   Related work

Blog search has been widely studied in the last years, especially in the context of the TREC Blog Track. The latter consists of different subtasks, one of which is blog distillation, which consists of returning blogs which are relevant by topic to a given query. TREC Blog Track participants have different approaches to the distillation task: most of them do not pay attention to quality- and authority-

related aspects of blogs and mainly use IR and statistical techniques, such as language models and/or query expansion [14, 15]. However, quality-related aspects are important in the blog domain, since content is user-generated and the authoritativeness of authors is thus highly variable [7].

Link analysis in blog distillation. A random walk algorithm has been used in [10] on the TREC Blogs06 collection for the blog distillation task. It is computed on a graph in which vertices are either blogs, or posts, or terms, in order to find relations between blogs and query terms. An edge between a blog and a post shows membership of that post in that specific blog and an edge between a post and a term means the term occurred in that specific post. There is an edge between two posts if they are connected by a hyperlink, or if they are in the same blog. The score attributed to each blog is proportional to the sum of the probabilities of reaching each query term in a predefined number of steps, starting from the blog node. In our work the blog score used for the final ranking is obtained via a linear combination of the score assigned by the random walk to the blog itself and to the posts contained in it. We use sentiment analysis to weight links, while in [10] the transition probabilities are uniform (except for the weights from posts to terms, which are calculated via the $tf * idf$ function).

In [17] different scoring strategies are used to rank blog posts. One strategy is to consider post authority by analysing link structure. [17] does not perform a random walk, but implements a similar notion via the use of "post in-degree". The results of [17] show that post in-degree does not improve accuracy. Our experience has actually been different, since in our experiments blog reranking via random walk always improves accuracy with respect to a standard text retrieval baseline.

Studies on the blogosphere are often dedicated to extracting social behaviour. This can be done by using link analysis methods, since authors tend to cite, explicitly or implicitly, other bloggers or articles. [6] creates a graph in which authors are linked to their posts and in which also people who have commented on a post are linked to it. [6] presents an algorithm similar to a random walk, since it iteratively obtains post scores by combining the authors' "authority" and "hub" scores. In our link analysis method instead the only vertices are blogs and blog posts, while the method of [6] also exploits the blogosphere structure, using authors as additional vertices and comments as additional hyperlinks. Additionally, [6] performs no analysis on the sentiment associated to hyperlinks.

Sentiment-sensitive link analysis. Link analysis of a blog network, with sentiment associated to links, is performed by [8] in order to examine trust propagation. The algorithm extracts sentiment from link contexts, then it uses an iterative algorithm and a trust matrix similar to a transition matrix. Trust is propagated according to the concepts of "direct propagation", "co-citation", "transpose trust", and "trust coupling"; a belief matrix is finally computed which represents relationships of trust between bloggers. [8] tests this approach on a network of political blogs, where trust is used to detect, given two political factions, "like-minded" blogs. While both the approach of [8] and our approach

are based on sentiment-sensitive link analysis, the goals are different, since [8] attempts to partition a set of blogs according to political orientation, while we attempt to improve the accuracy of blog retrieval.

Also the system of [19] uses link polarity in a retrieval application of text documents, with approval/disapproval information. Opinion relations between documents and citations are extracted from the contexts of the citations, using a lexical resource and a syntactic parser, in order to determine opinion polarity of the relations. We have not used syntactic analysis, and we have instead used the distance between the opinionated terms and the anchor text of the link. A user of the system described in [19] can access documents through an interface and can restrict search results to documents cited with a given polarity by other documents; sentiment analysis is thus a tool offered to the final user, and does not affect the retrieval methods and the accuracy.

It should also be noted that the sentiment analysis techniques used in both [8] and [19] are less sophisticated than the ones we use.

References

[1] James Allan, Ben Carterette, and Joshua Lewis. When will information retrieval be "good enough"? In *Proceedings of the 28th ACM International Conference on Research and Development in Information Retrieval (SIGIR'05)*, pages 433–440, Salvador, BR, 2005.

[2] Shlomo Argamon, Kenneth Bloom, Andrea Esuli, and Fabrizio Sebastiani. Automatically determining attitude type and force for sentiment analysis. In Hans Uszkoreit and Zygmunt Vetulani, editors, *Human Language Technology: Challenges of the Information Society (Revised selected papers from the 3rd Language Technology Conference)*, pages 218–231. Springer Verlag, Heidelberg, DE, 2009.

[3] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the 7th Conference on Language Resources and Evaluation (LREC'10)*, Valletta, MT, 2010.

[4] Allan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal, and Panayiotis Tsaparas. Link analysis ranking: Algorithms, theory, and experiments. *ACM Transactions on Internet Technology*, 5(1):231–297, 2005.

[5] Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

[6] Ko Fujimura, Takafumi Inoue, and Masayuki Sugisaki. The EigenRumor algorithm for ranking blogs. In *Proceedings of the WWW'05 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, Chiba, JP, 2005.

[7] Marti A. Hearst, Matthew Hurst, and Susan T. Dumais. What should blog search look like? In *Proceeding of the 2008 ACM Workshop on Search in Social media (SSM'08)*, pages 95–98, Napa Valley, US, 2008.

[8] Anubhav Kale, Amit Karandikar, Pranam Kolari, Akshay Java, Tim Finin, and Anupam Joshi. Modeling trust and influence in the blogosphere using link polarity. In *Proceedings of the 1st International Conference on Weblogs and Social Media (ICWSM'07)*, Boulder, US, 2007.

[9] Mostafa Keikha, Mark Carman, Robert Gwadera, Shima Gerani, Ilya Markov, Giacomo Inches, Az A. Alidin, and Fabio Crestani. University of Lugano at the TREC 2009 Blog Track. In *Proceedings of the 18th Text Retrieval Conference (TREC'09)*, Gaithersburg, US, 2009.

[10] Mostafa Keikha, Mark J. Carman, and Fabio Crestani. Blog distillation using random walks. In *Proceedings of the 32nd ACM Conference on Research and Development in Information Retrieval (SIGIR'09)*, pages 638–639, Boston, US, 2009.

[11] Chris P. Lee, Gene H. Golub, and Stefanos A. Zenios. A two-stage algorithm for computing PageRank and multistage generalizations. *Internet Mathematics*, 4(4):299–327, 2007.

[12] Elisabeth Lex, Michael Granitzer, and Andreas Juffinger. Facet classification of blogs: Know-Center at the TREC 2009 Blog Distillation Task. In *Proceedings of the 18th Text Retrieval Conference (TREC'09)*, Gaithersburg, US, 2009.

[13] Si Li, Huiji Gao, Hao Sun, Fei Chen, Oupeng Feng, and Sanyuan Gao. A study of faceted blog distillation - PRIS at the TREC 2009 Blog Track. In *Proceedings of the 18th Text Retrieval Conference (TREC'09)*, Gaithersburg, US, 2009.

[14] Craig Macdonald, Iadh Ounis, and Ian Soboroff. Overview of the TREC 2009 Blog Track. In *Proceedings of the 18th Text Retrieval Conference (TREC'09)*, Gaithersburg, US, 2009.

[15] Craig Macdonald, Rodrygo L. Santos, Iadh Ounis, and Ian Soboroff. Blog Track research at TREC. *SIGIR Forum*, 44(1):58–75, 2010.

[16] Richard McCreadie, Craig Macdonald, Iadh Ounis, Jie Peng, and Rodrygo L. Santos. University of Glasgow at TREC 2009: Experiments with Terrier. In *Proceedings of the 18th Text Retrieval Conference (TREC'09)*, Gaithersburg, US, 2009.

[17] Gilad Mishne. Multiple ranking strategies for opinion retrieval in blogs. In *Proceedings of the 15th Text Retrieval Conference (TREC'06)*, Gaithersburg, US, 2006.

[18] Iadh Ounis, Gianbattista Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Christina Lioma. Terrier: A high performance and scalable information retrieval platform. In *Proceedings of the SIGIR'06 Workshop on Open Source Information Retrieval (OSIR'06)*, Seattle, US, 2006.

[19] Scott S. Piao, Sophia Ananiadou, Yoshimasa Tsuruoka, Yutaka Sasaki, and John McNaught. Mining opinion polarity relations of citations. In *Proceedings of the 7th International Workshop on Computational Semantics (IWCS'07)*, pages 366–371, Tilburg, NL, 2007.

[20] Cristina Ribeiro and Gabriel David. FEUP at the TREC 2009 Blog Track: Temporal evidence in the Faceted Blog Distillation Task. In *Proceedings of the 18th Text Retrieval Conference (TREC'09)*, Gaithersburg, US, 2009.

[21] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Proceedings of the 6th International Conference on Data Mining (ICDM'06)*, pages 613–622, Washington, US, 2006.

[22] Wouter Weerkamp, Manos Tsagkias, and Maarten de Rijke. From blogs to news: Identifying hot topics in the blogosphere. In *Proceedings of the 18th Text REtrieval Conference (TREC'09)*, Gaithersburg, US, 2009.

[23] Xueke Xu, Yue Liu, Hongbo Xu, Xiaoming Yu, Linhai Song, Feng Guan, and Zeying Peng. ICTNET at Blog Track TREC 2009. In *Proceedings of the 18th Text REtrieval Conference (TREC'09)*, Gaithersburg, US, 2009.