Combining Document and Sentence Scores for Blog Topic Retrieval

Jose M. Chenlo, David E. Losada

Grupo de Sistemas Inteligentes
Departamento de Electrónica y Comunicación
Universidad de Santiago de Compostela, Spain
{josemanuel.gonzalez,david.losada}@usc.es
http://www.gsi.dec.usc.es/ir

Abstract. In recent years topic retrieval has become a core component in blog information retrieval. In this area, non-relevant documents that contain many query terms by chance or in the wrong context might be highly ranked. To address this issue, in this paper we propose some adjustments to effective blog retrieval methods based on the distribution of sentence scores. We hypothesize that we can successfully identify truly relevant documents by combining score features from document and sentences. This helps to detect right contexts related to queries. Our experimental results show that some of our proposed variants can outperform state-of-the-art blog topic retrieval models.

Key words: blog retrieval, query context, sentence retrieval

1 Introduction

With the rise of social networks and blogs, new opportunities and challenges appear to handle properly the opinionated nature of these resources. Our research is focused on one of the most important web sentiment-oriented resources: the blogosphere. People read frequently blogs to be acquainted with others' viewpoints. In order to build an effective information retrieval (IR) system which helps users to find what other people think, we have to consider opinions as a first-class object [1].

In the literature, most blog retrieval tasks are approached by a two-phase process that involves a topic retrieval task and a re-ranking phase based on opinionated features. The performance of the opinion finding system has proved to be strongly dependent on the quality of the initial topic retrieval process [2]. We design here powerful retrieval algorithms able to support successfully the subsequent opinion mining tasks. More specifically, we propose a new method to support blog IR based on the distribution of salient sentences in documents retrieved by state of the art models. A high sentence score is associated to a good matching with the query. Hence, by promoting documents with high score sentences, we could successfully identify documents that contain terms related to the query in the right context. By analyzing the flow of the presumed relevant

sentences in the BLOGS06 TREC collection [3] we were able to distinguish relevance-flow patterns that help us to identify relevant documents.

Our experiments show that this sentence-level information can be useful to push defective documents lower into the rank. Specifically, we consider several features such as the ratio of high-scored sentences in a document (peaks), the median number of unique terms matched for each sentence, the variance of sentence scores in a document and the maximum score of the sentences belonging to a document. These sentence-level features provide valuable information about the way in which a document matches a query.

The rest of the paper is organized as follows. Section 2 presents related work. Standard blog retrieval methods and the methodology followed to combine sentence retrieval scores with document scores are explained in Section 3. Section 4 reports the experiments and analyzes their outcomes. The paper ends with Section 5, where we present the conclusions and future lines of work.

2 Related Work

The main promoter of research in blog retrieval has been the TREC conference. The blog track was introduced as a main task in TREC 2006, continued in TREC 2007, 2008 and 2009 [4,5,2,6] and is running in TREC 2010. One of the lessons learnt from early blog tracks is that opinion-finding is strongly dominated by the underlying document ranking performance (topic-relevance baseline). As a matter of fact, rankings provided by good topic-retrieval baselines perform reasonably well for the opinion-finding task without any opinion-based adjustment. In past blog tracks, many research teams did not manage to improve the opinion-finding performance of their own topic-relevance baselines by applying opinion mining techniques. This motivated some adjustments in the proposed tasks. Since TREC 2008, the blog opinion retrieval task (the main task of blog track) was divided into two independent subtasks. In the first subtask (baseline adhoc retrieval task), traditional IR is used to find topically relevant documents. The second task (opinion-finding retrieval task) consists of finding subjective (positive/negative/neutral) posts and researchers were encouraged to apply their opinion-finding techniques on their own retrieval baselines or on a set of baselines provided by TREC. The idea was to provide the participating groups with an experimental setting where they could assess the impact of their opinion-finding techniques across a set of different topic-relevance baselines. Through this experiment, the Blog track went towards a better understanding of the most effective and stable opinion-finding techniques, by observing their performances on common topic-relevance baselines.

Our work is focused on blog topic retrieval and, more specifically, on the use of sentence-based features to improve these systems. Sentence or passage-level evidence has shown its merits in blog retrieval environments. Lee et al. [7] used a passage-based retrieval model combined with a feedback engine in TREC 2008 and their results demonstrated the strength of their system, being one of the best topic retrieval approaches in the track. Santos et al. [8] applied a novel approach

that takes into account the proximity of query terms to subjective sentences in a document. This led to improvements over state of the art baselines for blog opinion retrieval. We study here a new sentence-level approach based on the relevance flow of sentence scores.

Sentence-level features have also been used in other scenarios. Seo and Jeon [9] built a high precision document retrieval system supported by several sentence-based features. Specifically, they used features based on the position and distribution of the high-scored sentences to learn a probabilistic model that can distinguish relevance-flow patterns for relevant documents. Passage-level evidence has been recurrently applied in passage-based document retrieval. For instance, in [10], Kaszkiel and Zobel proposed several methods to estimate the relevance of documents by aggregating query-passage scores. Finally, the relation between relevance and subjectivity has been explored in other scenarios such as sentence retrieval [11].

3 Blog Topic Retrieval

Before we go into details about our method it is necessary to enumerate the main design points in blog information retrieval systems:

- *Retrieval Unit:* Many researchers choose the documents from blog permalinks [1] as retrieval units of their systems. The permalink document contains complete information about the blog entry: title, post, comments and many noisy components that is necessary to deal with. Other studies opted for fine-grained retrieval units. Recently, Lee et al. have successfully applied retrieval at passage level and, next, the passage scores were aggregated to build a permalink retrieval system [7]. The selection of the retrieval unit is therefore an open issue in blog retrieval systems.
- *Document preprocessing:* There is much noise in the structure of blog pages. In most permalink documents we can find off-topic data such as links to other blog posts, information related to the blog and many advertisements. This data might harm severely retrieval performance by promoting deceptive documents that have query terms in a wrong context. Effective preprocessing to identify the key components of the permalink (i.e. title, post and comments) and discard noisy pieces of information is necessary to obtain a good performance.
- *Topic Retrieval Method:* In past TREC tasks state of the art models (e.g. BM25, Language Models) worked very well in blog information retrieval, being extremely difficult to beat [2].

More details about these stages of the blog retrieval process and our design decisions are explained below.

[1] The permanent link to a specific page within a blog or post that remains unchanged. Permalinks are useful for bookmarking or tagging a specific blog post for future reference.

3.1 Retrieval Unit and Document preprocessing

The first decision was to set the unit of retrieval of our system. We opted for one of the most common units used in past TREC tracks: permalink documents from blog pages. The selection of this piece of information simplifies the evaluation of our system because the TREC blog track demands the construction of a ranking of permalinks (ordered decreasingly by presumed relevance).

Permalink files have useless information within their internal structure (e.g. links and advertisements). To remove noise we have built a preprocessing unit able to identify the main permalink components (i.e. title, post and comments) and discard the rest of the documents' content. This unit uses a common HTML parser [12] to process the structure of permalinks documents and a set of heuristics to find the core components. The main idea is to detect pieces of text in different HTML blocks and then classify them according to their positional information and size. This type of heuristics has been used in other contexts with relative success [13].

3.2 Baseline Retrieval

Once we have defined and correctly processed the documents of our collection, we need an effective information retrieval model. We decided to use the BM25 model [14] because it has been one of the strongest and powerful methods used since 1994 in the information retrieval field [15]. We used the Lemur's implementation of BM25 matching function [16]:

$$w = \log\left(\frac{N - n + 0.5}{n + 0.5}\right) \tag{1}$$

$$BM25(D,Q) = \sum_{t \in Q} w \cdot \frac{(k_1 + 1)\, tf_{t,D}}{k_1\left((1-b) + b \times (L_D / L_{ave})\right) + tf_{t,D}} \frac{(k_3 + 1)\, tf_{t,Q}}{K_3 + tf_{t,Q}} \tag{2}$$

where N is the total number of documents in the collection, n is number of documents that contain the term t, $tf_{t,D}$ is the frequency of t in document D, $tf_{t,Q}$ is the frequency of t in query Q, L_D and L_{ave} are the length of document D and the average document length in the whole collection. b, $K1$ and $K3$ are free parameters that we have to train.

3.3 Obtaining Sentence Scores

The sentence retrieval module is composed of two components: a preprocessing component and a weighting component. Given the collection of blogs, we split the documents into sentences (offline, more details about the splitting process are given in section 4). For efficiency reasons, we store all sentences in an inverted index. Finally, at query time, we only have to retrieve all the sentences of the collection that match at least one query term.

Finally, we have to select the appropriate weighting model for Sentence Retrieval (SR). Traditional SR methods proposed in the literature are often based on a regular matching between the query and every sentence. A vector-space approach, the tfisf model [17], is a simple but very effective SR method [18]. It is parameter-free but performs at least as well as tuned SR methods based on Language Models or BM25. The tfisf matching function is:

$$tfisf(S,Q) = \sum_{t \in Q} \log\left(tf_{t,Q} + 1\right) \log\left(tf_{t,S} + 1\right) \log\left(\frac{n+1}{0.5 + sf_t}\right) \tag{3}$$

where $tf_{t,Q}$ and $tf_{t,S}$ are the number of occurrences of the term t in the query Q and sentence S, respectively; sf_t is the number of sentences where t appears, and n is the total number of sentences in the collection.

3.4 Combining Document and Sentence Scores

Once we have selected document-query and sentence-query matching functions, it is necessary to define a combination method that assigns a single score to every document. For efficiency reasons, our adjustment was modeled as a re-ranking process for each query, in which we only re-rank the top-ranked documents retrieved by BM25 $\left(D_{T_q}\right)$[2].

First, to have the scores in the same scale ([0,1]) we apply the following normalization:

$$BM25_{norm}(D,Q) = \frac{BM25(D,Q)}{max_{D_i \in C} BM25(D_i,Q)} \tag{4}$$

$$tfisf_{norm}(S,Q) = \frac{tfisf(S,Q)}{max_{S_i \in D_{T_q}} tfisf(S_i,Q)} \tag{5}$$

where C is the collection of documents and S_i are the sentences belonging to top retrieved documents. Observe that the normalization in equation 5 considers all sentences in the BM25's top 1000 retrieved documents. In order to obtain these sentences, for each query we rank all sentences in the collection (using our inverted index of sentences) and then we filter out the sentences belonging to documents that are not in the top-ranked list.

Our objective in this paper is to apply sentence-level features able to clarify the pattern of matching between relevant documents and queries. The features used in our study are the following[3]:

- *Ratio of peaks:* The ratio of peaks in a document, being a peak each sentence of the document which has a normalized score greater than 0.5. We calculate the ratio of peaks as $(\#peaks/\#sentences)$. This measure might help us to promote documents with several sentences very similar to the query rather

[2] For each query, we re-rank the first 1000 documents from the BM25 retrieval ranking.

[3] Variance and Median were calculated only for sentences that match at least one query term.

than documents that contain many query terms scattered. We chose this threshold because the number of sentences that could be considered as a peak should be low (as a matter of fact many documents have no peaks). Hence, we hypothesize that the use of this threshold will be useful to detect only the best query-related sentences. The same threshold has been used in other studies to model the importance of sentences in documents [9].
- *Variance:* The variance of non-zero sentence scores in the document. With this feature we can model the tendency of query matching throughout the document. It is interesting to detect these matching trends in relevant documents and to study how they differ from the non-relevant documents trends.
- *MedianU:* The median of the number of unique query terms matched by the sentence in the document. We hypothesize that documents with high median score are potentially relevant because they contain many sentences on topic. In contrast, documents with low median scores have sentences with poor matching with the query. By incorporating this feature we expect to detect right query-context in documents.
- *Max:* The maximum score of the sentences belonging to a document. This measure promotes documents with a sentence that is presumably very related to the query. This is similar to passage-based retrieval methods that estimate relevance using the highest score passage.

The function used to combine the document and sentence-level score features is simply:

$$sim(D,Q) = \alpha \cdot BM25_{norm}(D,Q) + \beta \cdot SF_{norm}(D,Q) \tag{6}$$

where $SF_{norm}(D,Q)$ is one of the scores explained above and α and β are free parameters trained by linear regression [19].

In this paper we only consider the individual incorporation of these features in our model. Combinations of multiple features and more formal combination models (which take into account the possible dependency between document and sentence features) will be studied in the future.

4 Experiments

In the evaluation, we chose the Lemur Toolkit for indexing and retrieval tasks [16]. The remainder of this section details the aspects related with our experimentation methods and the results obtained.

4.1 Collection and Topics

Our experiments are based on the BLOGS06 TREC collection [3], which is one of the most renowned blog test collections with relevance assessments. Each token has been preprocessed with Porter stemmer and standard English stopwords have been removed. Some statistics of the collection are reported in Table 1.

Table 1. The main statistics of the BLOGS06 collection. This collection was utilized in the TREC 2006, TREC 2007 and TREC 2008 blog tracks

Attribute	Value
Number of Unique Blogs	100,649
RSS	62%
Atom	38%
First Feed Crawl	06/12/2005
Last Feed Crawl	21/02/2006
Number of Feeds Fetches	753,681
Number of Permalinks	3,215,171
Number of Homepages	324,880
Total Compressed Size	25GB
Total Uncompressed Size	148GB
Feeds (Uncompressed)	38.6GB
Permalinks (Uncompressed)	88.8GB
Homepages (Uncompressed)	20.8GB

In our experiments we worked with the TREC 2006, TREC 2007 and TREC 2008 blog track test collections. All of them utilize the BLOGS06 document collection as the reference collection for the retrieval experiments. Every year a new set of topics was provided and new relevance judgments were made according to the documents retrieved by participants. Details about these topics are reported in Table 2. We built three realistic and chronologically organised query datasets with these topics. First of all, to train the BM25 free parameters we used the TREC 2006 topics for training and the TREC 2007 and TREC 2008 topics for testing. We applied two different training/testing configurations for our regression model. Details about these configurations are shown in Table 3.

Table 2. Topics provided in the TREC Blog tracks

Blog Track	Topics(#)
2006	851-900 (50)
2007	901-950 (50)
2008	1001-1050 (50)

Table 3. Training and testing configurations used for regression

Label	Training	Testing
TREC 2007	851-900(2006)	901-950(2007)
TREC 2008	851-900(2006), 901-950(2007)	1001-1050(2008)

Each TREC topic contains three different fields (i.e. title, description and narrative) but we only utilized the title field, which is short and the best representation of real user web queries, as reflected in the official TREC Blog track literature [4,5,2].

The measures applied to evaluate the retrieval performance in our system were mean average precision (MAP), Precision at 10 documents (P@10) and Precision at 5 documents (P@5), which are commonly used in blog environments.

Finally, in order to detect all the sentences contained in the documents we used a Java port of the Carnergie Mellon University link grammar parser [20]. This software is included in the MorphAdorner project [21], developed by Northwestern University of Chicago. The link grammar parser is a natural language parser based on link grammar theory. Given a sentence, the system assigns to the sentence a syntactic structure consisting of a set of labeled links connecting pairs of words. The parser also produces a "constituent" representation of a sentence (showing noun phrases, verb phrases, etc.). With this software we were able to build a collection of sentences by tagging the sentences of the original parsed blog text collection. For efficiency reasons, these sentences were stored in a sentence-level inverted index.

4.2 Retrieval Baselines

Our ranking functions are tf-isf (sentence similarity) and BM25 (document similarity). Since tf-isf is parameter-free we only have to tune the BM25 parameters.

BM25 depends on three parameters: $k1$, which controls term frequency; b, which is a length normalization factor; and $k3$, which is related to query term frequency. We fixed $k3$ to 0 (observe that we work with short queries and, therefore, the effect of $k3$ is negligible[4]) and experimented with $k1$ and b values from 0 to 2 (both of them in steps of 0.1). The most robust parameter configuration is reported in Table 4, where we also include the performance obtained with the well-known BM25 suggested configuration ($k1 = 1.2, b = 0.75$). The configuration learnt in the training collection yields better performance than the default BM25 setting. This is somehow surprising because the default BM25 setting has proved to be very robust in many document retrieval experimentations [23]. Our optimal setting fixed $k1$ to 0.7 (instead of 1.2, which is the default value). Still, we found that performance is not very sensitive to $k1$ (the default $k1$ setting leads to quasi-optimal performance). The major difference between the default BM25 configuration and our configuration lies in the b parameter. The typical BM25 value for this parameter is 0.75 but, we observed a significant improvement in performance with parameters smaller than the default value. Specifically, we obtained the best performance with $b = 0.3$. We hypothesize that this is related to the nature of the documents. b is used as a length normalization factor. High b values increase the penalization for long documents in retrieval systems. On the other hand, smaller values apply less length normalization. In the case of BLOGS06 collection, where documents are posts and comments (usually the

[4] As a matter of fact $K3$ is barely used today[22]

main piece of text in a permalink), the distribution of lengths might be less skewed than the distribution found in standard text collection. Anyway, this requires further investigation that is out of the scope of this paper.

Table 4. BM25 results in TREC 2007 (topics 901-950) and TREC 2008 (topics 1001-1050) blog task topics. BM25 parameters were trained with 2006 (topics 851-900) topics. Statistical significance was estimated using the paired t-test (confidence levels of 95% and 99% were marked with $*$ and †, respectively)

	MAP		P@5		P@10	
	default $b = 0.75$ $k1 = 1.2$	trained $b = 0.3$ $k1 = 0.7$	default $b = 0.75$ $k1 = 1.2$	trained $b = 0.3$ $k1 = 0.7$	default $b = 0.75$ $k1 = 1.2$	trained $b = 0.3$ $k1 = 0.7$
2007	.3489	**.4017**$*$†	.6200	**.6600**	.6080	**.6440**
△%		(+15.13%)		(+6.45%)		(+5.92%)
2008	.3237	**.3812**$*$†	.6440	**.6760**	.6340	**.6640**
△%		(+17.76%)		(+4.97%)		(+4.73%)

These initial results are very promising. With a good preprocessing and parameter setting we were able to obtain a strong baseline which is competitive with TREC 2007 and 2008 blog retrieval systems [5,2]. The BM25 parameters learnt in this process ($b = 0.3$ and $K1 = 0.7$) were fixed for the subsequent experiments.

4.3 Experimental Results

In this section, we discuss the results of our combination model with the two datasets discussed above: TREC 2007 and TREC 2008.

Table 5 presents the results of our approach against BM25. The first column reports the baseline (BM25) and the rest of the columns our model with different sentence-level features: *ratio of peaks*, *medianU*, *variance* and *max*. The best value in each row is bolded. Statistical significance was estimated using the paired t-test (confidence levels of 95% and 99% were marked with $*$ and †, respectively) between each combined model and the baseline. An additional row was included for each dataset to report the values of α and β learnt in the training process.

Two features yield to improvements in performance over the blog retrieval baseline: $RatioPeaks$ and $MedianU$. $RatioPeaks$ significantly outperforms the baseline in terms of MAP and also improves consistently the rest of metrics (i.e. P@5 and P@10). Ratio of peaks is therefore the feature that performs the best. On the other hand, $MedianU$ improves consistently MAP and P@5 but is slightly less competitive in terms of P@10.

The combination weights obtained in the training process for both features help to understand the reasons behind the improvements in performance. The $RatioPeaks$ configuration has a negative weight for the feature score, meaning

Table 5. Results with TREC 2007 and TREC 2008 dataset

		$\alpha \cdot BM25_{norm} + \beta \cdot SF_{norm}$			
	baseline	RatioPeaks	MedianU	Var	Max
		TREC 2007			
(α, β)		(2.0751,-0.3484)	(2.0916,0.0642)	(1.9952,1.8750)	(1.9701,0.1716)
MAP	.4017	**.4100**$*$	.4060	.3987	.3965
△%		(+2.07%)	(+1.07%)	(-0.75%)	(-1.31%)
P@5	.6600	**.6960**	.6640	.6120	.6200
△%		(+5.45%)	(+0.60%)	(-7.84%)	(-6.45%)
P@10	.6440	**.6880**$*$	.6360	.6140	.6000
△%		(+6.83%)	(-1.26%)	(-4.89%)	(-7.33%)
		TREC 2008			
(α, β)		(1.5944,-0.2436)	(1.5345,0.1042)	(1.4578,5.0085)	(1.3233,0.4103)
MAP	.3812	.3863$*^{\dagger}$	**.3922**	.3567	.3665
△%		(+1.34%)	(+2.89%)	(-6.87%)	(-4.01%)
P@5	.6760	.6880	**.7040**	.5120	.5680
△%		(+1.78%)	(+4.14%)	(-32.03%)	(-19.01%)
P@10	.6640	**.6900**	.6780	.5460	.5740
△%		(+3.92%)	(+2.11%)	(-21.61%)	(-15.68%)

that we promote documents with few *RatioPeaks*. Observe also that we only re-rank top-retrieved documents and, therefore, these sentence-level features are applied to documents with high query-document similarity score. By promoting top-ranked documents with few peaks we are selecting documents that have the query-document score highly concentrated in a few sentences. On the other hand, documents with many peaks have the score distributed over many places. This seems to indicate that query topics should be discussed in few concentrated places of the document.

MedianU has positive weight for the feature score, meaning that we promote documents with a high *medianU*. By promoting this kind of documents we are giving less weight to documents that contain many sentences with poor overlapping with the query (few query terms), and hence, vaguely related to the query. For example, consider two documents, D_1 and D_2, and a query that matches with two D_1 sentences and six D_2 sentences. If the number of unique query terms matched by D_1 sentences and D_2 sentences are $\{3, 4\}$ and $\{1, 1, 1, 2, 1, 1\}$, respectively [5] then: we can observe that D_2 has more matching sentences than D_1, but D_1 sentences are more highly related to the query. Hence, D_1 is likely more relevant than D_2 (in terms of *medianU* D_1 and D_2 have a score of 3.5 and 1.5 respectively).

The analysis of *RatioPeaks* and *MedianU* suggests that we should prefer documents with a few focalized (and high-quality) sentences on topic rather than documents with many (low-quality) sentences poorly related to the query. Re-

[5] Each document is represented by the number of unique terms matched by their sentences. Note that we only consider sentences that match at least one query term.

garding to the behavior of the rest of features (*var* and *max*), *max* was not able to produce any benefit. This seems to indicate that a single high-quality sentence is not enough (a relevant post should contain a few more on-topic sentences). With respect to the variance, it has shown its merits in past document-retrieval studies [9]. In these works we can observe that relevant documents have higher variance of scores than non-relevant documents, and variance of sentence scores is extremely useful to estimate relevance in topic-retrieval rankings. We believe that a future analysis of feature trends in the collection will help us to find a better way to include this feature into our models.

5 Conclusions and Future Work

In this paper, we have proposed a novel way to incorporate sentence-level features in blog topic retrieval baselines. We worked with an effective BM25 parameter configuration (which outperforms the default BM25 configuration in blog retrieval tasks) and we adapted this retrieval baseline in order to incorporate new document features based on sentence scores.

We have evaluated the effectiveness of our approach by combining four sentence-oriented features in two different datasets. We found two features (ratio of peaks and median of unique terms) that offer a good performance and are able to define combinated models that outperform state-of-the-art models for blog topic retrieval.

Our experiments have demonstrated that individual document features related to the pattern of matching between query and document sentences can outperform effective blog retrieval methods. This work is our first step to understand relevance through analysis at the sentence level. In the future, we plan to explore different methods to combine more than one sentence-level feature.

Acknowledgments

This work was partially supported by FEDER and Xunta de Galicia under projects references 2008/068 and 07SIN005206PR.

References

1. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval **2**(1-2) (2007) 1–135
2. Ounis, I., Macdonald, C., Soboroff, I.: Overview of the TREC 2008 blog track. In: Proc. TREC 2008, the 17th Text Retrieval Conference, Gaithersburg, United States (2008)
3. Macdonald, C., Ounis, I.: The TREC Blogs 2006 collection: Creating and analysing a blog test collection. Technical Report TR-2006-224, Department of Computing Science, University of Glasgow (2006)
4. Ounis, I., Macdonald, C., de Rijke, M., Mishne, G., Soboroff, I.: Overview of the TREC 2006 blog track. In: Proc. TREC 2006, the 15th Text Retrieval Conference. (2006)

5. Macdonald, C., Ounis, I., Soboroff, I.: Overview of the TREC 2007 blog track. In: Proc. TREC 2007, the 16th Text Retrieval Conference, Gaithersburg, United States (2007)
6. Macdonald, C., Ounis, I., Soboroff, I.: Overview of the TREC 2009 blog track. (2009)
7. Lee, Y., Na, S.H., Kim, J., Nam, S.H., Jung, H.Y., Lee, J.H.: KLE at TREC 2008 blog track: Blog post and feed retrieval. In: Proc. TREC 2008, the 17th Text Retrieval Conference, Gaithersburg, United States (2008)
8. Santos, R.L.T., He, B., Macdonald, C., Ounis, I.: Integrating proximity to subjective sentences for blog opinion retrieval. In: Proc. 31st European Conference on Information Retrieval , ECIR 2009. (2009) 325–336
9. Seo, J., Jeon, J.: High precision retrieval using relevance-flow graph. In: Proc. 32nd Annual International ACM SIGIR Conference. (2009) 694–695
10. Kaszkiel, M., Zobel, J.: Effective ranking with arbitrary passages. JASIST **52**(4) (2001) 344–364
11. Fernández, R.T., Losada, D.E.: Using opinion-based features to boost sentence retrieval. In: Proceedings of the ACM 18^{th} Conference on Information and Knowledge Management (CIKM 2009), Hong Kong, China, ACM press (November 2009)
12. Jericho, M.: Jericho HTML parser. `http://jericho.htmlparser.net/docs/index.html` (2009)
13. Parapar, J., Barreiro, A.: An effective and efficient web news extraction technique for an operational newsir system. In: XIII Conferencia de la Asociación Española para la Inteligencia Artificial CAEPIA - TTIA 2007, Salamanca, Spain, Actas Vol II (November 2007) 319–328
14. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3. In: Proc. of TREC-3, the 3th Text Retrieval Conference, Gaithersburg, United States (1994)
15. Armstrong, T.G., Moffat, A., Webber, W., Zobel, J.: Has adhoc retrieval improved since 1994? In: Proc. 32nd Annual International ACM SIGIR Conference. (2009) 692–693
16. The Lemur Project. `http://www.lemurproject.org/`
17. Allan, J., Wade, C., Bolivar, A.: Retrieval and novelty detection at the sentence level. In: Proc. 26th Annual International ACM SIGIR Conference. (2003) 314–321
18. Losada, D.E.: Statistical query expansion for sentence retrieval and its effects on weak and strong queries. Information Retrieval 2010 (2010)
19. Croft, W.B.: 1. In: Combining Approaches to Information Retrieval. Kluwer Academic Publishers (2000) 1–36
20. The Link Grammar Parser. `http://www.link.cs.cmu.edu/link`
21. MorphAdorner Project. `http://morphadorner.northwestern.edu/`
22. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval **3**(4) (2009) 333–389
23. Robertson, S.: How okapi came to TREC. E.M. Voorhees and D.K. Harman (eds.), TREC: Experiments and Evaluation in Information Retrieval (2005) 287–299