Information Sciences 249 (2013) 38–47

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Endorsements and rebuttals in blog distillation ☆

Giacomo Berardi, Andrea Esuli, Fabrizio Sebastiani *, Fabrizio Silvestri

Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, 56124 Pisa, Italy

ARTICLE INFO

Article history:
Received 26 April 2012
Received in revised form 10 May 2013
Accepted 30 May 2013
Available online 14 June 2013

Keywords:
Blog distillation
Blog search
Link analysis
Sentiment analysis

ABSTRACT

In this paper we test 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 check whether a given hyperlink is a citation with a positive or a negative nature, i.e., if it expresses approval or disapproval of the hyperlinked page by the hyperlinking page. This allows us to test the hypothesis that distinguishing approval from disapproval brings about benefits in the blog distillation task.

We have tested our method on the Blogs08 collection used in the last two editions (2009 and 2010) of the TREC Blog Track, a collection consisting of more than one million blogs and more than 28 million blog posts. Unfortunately, the experimental results seem to disconfirm the above hypothesis, due to the low level of connectivity of the collection which severely limits the impact of a link analysis phase (and, *a fortiori*, of the attempt to distinguish endorsements from rebuttals). Application contexts other than the blogosphere (such as, e.g., the domain of eBay transactions) are probably more suited to such an approach.

1. Introduction

Blog distillation is a subtask of blog search. It is commonly 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 and principal topic of interest. Blog distillation has been intensively investigated within the TREC Blog Track [24,25], 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. Note that blog distillation deals with the ranking and retrieval of entire blogs, and not of individual blog posts; this sets it apart from other tasks investigated within the TREC Blog Track, which have individual posts as the main units of interest. As observed in [25], since blogs can be seen as collections of posts, blog distillation is somehow akin to resource selection in distributed information retrieval [33], whose goal is to identify the resources (e.g., document collections) which are more likely to contain documents relevant to a query.

☆ This is an extended version of a short paper originally presented at NLDB 2012 [4]. The order in which the authors are listed is purely alphabetical; each author has given an equally valuable contribution to the present work.

* Corresponding author. Tel.: +39 050 3152892.

E-mail addresses: giacomo.berardi@isti.cnr.it (G. Berardi), andrea.esuli@isti.cnr.it (A. Esuli), fabrizio.sebastiani@isti.cnr.it (F. Sebastiani), fabrizio.silvestri@isti.cnr.it (F. Silvestri).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.05.037

In this paper we test a method for blog distillation in which, on top of a standard system for retrieval by topicality, we add a link analysis phase, which is meant to account for the reputation/popularity of the blog. Link analysis has been extremely popular in Web search in the late '90 s and early 2000 s [5], also due to the success of the PageRank link analysis algorithm and the Google search engine, which had PageRank at the center of its ranking strategy. However, link analysis has witnessed a somehow decreased interest in the late 2000s, mainly due to the fact that the *hyperlink-as-endorsement hypothesis* (i.e., the hypothesis that the presence of a hyperlink denotes an endorsement of the hyperlinked page on the part of the hyperlinking page) is less and less justified in the Web at large [8,9,12], also 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 hyperlink-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, it is a matter of fact that many hyperlinks express a *rebuttal* (or "negative endorsement"), and not an approval, of the hyperlinked post on the part of the hyperlinking post. This might generate problems for the classic link analysis algorithms of Web search [5,6,22], since they are generally based on the hypothesis that hyperlinks represent (positive) endorsements, and not rebuttals; taking all hyperlinks as expressing undifferentiated endorsement might actually lead to suboptimal results.

In this work we test the hypothesis that distinguishing hyperlinks expressing approval from hyperlinks expressing rebuttal may be beneficial to blog distillation. In order to do so, we define a *sentiment-sensitive link analysis* method, i.e., a random-walk method on which the two types of hyperlinks have a different impact. We try to detect the sentimental valence (or "polarity") of a given hyperlink (i.e., to establish whether the hyperlink conveys a positive or a negative endorsement) by performing sentiment analysis on a text window around the hyperlink.

We have tested our method on the Blogs08 collection used in the last two editions (2009 and 2010) of the TREC Blog Track,[1] a collection consisting of more than one million blogs and more than 28 million blog posts. Unfortunately, the experimental results seem to disconfirm the above hypothesis, due to the low level of connectivity of the collection which severely limits the impact of a link analysis phase (and, *a fortiori*, of the attempt to distinguish endorsements from rebuttals).

This paper is organized as follows. In Section 2 we present the approach to blog distillation based on sentiment-sensitive link analysis, as sketched above. Section 3 presents and discusses the results of experiments we have performed on the Blogs08 collection used in the last editions (2009 and 2010) of the TREC Blog Track. Section 4 presents related work, while Section 5 concludes, sketching some avenues for further research.[2]

2. Sentiment-sensitive link analysis for blog ranking

In this section we discuss the model we have adopted to compute the sentiment-sensitive, link-analysis-based ranking of blogs and blog posts. We rely on a graph-based model in which nodes represent either blogs or blog posts, and the weights attached to nodes represent their importance.

2.1. A graph-based model of the blogosphere

Throughout the paper we will assume the existence of a set $\mathcal{P}$ of blog posts, partitioned into a set of blogs $\mathcal{B}$. That is, a blog is viewed just as a *set* of blog posts, and the order of the posts within a blog is disregarded.

Let $G_{\mathcal{P}} = (V_{\mathcal{P}}, E_{\mathcal{P}})$ be a graph, where each node in the set $V_{\mathcal{P}} = \{p_1, p_2, \ldots, p_n\}$ represents a post in $\mathcal{P}$, and where each edge in $E_{\mathcal{P}}$ represents a hyperlink between two posts in $\mathcal{P}$ belonging to different blogs; i.e., edge e_{xy} from node p_x to node p_y denotes the presence of at least one hyperlink from the post represented by p_x to the post represented by p_y, where p_x and p_y belong to two different blogs.[3] Similarly, let $G_{\mathcal{B}} = (V_{\mathcal{B}}, E_{\mathcal{B}})$ be a graph, where each node in the set $V_{\mathcal{B}} = \{b_1, b_2, \ldots, b_m\}$ represents a blog in $\mathcal{B}$, and where $E_{\mathcal{B}}$ is a set of edges corresponding to hyperlinks between blogs, i.e., edge $e_{xy} \in E_{\mathcal{B}}$ from node b_x to node b_y denotes the presence of at least one hyperlink from a post $p_z \in b_x$ to the homepage of blog b_y.

Let $w_{\mathcal{P}} : E_{\mathcal{P}} \to \mathbb{R}$ and $w_{\mathcal{B}} : E_{\mathcal{B}} \to \mathbb{R}$ be two weighting functions, where $\mathbb{R}$ denotes the set of the reals. Informally, the weight assigned to an edge captures the importance the corresponding hyperlink confers onto the hyperlinked post (for $E_{\mathcal{P}}$) or blog (for $E_{\mathcal{B}}$). These weighting functions, which we have defined using sentiment analysis techniques, are described in detail in the next section.

2.2. Weighting the hyperlinks via sentiment analysis

We define the weighting functions $w_{\mathcal{P}}$ and $w_{\mathcal{B}}$ on the basis of a sentiment-based analysis, whose aim is to determine if the hyperlink represented by edge e_{xy} denotes a positive or a negative attitude of post p_x towards post p_y (for $w_{\mathcal{P}}$), or of blog b_x

[1] The TREC Blog Track has not taken place since 2011.

[2] The present work is an extension of a short paper published as [4], and contains a much more detailed description of the algorithm we use and a much more extensive analysis of the dataset and of the experimental results, and also presents for the first time additional experiments discussed at the end of Section 3.2.

[3] We disregard hyperlinks between posts belonging to the same blog because our units of interest are blogs, not posts; that is, as observed in the introduction, blog distillation is about ranking entire blogs. As a consequence, hyperlinks between posts belonging to the same blog are, for our purposes, self-referential, and it does not make sense to let them have an impact on the blog scoring process.

towards blog b_y (for $w_{\mathcal{B}}$). A positive (resp., negative) value of $w_{\mathcal{P}}(e_{xy})$ will indicate a positive (resp., negative) attitude of post p_x towards post p_y, and the absolute value of $w_{\mathcal{P}}(e_{xy})$ will indicate the intensity of this attitude. Similarly, a positive (resp., negative) value of $w_{\mathcal{B}}(e_{xy})$ will indicate an altogether positive (resp., negative) attitude towards blog b_y of the posts in blog b_x that do link to the homepage of b_y, and the absolute value of $w_{\mathcal{B}}(e_{xy})$ will indicate the intensity of this attitude.

For determining $w_{\mathcal{P}}(e_{xy})$ all the hyperlinks from p_x to p_y are taken into account; similarly, for determining $w_{\mathcal{B}}(e_{xy})$ all the hyperlinks from any post $p_z \in b_x$ to the homepage of blog b_y are taken into account. For determining the impact of a hyperlink on the weighting function, the anchor text and the sentence in which the anchor text is embedded are analysed, in search of sentiment-carrying expressions that determine the overall sentiment of the sentence. This analysis begins with POS-tagging the sentence in order to identify multiwords matching the `(RB|JJ)+` and `NN+` patterns, and that we call *candidate sentiment chunks*.[4]

In order to determine if these candidate chunks are indeed sentiment-laden, we first assign a sentiment score to each term in the chunk by using SentiWordNet [3], an extension of WordNet [11], as the source of sentiment scores. From SentiWordNet we have created a word-level dictionary (that we call SentiWordNet$_W$) in which each POS-tagged word w (rather than each word sense $s(w)$, as in full-fledged SentiWordNet) is associated with a score $\sigma(w)$ that indicates its sentimental valence, averaged across its word senses $s_1(w),\ldots,s_{n(w)}(w)$. We have heuristically obtained this score by computing a weighted average

$$\sigma(w) = \frac{1}{Z_w}\sum_{i=1}^{n(w)}\frac{1}{i}(Pos(s_i(w)) - Neg(s_i(w))) \tag{1}$$

of the differences between the positivity and negativity scores assigned to the various senses of w, where $Z_w = \sum_{i=1}^{n(w)}\frac{1}{i}$ is a normalization factor. In this weighted average the weight is the inverse $\frac{1}{i}$ of the sense index i. This is due to the fact that, in WordNet (and in SentiWordNet too), the senses $s_1(w), \ldots, s_{n(w)}(w)$ of a given word w are sorted from most to least frequently used; weighting by $\frac{1}{i}$ thus lends more prominence to the senses of w that are most frequently used in language.

All candidate sentiment chunks only composed by terms that have been assigned a sentiment score equal to 0 (i.e., sentiment-neutral words) are discarded from consideration. Each remaining *sentiment chunk* is then checked for the presence of *valence shifters*, i.e., negators (e.g., "no", "not") or intensifiers/downtoners (e.g., "very", "strongly", "barely", "hardly"); we have used the valence shifters of the "appraisal lexicon" of [1], a sentiment lexicon for English built according to the principles of "appraisal theory" [26]. When a valence shifter is found, the score of the word that follows is modified accordingly (e.g., "very good" is assigned a doubly positive score than "good"). We then assign a sentiment score to a sentiment chunk by heuristically summing the sentiment scores of all the words in the chunk, also taking into account the modifications resulting from the application of the valence shifters.

In order to determine a sentiment score for the hyperlink, we compute a weighted sum of the sentiment scores of all the chunks that appear in the sentence containing the anchor text. The weights in this weighted sum are computed as a decreasing function of the distance d of the chunk from the anchor text; this implements the intuition that the closer a chunk is to the hyperlink, the more it is related to it. We compute the weight γ via the function $\gamma(d) = 1/(0.3d + 1)$. The distance d (measured in number of words between the anchor text and the chunk) is itself computed as a weighted sum, where each such token has its own weight depending on its type. For example, mood-changing particles such as "instead" are assigned a higher weight, while prepositions have a weight of zero. The following is a fully worked-out example of how the sentiment score of a candidate sentiment chunk contributes to the sentiment score of a hyperlink.

Example 1. Assume that the linking post contains the sentence "womens fashion trends, which are set to be very popular", and that the expression "womens fashion" is the anchor text of the hyperlink (this is indeed a real example from the Blogs08 collection that we have used in the experiments of Section 3).

POS tagging returns the following result: ('womens', NNS), ('fashion', NN), ('trends', NNS), ('which', WDT), ('are', VBP), ('set', VBN), ('to', TO), ('be', VB), ('very', RB), ('popular', JJ).

The expression "very popular" is recognized as a candidate sentiment chunk since it matches pattern (RB|JJ)+, and its sentiment score must then be computed. SentiWordNet$_W$ returns the value 0.125 for "popular", while "very" is a valence shifter to which a 1.5 multiplicative factor is assigned; thus, "very popular" is assigned a sentiment score of $0.125 \cdot 1.5 = 0.1875$.

Between the anchor text "womens fashion" and "very popular" lie the words "trends, which are set to be"; all these words except "trends" are stop words and are thus given zero weight, so the distance between "womens fashion" and "very popular" is taken to be 1. Therefore, $\gamma(1) = 1/(0.3 + 1) = 0.7692$. This means that "very popular" contributes a sentiment score of $0.1875 \cdot 0.7692 = 0.1442$ to the sentiment score of the hyperlink anchored at "womens fashion". □

[4] This use of the term "chunk" is consistent with the use of this term as in the natural language processing literature, which sees chunks as "flat non-overlapping segments of a sentence that constitute the basic non-recursive phrases corresponding to the major parts-of-speech found in most wide-coverage grammars" [18, p. 451]. Indeed, the patterns `(RB|JJ)+` and `NN+` identify flat, non-overlapping, non-recursive segments of a sentence that we deem likely to be sentiment-bearing. For POS tagging we have used the Natural Language Toolkit available at http://www.nltk.org.

Finally, we compute $w_{\mathcal{P}}(e_{xy})$ as the average of the sentiment scores assigned to all the hyperlinks from p_x to p_y; if both positive and negative hyperlinks are present, they compensate each other. Similarly, we compute $w_{\mathcal{B}}(e_{xy})$ as the average of the sentiment scores assigned to all the hyperlinks from posts in b_x to the homepage of blog b_y.

Using the $w_{\mathcal{P}}$ function we then split $G_{\mathcal{P}}$ into two graphs $G^+_{\mathcal{P}} = (V_{\mathcal{P}}, E^+_{\mathcal{P}})$ and $G^-_{\mathcal{P}} = (V_{\mathcal{P}}, E^-_{\mathcal{P}})$, where $E^+_{\mathcal{P}}$ and $E^-_{\mathcal{P}}$ are the sets of edges e_{xy} such that $w_{\mathcal{P}}(e_{xy}) \geqslant 0$ and $w_{\mathcal{P}}(e_{xy}) < 0$, respectively. Analogously, we split $G_{\mathcal{B}}$ into $G^+_{\mathcal{B}} = (V_{\mathcal{B}}, E^+_{\mathcal{B}})$ and $G^-_{\mathcal{B}} = (V_{\mathcal{B}}, E^-_{\mathcal{B}})$, where $E^+_{\mathcal{B}}$ and $E^-_{\mathcal{B}}$ are the sets of edges e_{xy} such that $w_{\mathcal{B}}(e_{xy}) \geqslant 0$ and $w_{\mathcal{B}}(e_{xy}) < 0$, respectively. The rationale of dividing the graphs into subgraphs containing positive or negative edges only will be apparent in the next sections, and has to do with the fact that we want to treat the two kinds of links differently. Positive links are of the navigational type, in the sense that the author invites the reader to follow them and navigate the linked contents. Negative links can instead be interpreted as simply the author's attempts to bestow a negative light on the cited page.

2.3. Ranking the nodes

We use the graphs $G^+_{\mathcal{P}}, G^-_{\mathcal{P}}, G^+_{\mathcal{B}}, G^-_{\mathcal{B}}$ in order to compute the ranking of posts and blogs based on sentiment-sensitive link analysis. We use an algorithm known as *random walk with restart* (RWR – [34]), also known as *personalized* (or *topic-sensitive*) *random walk* [15]. This algorithm differs from more standard random walk algorithms such as PageRank for the fact that the $\mathbf{v}_{\mathcal{P}}$ and $\mathbf{v}_{\mathcal{B}}$ vectors of Eq. (2) (see below) are not uniform. The values in the latter vectors are sometimes referred to as the *restart probabilities*. Two RWR computations are run on $G^+_{\mathcal{P}}$ and $G^+_{\mathcal{B}}$, respectively, yielding $\mathbf{r}_{\mathcal{P}}$ and $\mathbf{r}_{\mathcal{B}}$ (i.e., the vectors of scores for posts and blogs) as the principal eigenvectors of the matrices

$$\begin{aligned} \mathbf{P} &= (1-d)\cdot \mathbf{A}^+_{\mathcal{P}} + \frac{d}{k}\cdot \mathbf{v}_{\mathcal{P}} \\ \mathbf{B} &= (1-d)\cdot \mathbf{A}^+_{\mathcal{B}} + \frac{d}{k'}\cdot \mathbf{v}_{\mathcal{B}} \end{aligned} \tag{2}$$

where $\mathbf{A}^+_{\mathcal{P}}$ (resp., $\mathbf{A}^+_{\mathcal{B}}$) is the adjacency matrix associated with graph $G^+_{\mathcal{P}}$ (resp., $G^+_{\mathcal{B}}$), and d is the damping factor of the random walk (i.e., the factor that determines how much backlinks influence random walks).

In order to explain what $k, k', \mathbf{v}_{\mathcal{P}}$, and $\mathbf{v}_{\mathcal{B}}$ are, we first need to describe how the first step of our blog distillation method will look like (the method itself will be thoroughly discussed in Section 2.4). Let q be a query whose aim is to rank the blogs in descending order of relevance to the query topic. In the first step of our blog distillation method, a standard (i.e., text-based) retrieval engine is run on $\mathcal{P}$, yielding a ranked list of the k top-scoring posts for q. Let $L = (l_1, l_2, \ldots, l_k)$ be this ranked list (with l_1 the top-scoring element), and let $(s_1, s_2, \ldots, s_k)$ be the list of the corresponding scores returned by the retrieval engine. We may now go back to explaining Eq. (2). Vector $\mathbf{v}_{\mathcal{P}}$ is the preference vector for blog posts, i.e., a vector whose entries corresponding to the k pages in L are set to 1 and whose other entries are set to 0. Vector $\mathbf{v}_{\mathcal{B}}$ is obtained in a slightly different way. We first group together posts belonging to the same blog and build a vector $\overline{\mathbf{v}}_{\mathcal{B}}$ whose entries count the number of retrieved posts belonging to the blog corresponding to the entry. Vector $\overline{\mathbf{v}}_{\mathcal{B}}$ is then normalized into vector $\mathbf{v}_{\mathcal{P}}$ and, in this case, k' is the number of entries in $\mathbf{v}_{\mathcal{B}}$ greater than zero.

In order to find the principal eigenvectors of $\mathbf{P}$ and $\mathbf{B}$ we solve the following eigenproblems:

$$\begin{aligned} \mathbf{r}^+_{\mathcal{P}} &= \mathcal{P}\cdot \mathbf{r}^+_{\mathcal{P}} = (1-d)\cdot \mathbf{A}^+_{\mathcal{P}}\cdot \mathbf{r}_{\mathcal{P}} + \frac{d}{k}\cdot \mathbf{v}_{\mathcal{P}} \\ \mathbf{r}^+_{\mathcal{B}} &= \mathcal{B}\cdot \mathbf{r}^+_{\mathcal{B}} = (1-d)\cdot \mathbf{A}^+_{\mathcal{B}}\cdot \mathbf{r}_{\mathcal{B}} + \frac{d}{k'}\cdot \mathbf{v}_{\mathcal{B}} \end{aligned} \tag{3}$$

We calculate the vectors of negative scores $\mathbf{r}^-_{\mathcal{P}} = \mathbf{A}^-_{\mathcal{P}}\cdot \mathbf{r}^+_{\mathcal{P}}$ and $\mathbf{r}^-_{\mathcal{B}} = \mathbf{A}^-_{\mathcal{B}}\cdot \mathbf{r}^+_{\mathcal{B}}$, where $\mathbf{A}^-_{\mathcal{P}}$ (resp., $\mathbf{A}^-_{\mathcal{B}}$) is the adjacency matrix associated with graph $G^-_{\mathcal{P}}$ (resp., $G^-_{\mathcal{B}}$) and we normalize them so that the sum of their components is-1. $\mathbf{A}^-_{\mathcal{P}}$ and $\mathbf{A}^-_{\mathcal{B}}$ contain the negative values associated with the edge weights of $G^-_{\mathcal{P}}$ and $G^-_{\mathcal{B}}$. Finally, the scoring vectors $\mathbf{r}_{\mathcal{P}}$ and $\mathbf{r}_{\mathcal{B}}$ that result from taking into account both positive and negative links are given by

$$\begin{aligned} \mathbf{r}_{\mathcal{P}} &= (1-\theta)\cdot \mathbf{r}^+_{\mathcal{P}} + \theta\cdot \mathbf{r}^-_{\mathcal{P}} \\ \mathbf{r}_{\mathcal{B}} &= (1-\theta)\cdot \mathbf{r}^+_{\mathcal{B}} + \theta\cdot \mathbf{r}^-_{\mathcal{B}} \end{aligned} \tag{4}$$

where θ is a parameter that allows tuning the relative impact of positive and negative links on the overall score of a post (or blog).

2.4. Scoring blogs against a query

We are now in a position to describe our blog distillation method. As anticipated in the previous section, let q be a query whose aim is to rank the blogs in descending order of relevance to the query topic. Our method is articulated in three steps:

1. A standard (i.e., text-based) retrieval engine is run on $\mathcal{P}$, yielding a ranked list $L = (l_1, l_2, \ldots, l_k)$ of the k top-scoring posts for q, with l_1 the top-scoring element and with $(s_1, s_2, \ldots, s_k)$ the corresponding scores returned by the retrieval engine.
2. The scores s_x returned by the retrieval engine are combined with the link-based scores. For this we use the combination rule

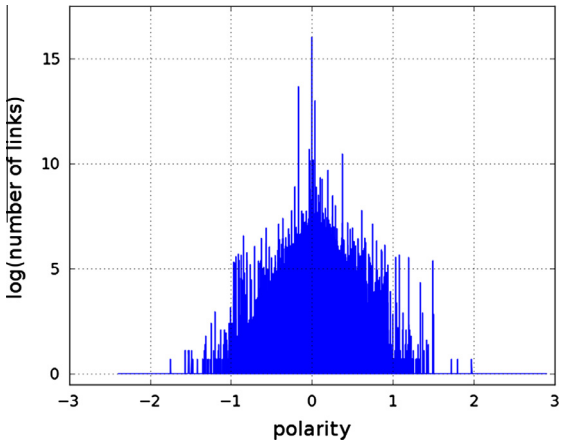

Fig. 1. Distribution of the polarity scores for the edges in the blog posts graph.

$$w_x = (1-\alpha)\cdot s_x + \alpha \cdot \mathbf{r}_{\mathcal{P}x} \tag{5}$$

for all $x \in \{1,\ldots,k\}$, where $\mathbf{r}_{\mathcal{P}x}$ is the x-th component of the vector $\mathbf{r}_{\mathcal{P}}$ of post scores as computed via Eq. (4), and α is a parameter which allows to tune the relative impact of the text-based and the link-based scores.

3. The scores computed for each post are merged according to the blog the post itself comes from.[5] Obviously the choice of the merging function is an important issue. We have adopted the very simple approach of scoring each blog b_x with a weighted sum of (i) the average of the scores w_z for each post $p_z \in b_x$, and (ii) the static score $\mathbf{r}_{\mathcal{B}x}$ of blog b_x smoothed using the actual number of posts retrieved for blog b_x. We use the same α coefficient as above as the coefficient of the weighted sum. So, we score blog b_x by means of the equation

$$\omega_x = (1-\alpha)\frac{\sum_{p_z \in b_x} w_z}{|b_x|} + \alpha \cdot \mathbf{r}_{\mathcal{B}x}\frac{|L \cap b_z|}{|b_z|} \tag{6}$$

where by $|L \cap b_z|$ we denote the number of posts of blog b_z retrieved as top-k posts in the list L.

Eventually, our blog retrieval system ranks the blogs according to the scores computed by Eq. (6).

3. Experiments

3.1. Experimental setting

We have tested our method on the Blogs08 collection used in the 2009 and 2010 editions of the TREC Blog Track [25]. Blogs08 consists of a crawl of 1,303,520 blogs, crawled from 14 January 2008 to 10 February 2009. The crawl resulted in a total of 28,488,766 blog posts, each identified by a unique URL. Each blog post has been effectively downloaded 2 weeks after its first identification from the crawler, in order to include also a number of comments to the post by 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 [24].

The graph of blog posts contains 4,697,700 nodes (only posts with outlinks and inlinks are considered) and 12,633,788 edges.

We have processed all the hyperlinks via our hyperlink polarity scoring method (see Section 2.2). The processing of hyperlinks relative to the graph of blog posts resulted in the identification of 8,947,325 neutral hyperlinks (i.e., polarity weight equal to zero), 1,906,182 positive hyperlinks, and 1,780,281 negative hyperlinks. Fig. 1 shows the distribution of polarity scores for the edges in the graph of the blog posts.

We have evaluated the results of our experiments via the evaluation measures that are typically adopted in the TREC Blog Track, i.e., the well-known *precision at 10* (P@10), *mean average precision* (MAP), and *binary preference* (BPref):

[5] For this step we had originally considered a time-based merging function, i.e., one that gives higher weight to more recent posts. We eventually rejected this option because, as pointed out in the introduction, the topic expressed by the query q must be *recurring*, i.e., it must underlie the entire temporal stream of posts that makes up the blog; we are not interested in blogs whose *recent* posts are about the topic, we are interested in blogs that tend to be about the topic throughout their entire lifespan. So, we think that making recency a factor would be counterintuitive. Additionally, weighting posts according to recency would introduce further parameters that could excessively increase the complexity of the model.

- *Precision at k* (noted *prec*(*k*), or *P@k*), is the fraction of the documents to be found down to rank r_k that are judged relevant, i.e.,

$$prec(k) = \frac{1}{k} \sum_{1 \leqslant i \leqslant k} r_i \tag{7}$$

P@10 is a very widely used measure, which emphasizes the behavior of a ranked retrieval system on the first 10 positions; this is meant to reflect the fact that, in most search engines, the first page of displayed search results exactly contains 10 results, and that most users hardly ever look at the second page.

- MAP is defined as

$$MAP(Q) = \frac{1}{|Q|} \sum_{q \in Q} AP(q)$$

i.e., the mean value of *average precision* (AP) across all involved queries. $AP(q)$ is defined in turn as the average precision (see below) obtained for query q across all the ranks at which a document judged relevant is found, i.e.,

$$AP(q) = \frac{1}{|R|} \sum_{1 \leqslant k \leqslant |C|} r_k \cdot prec(k)$$

where R is the set of documents judged relevant with respect to the query, C is the entire collection, r_i evaluates to 1 if the document in position i in the results is relevant with respect to the query and to 0 otherwise. MAP is probably the best-known and most widely used measure for evaluating ranked retrieval systems.

- The BPref measure [7] is defined as

$$BPref = \frac{1}{|R|} \sum_{r} \sum_{n} \left(1 - \frac{|r \prec n|}{min(|R|, |N|)}\right) \tag{8}$$

where N is the set of documents judged non-relevant, r is any document judged relevant, n is any among the $|R|$ top-ranked documents judged nonrelevant, and $r \prec n$ denotes the fact that r precedes n in the ranking. BPref is a widely used measure for measuring ranked retrieval system on test collections in which there is a large number of unjudged documents, i.e., documents whose relevance to the query has not been assessed.

The highest possible value for all these measures is 1.

As the system for performing the first step of our method we have used Terrier [28]. With Terrier we have built a baseline for the comparison of our results, based on ranking the documents with respect to each query via the well-known BM25 weighting method [32]. These rankings have been used also as the input to the random-walk-based reranking phase of our method. For each query we retrieve the first 100 posts, which are set as the nodes with non-null restart probabilities given as input to the RWR method of Section 2.3. For the damping factor d (see Eq. (3)) we have used the value 0.85, a fairly typical value in these methods [5]; we leave the optimization of this parameter to future work. For the parameter θ that sets the relative importance of positive and negative links on the overall score of a post (see Section 2.3) we have used the value $\theta = 0.5$; in preliminary experiments we had tested all values of θ in the set $\{0.1, 0.2, \ldots, 0.8, 0.9\}$, and 0.5 had proven the best value, although the differences between different values had proven inessential.

We solve the two eigenproblems of Eq. (3) via the well-known "power method" [14], which consists in iteratively refining the two vectors $\mathbf{r}_{\mathcal{P}^+}$ and $\mathbf{r}_{\mathcal{B}^+}$ until convergence. For each of the two eigenproblems, convergence is taken to mean that the difference between the norms of the vectors at two successive iterations is smaller than a constant ϵ. In our experiments we set $\epsilon = 10^{-9}$.

We have tested various values for parameters α (see Eqs. (5) and (6)) and k (the number of posts retrieved in the 1st step of our method). For α we have tested the values 0.65, 0.75, 0.80, 0.85; preliminary experiments with values outside this range had shown clear deteriorations in effectiveness. For k we have tested the values 1000, 2000, 3000, 4000; here too, preliminary experiments with values outside this range had shown clear deteriorations.

3.2. Results

The different values for the α parameter do not yield substantial differences in performance. This can be gleaned from Table 1 by looking at the differences among best and average (across the four tested values of α) values returned by our method (2nd and 3rd column of each block), which are always very small.

The results do show an improvement over the retrieval-by-topicality baseline, but this improvement is very small. A closer inspection of the results reveals that this is due to the sparsity of the graphs resulting from the collection of posts retrieved in the first step: these posts are often isolated nodes in the posts graph, which means that the random-walk step only affects a small subset of the results. Increasing the value of the k parameter determines an increase of the improvement over the baseline, since more posts are affected by the random walk scores, but the relative improvement is still small

Table 1
Ranking effectiveness on the Blogs08 collection. "Base" indicates the baseline; "$\alpha_{best} = r$" indicates a run of our method, where r is the value of α which has returned the best MAP result for the specific value of k; "Mean" indicates the average performance of our method across the four tested values of α. **Boldface** indicates the best result.

	$k = 1000$			$k = 2000$		
	Base	$\alpha_{best} = 0.85$	Mean	Base	$\alpha_{best} = 0.75$	Mean
MAP	0.1775	0.1802	0.1801	0.1914	0.1943	0.1942
P@10	0.2878	**0.2898**	0.2880	0.2796	0.2837	0.2819
BPref	0.2039	0.2056	0.2050	0.2203	0.2222	0.2220
	$k = 3000$			$k = 4000$		
	Base	$\alpha_{best} = 0.80$	Mean	Base	$\alpha_{best} = 0.65$	Mean
MAP	0.1958	**0.1986**	0.1983	0.1949	0.1977	0.1976
P@10	0.2653	0.2735	0.2719	0.2592	0.2673	0.2650
BPref	0.2226	**0.2247**	0.2242	0.2210	0.2230	0.2224

Table 2
Comparison of MAP results between random walk with restart (RWR) and PageRank (PR), with $k = 3000$. **Boldface** indicates best results for each weight.

α	0.65	0.75	0.80	0.85
RWR	**0.1981**	**0.1983**	**0.1986**	0.1983
PR	0.1978	**0.1983**	0.1985	**0.1988**

anyway. We have seen that increasing k beyond the value of 4000, instead, introduces a higher number of irrelevant posts in the results, which decreases the magnitude of the improvement.

We have also tried different values for θ (the linear combination coefficient in Eq. (4)), but this has not brought about any substantial improvement. The main reason is that, due to the above-mentioned sparsity of the posts graph, the impact of the link analysis phase on the final ranking is low anyway, regardless of how this link analysis balances the contribution of the positive and negative links.

In order to obtain a further confirmation of the fact that using link analysis does not impact substantively on the final ranking, we have run an experiment in which the RWR method has been replaced by standard PageRank. In this latter experiment, score vectors $\mathbf{r_P}$ and $\mathbf{r_B}$ are computed similarly to our method, but for the fact that the entire G_P and G_B graphs are used, i.e., without differentiating positive and negative links via sentiment analysis. Edges are weighted with uniform probability of transition equal to $1/outdegree(node)$. The differences in the final results are negligible, as we show in Table 2. The very slight improvement brought about by PageRank over our method is probably due to the fact that the two graphs used by PageRank have a larger proportion of non-isolated nodes than each of the graphs on which our method operates. Our algorithm, however, is faster than PageRank, because it works on the reduced transition matrices given by G_B^+ and G_P^+.

In sum, we can conclude that the hypothesis according to which it makes sense to distinguish positive from negative endorsements in blog analysis has neither been confirmed nor disconfirmed. To see this, note that the literature on blog search has unequivocally shown that the best results are obtained when sentiment analysis is performed not on the entire blogosphere, but on the subset of blogs which have been top-ranked by a standard retrieval-by-topicality engine [24,25,29]. The essential conclusion that can be drawn from our results is that the blogs (and their posts) retrieved in the retrieval-by-topicality phase contain too few hyperlinks/ endorsements, no matter whether positive or negative, for a link analysis phase to have a substantial impact on retrieval.

To support this conclusion we have run an additional set of experiments with the purpose of evaluating the connectivity of the dataset and the quality of its entries. First of all we have evaluated the connectivity of the posts in the collection. As already described, within the 28,488,766 posts only 4,697,700 (≈16.3%) of the posts have an incoming or outgoing link. The remaining posts are completely disconnected. Furthermore, considering the 1,303,520 blogs contained in the collection, only 634,313 (≈48.6%) have either an incoming or an outgoing link (i.e., links from a post of a blog b_i to another blog b_j); if we look for the percentage of nodes whose indegree is at least x, the figures become substantially smaller, even for modest values of X such as 1 or 2 (same for outdegree).

These facts can be inferred from Fig. 2, which plots the distributions of inlinks and outlinks for the graph of blogs. As expected, they exhibit a power-law behavior and, more importantly, the two curves are very steep. This means that the distribution of links is very skewed towards the total absence of connections. Furthermore, the most highly linked nodes (i.e., the "hubs") have a "generalist" nature, since they have a high number of outlinks (see Table 3, which lists the 10 most highly linked blogs.). In fact, blogs linked from these highly central blogs rarely point to other pages. This will cause the PageRank value transferred by the hubs to not being propagated any further than one page.

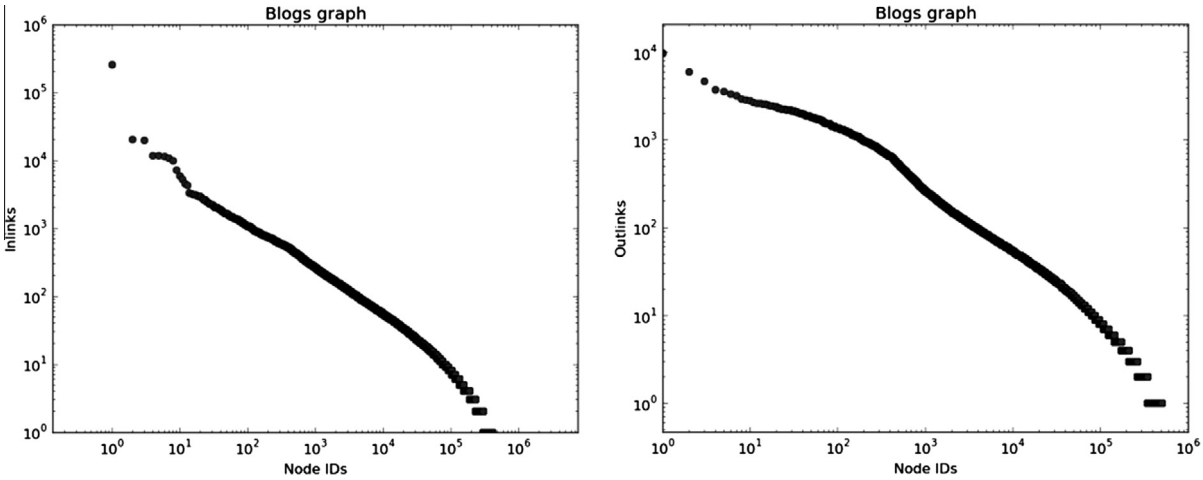

Fig. 2. Distribution of inlinks (left) and outlinks (right).

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 we have discussed at the beginning of Section 1. TREC Blog Track participants have different approaches to the blog distillation task: most of them do not pay attention to quality-and authority-related aspects of blogs, and mainly use a combination of (i) IR and statistical techniques, such as language models and/or query expansion [24,25], and sentiment analysis techniques [30]. However, quality-related aspects are important in the blog domain, since content is user-generated and the authoritativeness of authors is thus highly variable [16].

4.1. Link analysis in blog distillation

A random walk algorithm has been used in [20] 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 indicates membership of that post in that specific blog, and an edge between a post and a term means that the term occurs 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 hyperlinks, while in [20] the transition probabilities are uniform (except for the weights from posts to terms, which are calculated via the $tf{*}idf$ function).

In [27] different scoring strategies are used to rank blog posts. One strategy is to consider post authority by analysing hyperlink structure. [27] does not perform a random walk, but implements a different link-based notion via the use of "post in-degree". The results of [27] show that post in-degree does not improve accuracy.

Studies on the blogosphere are often dedicated to extracting social behavior. This can be done by using link analysis methods, since authors tend to cite, explicitly or implicitly, other bloggers or articles. Fujimura et al. [13] create a graph in which authors are hyperlinked to their posts, and in which also people who have commented on a post are hyperlinked to it. They present an algorithm similar to a random walk, since it iteratively obtains post scores by combining the authors' "authority" and "hub" scores. Instead, in our hyperlink analysis method the only vertices are blogs and blog posts, while the method of [13] also exploits the blogosphere structure, using authors as additional vertices and comments as additional hyperlinks. Additionally, [13] performs no analysis on the sentiment associated to hyperlinks.

4.2. Sentiment analysis in blog distillation.

A text retrieval system that exploits opinionated terms in query contexts is described in [2]. Documents are indexed with terms, and these latter are enriched with tags, one of which is the Opinionated tag. The user, upon submitting a query, may ask to retrieve only documents in which query terms occur close to opinionated (i.e., sentiment-laden) words. The results show that opinionated words have a positive effect also in identifying blogs topically relevant to a given query. As in our work, [2] uses sentiment analysis to improve accuracy in text retrieval; however, the key difference is that we analyse the sentiment-related properties of document inlinks and outlinks, while [2] analyses the sentiment-related properties of the terms occurring in the document.

Table 3
List of the top-10 most frequent blogs in the collection.

http://community.livejournal.com/lj_dev/
http://press.xanga.com
http://www.google.com
http://news.livejournal.com/
http://quizilla.com/
http://community.livejournal.com/lj_spotlight/
http://wordpress.com
http://www.delightfulblogs.com/
http://juicyfruiter.blogspot.com/
http://news.google.com/

4.3. Sentiment analysis of contexts

The problem of identifying and extracting the polarity of sentences has been widely studied in the sentiment analysis literature, and has been approached either via supervised or unsupervised methods [23]. While for the first approach an annotated dataset is necessary, the second approach is normally based on the use of sentiment-oriented lexical resources. The present paper adopts the unsupervised approach, since datasets annotated by opinion and consisting of content from the blogosphere are not available. Other instances of the unsupervised approach are in [10,17], where sentiment analysis at the sentence level is performed in order to extract the sentiment polarity of features of products from user reviews. Similarly to what we do, [10] uses a function that weights sentiment by the distance of the sentiment-carrying expression from the expression denoting the product feature. Another possible approach to identifying the relation between sentiment expressions and link could involve the creation of the dependency parse tree of the sentence, similarly to what is done in [21].

4.4. Personalized random walks

We have chosen to perform a personalized random walk, executed for each query, similarly to the algorithms surveyed in [15]. Many approaches to personalized random walks have been proposed, most of which concentrate on making personalization computationally feasible, often by exploiting data structures that allow to solve an approximation of the original problem. In the *fast random walk with restart* [34] the transition matrix of the random walk is divided in two matrices: one is partitioned according to graph sub-communities, while the other is reduced by means of low-rank approximation. Personalized ranking is then computed via a matrix product. We have instead used an iterative algorithm, since the approach of [34] would have been computationally too expensive on a graph of the size of ours.

4.5. Sentiment-sensitive link analysis

Link analysis of a blog network, with sentiment associated to hyperlinks, is performed in [19] in order to examine trust propagation. The algorithm extracts sentiment from hyperlink 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. The authors test 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 [19] and our approach are based on sentiment-sensitive hyperlink analysis, the goals are different, since [19] 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 [31] uses hyperlink 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 hyperlink. A user of the system described in [31] 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 [19,31] are less sophisticated than the ones we use.

5. Conclusions

We have presented a study on the impact of sentiment analysis on blog distillation. We have designed a graph-based technique that exploits opinions expressed in the sentence that contains a hyperlink. We have measured the retrieval performance of our techniques using a standard testbed, namely the Blogs08 collection used in the 2009 and 2010 editions of the TREC Blog Track.

The main conclusion that can be drawn from our experiments is that the blogs retrieved in the retrieval-by-topicality phase (an essential component of any blog distillation method) contain too few hyperlinks, no matter whether representing a positive or a negative endorsement, for a link analysis phase to be robust enough. This means that a method that differentiates positive and negative links may have an impact, if any, in a context in which the density of hyperlinks is much higher than in the dataset we have analysed. Whether such a context exists within Web retrieval, of course, is not yet clear; applications such as e-commerce (e.g., positive and negative feedback in eBay transactions) might be an interesting testbed for this idea. We think that, in the future, it might be promising to apply this technique in a context in which the hyperlinked documents are scientific papers. In this context the hyperlinks would stand for the citations to the papers listed in the bibliography. In this context too, it would make sense distinguishing positive endorsements (a paper that is cited approvingly, e.g., because the technique it proposes is deemed interesting) from negative endorsements (a paper that is cited disapprovingly, e.g., because the methodology it follows is considered weak). What makes this application context promising is that most nodes in the graph would (unlike in the application considered in the present paper) have several outlinks, since most scientific papers have a bibliography consisting of ten or more other papers.

References

[1] S. Argamon, K. Bloom, A. Esuli, F. Sebastiani, Automatically determining attitude type and force for sentiment analysis, in: H. Uszkoreit, Z. Vetulani (Eds.), Human Language Technology: Challenges of the Information Society, Springer Verlag, Heidelberg, DE, 2009, pp. 218–231.
[2] G. Attardi, M. Simi, Blog mining through opinionated words, in: Proceedings of the 15th Text Retrieval Conference (TREC 2006), Gaithersburg, US.
[3] S. Baccianella, A. Esuli, F. 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 2010), Valletta, MT.
[4] G. Berardi, A. Esuli, F. Sebastiani, F. Silvestri, Blog distillation via sentiment-sensitive link analysis, in: Proceedings of the 17th International Conference on Applications of Natural Language Processing to Information Systems (NLDB 2012), Groningen, NL, pp. 228–233.
[5] A. Borodin, G.O. Roberts, J.S. Rosenthal, P. Tsaparas, Link analysis ranking: algorithms, theory, and experiments, ACM Transactions on Internet Technology 5 (2005) 231–297.
[6] S. Brin, L. Page, The anatomy of a large-scale hypertextual Web search engine, in: Proceedings of the 7th International Conference on the World Wide Web (WWW 1998), Brisbane, AU, pp. 107–117.
[7] C. Buckley, E.M. Voorhees, Retrieval evaluation with incomplete information, in: Proceedings of the 27th ACM International Conference on Research and Development in Information Retrieval (SIGIR 2004), Sheffield, UK, pp. 25–32.
[8] D. Cai, X. He, J.R. Wen, W.Y. Ma, Block-level link analysis, in: Proceedings of the 27th ACM International Conference on Research and Development in Information Retrieval (SIGIR 2004), Sheffield, UK, pp. 440–447.
[9] E. DiMuzio, S.S. Sundar, Does a hyperlink function as an endorsement? in: Proceedings of the 7th International Conference on Persuasive Technology, Linköping, SE, pp. 268—273.
[10] X. Ding, B. Liu, P.S. Yu, A holistic lexicon-based approach to opinion mining, in: Proceedings of the International Conference on Web Search and Data Mining (WSDM 2008), Palo Alto, US, pp. 231–240.
[11] C. Fellbaum (Ed.), WordNet: An Electronic Lexical Database, MIT Press, Cambridge, US, 1998.
[12] S. Fragoso, Understanding links: web science and hyperlink studies at macro, meso and micro-levels, New Review of Hypermedia and Multimedia 17 (2011) 163–198.
[13] K. Fujimura, T. Inoue, M. Sugisaki, The EigenRumor algorithm for ranking blogs, in: Proceedings of the WWW 2005 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, Chiba, JP.
[14] G.H. Golub, C.F.V. Loan, Matrix Computations, third ed., The Johns Hopkins University Press, Baltimore, US, 1996.
[15] T.H. Haveliwala, Topic-sensitive PageRank: a context-sensitive ranking algorithm for Web search, IEEE Transactions on Knowledge and Data Engineering 15 (2003) 784–796.
[16] M.A. Hearst, M. Hurst, S.T. Dumais, What should blog search look like? in: Proceeding of the 2008 ACM Workshop on Search in Social Media (SSM 2008), Napa Valley, US, pp. 95–98.
[17] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004), Seattle, US, pp. 168–177.
[18] D. Jurafsky, J.H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, 2nd ed., Computational Linguistics, and Speech Recognition, Prentice-Hall, Upper Saddle River, US, 2009.
[19] A. Kale, A. Karandikar, P. Kolari, A. Java, T. Finin, A. Joshi, Modeling trust and influence in the blogosphere using link polarity, in: Proceedings of the 1st International Conference on Weblogs and Social Media (ICWSM 2007), Boulder, US.
[20] M. Keikha, M.J. Carman, F. Crestani, Blog distillation using random walks, in: Proceedings of the 32nd ACM Conference on Research and Development in Information Retrieval (SIGIR 2009), Boston, US, pp. 638–639.
[21] J.S. Kessler, N. Nicolov, Targeting sentiment expressions through supervised ranking of linguistic configurations, in: Proceedings of the Third International AAAI Conference on Weblogs and Social Media (ICWSM 2009), San Jose, US.
[22] J.M. Kleinberg, Authoritative sources in a hyperlinked environment, Journal of the ACM 46 (1999) 604–632.
[23] B. Liu, Sentiment Analysis and Opinion Mining, Morgan and Claypool Publishers, San Rafael, US, 2012.
[24] C. Macdonald, I. Ounis, I. Soboroff, Overview of the TREC 2009 Blog Track, in: Proceedings of the 18th Text Retrieval Conference (TREC 2009), Gaithersburg, US.
[25] C. Macdonald, R.L. Santos, I. Ounis, I. Soboroff, Blog Track research at TREC, SIGIR Forum 44 (2010) 58–75.
[26] J.R. Martin, P.R. White, The Language of Evaluation: Appraisal in English, Palgrave, London, UK, 2005.
[27] G. Mishne, Multiple ranking strategies for opinion retrieval in blogs, in: Proceedings of the 15th Text Retrieval Conference (TREC 2006), Gaithersburg, US.
[28] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, C. Lioma, Terrier: a high performance and scalable information retrieval platform, in: Proceedings of the SIGIR 2006 Workshop on Open Source Information Retrieval (OSIR 2006), Seattle, US.
[29] I. Ounis, C. Macdonald, I. Soboroff, Overview of the TREC 2010 Blog Track, in: Proceedings of the 19th Text Retrieval Conference (TREC 2010), Gaithersburg, US.
[30] B. Pang, L. Lee, Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval 2 (2008) 1–135.
[31] S.S. Piao, S. Ananiadou, Y. Tsuruoka, Y. Sasaki, J. McNaught, Mining opinion polarity relations of citations, in: Proceedings of the 7th International Workshop on Computational Semantics (IWCS 2007), Tilburg, NL, pp. 366–371.
[32] S.E. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Foundations and Trends in Information Retrieval 3 (2009) 333–389.
[33] M. Shokouhi, L. Si, Federated search, Foundations and Trends in Information Retrieval 5 (2011) 1–102.
[34] H. Tong, C. Faloutsos, J.Y. Pan, Fast random walk with restart and its applications, in: Proceedings of the 6th International Conference on Data Mining (ICDM 2006), Washington, US, pp. 613–622.