Web Document Retrieval using Passage Retrieval, Connectivity Information, and Automatic Link Weighting – TREC-9 Report

Franco Crivellari   Massimo Melucci*

Department of Electronics and Computer Science
University of Padova (Italy)

Abstract

This report describes the participation at the Web track of the TREC-9 of the Information Management Systems research group of the Department of Electronics and Computer Science at the University of Padova (Italy). TREC-9 has been our first participation to TREC and, then, to the Web track. In the following, we describe the experimental approach we have chosen, the research hypotheses and questions, the problems we encountered, the results we reached and our conclusions. We consider this experience as the first step towards the participation to the next Web tracks.

1 Experimental Approach

The approach we have taken to address the problems and the research questions regards both the scientific side and the implementation side. As regards to the scientific side, we employed an experimental approach that mixes both classical advanced information retrieval (IR) techniques, and connectivity-based algorithms for IR on the Web. Figure 1 depicts the whole process being described below. Specifically, we have chosen those classical IR techniques, i.e. passage retrieval and blind relevance feedback, which have proven to be effective to produce good retrieval results [1]. Moreover, we are interested to test whether the connectivity-based algorithms, which have been proposed in different Web contexts, are effective tools to improve classical techniques. As regards to the implementation side, we developed in-house software and employed other software modules that are publicly available.

*Correspondence author: Dipartimento di Elettronica e Informatica – Via Gradenigo, 6/A – 35131 Padova – Italy – E-mail: melo@dei.unipd.it – Telephone: +39-049-827-7802 – Fax: +39-049-827-7826

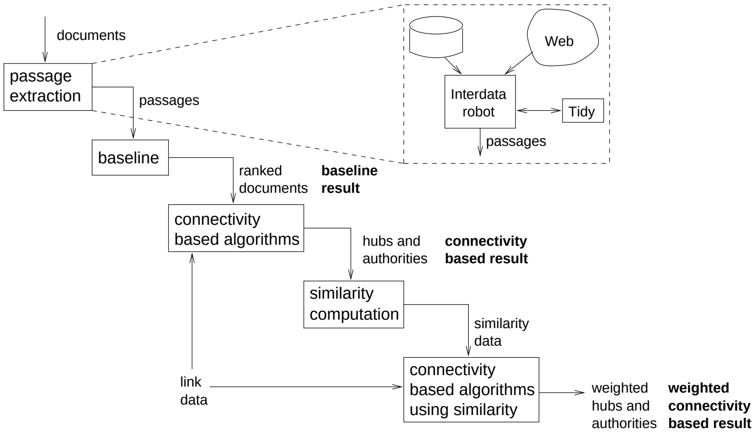


Figure 1: The experimental process. Bold text refers to the submitted runs.

Baseline. First 10 passages – title and paragraphs – are extracted from each document and indexed using a stop-list augmented with Web stopwords, the Porter's stemming algorithm, and by keeping non-stemmed words; for example, the word "White" has been stored together with "white". Title-only and title-description queries are automatically generated, and indexed as passages did. For each query, top 10,000 passages are retrieved and ranked by F-4 [2]. The lists of retrieved passages are reweighted through blind relevance feedback by considering top 100 passages as relevant. The lists of newly 10,000 retrieved passages are mapped to retrieved documents. The document score is the sum is the of the scores of the mapped passages.

Connectivity-based algorithm. A modified version of the HITS (Hyperlink Induced Topic Search) algorithm is applied on the provided link files, where the link weight is the baseline score;

Similarity-based algorithm. In- and out-links are weighed using similarity among documents; the similarity between two documents is the average similarity between the passages of a document and the passages of another document;

Connectivity-based algorithm using similarity. The modified HITS algorithm is applied on the weighted link files, where the weight of the link between two documents is the content similarity between the documents.

We have then submitted six runs – three runs for each query type, i.e. topic title-based queries and topic description and title-based queries:

- baseline: F-4-based passage ranking and query term reweighing using blind relevance feedback (PuShortBase, PuLongBase);

- modified HITS: baseline lists are re-ranked using authority weights that are computed considering links equally (PuShortAuth, PuLongAuth);
- modified HITS with weighted links: baseline lists are re-ranked using authority weights that are computed weighing links by text similarity (PuShortWAuth, PuLongWAuth).

1.1 Web stopword list

Stoplists are fundamental tools to reach effective and efficient indexing and retrieval results. So far, different stoplists have been developed for different languages and application domains. Differently from classical document collection, the Web is a potentially infinite universe which is about many different subjects and is a container of many different languages. Thus, a search engine should be provided with many different stoplists to consider such a myriad. However, a word, which is a stopword in a stoplist of an application domain or for a language, could be a keyword within another application domain or for another language.

Web pages are often rich of terms, words or sentences including strings that represents words of languages and protocols of the Internet and of the Web. Actually, Web pages are written using a mark-up language, such as HTML or XML. Therefore, these sort of documents contain both text encoding the information that are explicitly communicated to the user, and text representing "net-stopwords", i.e. mark-up language or Internet words being used to write the page down and to allow for the transmission of the page through networks. Indexing algorithms does sometimes extract "net-stopwords" and have to decide if to keep them as keywords.

To address the problem of the presence of "net-stopwords", we have developed a list of 65 stopwords that are considered very frequent in Web pages, and that can be considered as "net-stopwords". Examples of "net-stopwords" are HTML words, such as "www" or "html", or the most common strings that are used to compose electronic mail or Web addresses, such as "com" or "net".

We computed the frequency distribution of the most used words in the training set. We realized that the classical stoplist is still valid for IR applications on the Web. Furthermore, we identified additional words and we selected very frequent words that are about the World Wide Web and not about a specific domain.

1.2 Passage Retrieval

We used passage retrieval because Web pages are often long or multi-topic documents. Using the mark-up information and some numerical parameters, such as passage size, we have extracted passages from the Web pages and have used these passages as source of evidence to index and retrieve documents. From each document, we have extracted the following passages:

- meta-data fields, such as authors' names, keywords, and description, identified by the <META> tag,
- page title, identified by the <TITLE> tag,
- paragraphs, identified by the <P> tag,
- headings, identified by the <H1>, <H2>, and <H3> tags.

We have chosen these tags assuming these passages are likely to include most part of the discriminating keywords. Moreover, we assumed that some tags play a specific role to carry the semantic content description of Web pages; for example, page authors are likely to use headings to give weight to the keywords being stored in the headed passages. Similarly, we assumed that page title often contains important keywords, and that paragraphs are effective ways to structure the relevant information. We also assumed that meta-data are effective means to represent relevant information, and to identify relevant document. Indeed, meta-data, e.g. like keywords and description, are manually filled by the page's authors and then they are likely to describe the semantic content precisely and exhaustively. However, we realized that a very low percentage of documents include manually filled meta-data that are the result of an intellectual work of content description, while many of the documents with meta-data include automatically filled fields, such as the page In total, we have extracted 8.6 million passages. The engine retrieved and scored passages before building the list of retrieved documents. composer product name, that are poor semantic content descriptors.

The formula $g_P(d) = \sum_{i=1}^{N_d} g(d_i)$ has been used to compute the document score starting from the passage scores, where: N_d is the number of passages $d_i, i = 1, ..., N_d$ of d, $g(d_i) = \sum_{k=1}^{K} q_k \ t_{ik} \ c_{1k}$, K is the number of index terms, $q_k = 1$ if index term k occurs in the query, $t_{ik} = 1$ if index term k occurs in d_i, c_i is the relevance term weight computed using the distribution of terms in passages.

1.3 Connectivity-based Algorithms

We employed the HITS (Hyper-link Induced Topic Search) algorithm [4] to re-rank the baseline document list. The document list has been given as input to the algorithm and each document has been assigned an authority and a hub weights. Authority weights has been used to rank the list, so that the most authoritative pages are placed on the top of the list.

1.4 Similarity-based Link Weighing

HITS and the modified version we used in our experiments ignore the semantic content of the linked documents, then, links between documents with a dissimilar content are treated equally to links between documents about similar content. To test if semantic content affects the effectiveness of connectivity-based algorithms, we weigh the links being provided with the test collection using a similarity function.

Inter-document similarity-based reweighing is computed as follows. We are provided with two link files – in-link and out-link files. Given a link file, a new weighted link file is computed. After weighing link files, we obtain two weighted link files being similar to the provided link files, but links are weighted using a linear combination of the manual weight, and the similarity between the linked documents. This linear combination uses the coefficient α.

The weight of the link between d and c is $\alpha + (1-\alpha) sim(d, c)$, where α is the weight given to the manual link and $sim(d, c)$ is the inter-document similarity between d and c ($\alpha = 0.5$, in the submitted runs). Figure 2 depicts an example of combination of Web links and similarity links; for example, the weight of the Web link from A to B is $\alpha + (1-\alpha)\frac{1}{3}(.5 + .4 + .2)$ which is the average passage similarity link weight.

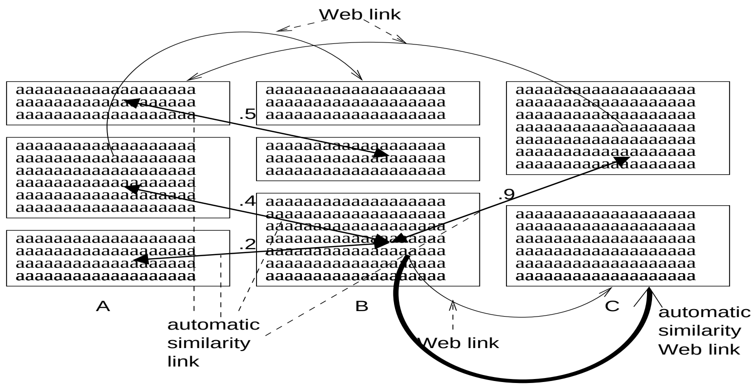


Figure 2: An example of combination of Web link and similarity link. Light arrows represent Web links starting from a passage and ending to a page. Heavy arrows represent similarity link between passages.

2 Development Approach

We have chosen to implement mainly in-house the software being necessary to carry experiments out. We have preferred to supervise the underlying algorithms and to make changes to the software whenever it was necessary.

As regards to the step of passage extraction, we developed a tool to extract passages from Web pages. The tool was originally been designed as a software agent that follows the Web links to retrieve the Web pages; indeed, it is a robot. This robot has been developed within the National InterData research project [5]. For the purposes of the TREC experiments, a different version of the robot has been designed and developed because the data to be retrieved were locally stored, and not on the Web. Moreover, the data are encoded in SGML also and then the tool has been modified to deal with this additional format. To only extract the tagged text, our robot employed a tool for HTML syntax analysis, called Tidy, that is reported in [6]. Tidy allows for correcting HTML syntax by adding, for example, missing end tags.

We reused the TACHIR software library to implement the indexing and retrieval engine [7]. The indexing, retrieval and connectivity analysis software has entirely been implemented in C++ and persistence has been managed using GNU Database Manager (GDBM) [8].

3 Experimental Hypotheses and Questions

In carrying our experiments out, we have made some hypothesis, which are listed in the following:

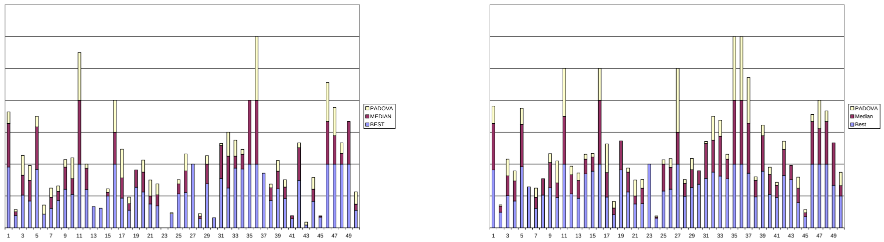
- Passage retrieval and blind relevance feedback are useful. Past research and experiments have shown that extracting passages and using blind relevance feedback are effective means to improve performance. We have therefore employed those methods and produced baseline results that already incorporates them. Thus, we made no comparison with experiments without passage retrieval and blind relevance feedback.

  At training phase, we tested that passage retrieval and blind relevance feedback for query term reweighing are effective means to improve performance. Training was performed using the WT2G test collection and the TREC-7 and TREC-8 topic sets. We then decided to use passage retrieval and blind relevance feedback as method to produce the baseline results.

- Only a part of a Web page can be indexed to reach acceptable levels of effectiveness. We assumed that the representation of relevant information are concentrated in few passages and few passage types. Specifically, we assumed that we could concentrate indexing on only some tags (some tags are useful, others are useless), only the top part of the document, only the initial part of passages.

- The documents are written using the Latin alphabet and in English. We have therefore developed no software being dependent to specific alphabet or language. Apart the Web stoplist, only an English stoplist has been used and only the Porter's

Our experiments aimed to test the impact on effectiveness of connectivity-based algorithms, similarity-based link weighing and connectivity-based algorithms. Specifically, we wanted to test whether the use of the modified version of the HITS algorithm increases the levels of effectiveness reached through the baseline results. Moreover, we wanted to test whether weighing the links employed to perform the modified version of the HITS algorithm increases the levels of effectiveness reached through the baseline results. In other words, we tested whether adding information about the semantic content of the linked documents is useful.



(a) (b)

Figure 3: Baseline results using topic title-based (a) and description-based queries (b)

4 Official Results

Figures 3(a) and 3(b) depict the results reached through the baseline methods using the short query and the long query version, i.e. queries being based on the topic title only, and queries being based on the topic title and description. In both Figures, we reported the best, the median and our precision level after 100 retrieved documents for each topic. Each bar of a histogram refers to a topic and depicts the proportion of a precision level – best, median and our – with respect to the percentage of documents being relevant to the topic. The grey (bottom) part of a bar refers to the best result, the dark (middle) part of a bar refers to the median result, the light (top) part of a bar refers to our result. Table 1 reports the official results expressed as average R-Precision (precision after R docs retrieved).

		Unweighted		Similarity-based	
	Baseline	Authorities	Hubs	Authorities	Hubs
Title-only	18.2%	18.9%	18.9%	18.9%	18.9%
Title+description	16.7%	11.7%	16.7%	16.7%	16.7%

Table 1: The official results.

On average, our results are worse than the median results. In some cases, our result is far less than the median, and then of the best result. Note that, in some cases, our result is comparable to or better than the median or to the best result.

The results reached using topic title and description-based queries are comparable to those reached using topic title-based queries. Indeed, no significant improvements have been reached using longer queries. On average, long query results are better than short query results.

The results reached using the connectivity-based algorithms – modified HITS and similarity weighing links – give no significant variations of the baseline results. The pictorial description of those results would be very similar, and would be equal for many topics.

5 Problems

As we have participated to TREC at the first time, we encountered plenty of problems, mainly because of the need of interleaving implementation issues and methodological problems. This meant that we had to sacrifice some methodological solutions to finish the experiments on time and to cope with some implementation deficiencies. As consequence, we had to limit: The number of passage types – we used only meta-data, paragraphs and headings; the number of retrieved passages – only 10,000 passages are retrieved for each query; the number of passage words – we considered 20 words per passage only; the use of query expansion – queries were not expanded after blind relevance feedback.

Moreover, we had implementation problems. We think that W3C HTML Tidy is too “severe”, yet is a useful and powerful tool to extract passages from Web pages. We encountered other problems related to the presence of Web pages written in Japanese that created some difficulties for our passage extraction software. We had to eliminate these documents semi-automatically. We have “lost” some pages because of the presence of frames and CGI script calls. In one case, we found a page being splitted into two parts – a part is read by the browser if it is enable to process frames, otherwise, the browser reads the other part. The part that is activated if the is enable to process frames stores a call to a CGI scripts and no other data is stored. The other part stores the text that has been indexed, but that is different from the text that would be produced by the CGI script, if called. Therefore, our software indexed the “explicit” text, by the judges maybe assessed the text being produced by the CGI script. We encountered some problems in dealing with passage extraction from very long and non-tagged texts, such as those included by `<PRE>` tags. Of course, we were unable to cope with “wrong” query words, such as “nativityscenes”.

6 Conclusions

After this first experience, we learned a lot about basic issues of text retrieval and about advanced issues of Web page retrieval. Basically, we learned that investing human resources is the most crucial factor affecting results. We believe that we can invest more time to the methodological issues at the next TREC because many implementation problems have been addressed at this TREC.

It is necessary to index all the document – all the tags because they are very often used for presentation purpose and not for carrying semantics; this means that, for example, headings carry no more information than other pieces of text. All the of the document parts because a document can be relevant because there can be a relevant passage on the bottom; this is the case of long documents, especially, but also for short and structured documents, such as list of items that include links. All the passage because there can be many long passages that store relevant information in the middle or at the end of the text.

Passage retrieval requires too large data files if implementing passages as individual documents. We had then to cut passages off, but we have lost many useful information; alternative data structures that employ proximity-based collocations are currently under investigation. More sophisticated document scoring system is necessary. Summing passage scores is a rather simplistic way to compute the score of the document which passages belong to. There can be irrelevant very large documents with many short high scored small passages or few high scored large passages.

The connectivity-based experiments gave no variations probably because we applied no expansion of the root page set by adding in-linked and out-linked pages. Thus, the root page set was equal to the base page set and the connectivity-based algorithms have made no significant changes to the original ranking. The use of the baseline document score and of the similarity-based link weight gave no contributions.

Our experiments confirmed that in classical IR, documents are organized texts and text organization carries some semantics about the document content; on the contrary, Web documents are sometimes more structured than classical documents, but this structure carries little semantics about the document content. In classical IR, end users are expert persons about the application domain, then queries are well formulated and often the query vocabulary correspond to the vocabulary used by the document authors. On the contrary, Web queries are formulated by non-expert persons because the Web collection are not about an application domain.

References

[1] E. Voorhees and D. Harman, editors. *Overview of the Sixth Text Retrieval Conference (TREC-6)*, volume 36(1), 2000.

[2] S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, May 1976.

[3] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[4] J. Kleinberg. Authorative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, September 1999.

[5] F. Crivellari and M. Melucci. Awir: Prototipo di un motore di ricerca per la raccolta, indicizzazione e recupero di documenti web sulla base dei loro frammenti. Rapporto tecnico T2-S12, Progetto INTERDATA - MURST e Università

di Padova: “Metodologie e tecnologie per la gestione di dati e processi su reti Internet e Intranet”. Tema 2: “Estrazione di informazioni distribuite sul WWW”., ftp://ftp-db.deis.unibo.it/pub/interdata/tema2/T2-S12.ps, Febbraio 1999. In Italian.

[6] World Wide Web Consortium (W3C) HTML Tidy. `http://www.w3.org/People/Raggett/tidy/`, October 2000. Last visited: October 25th, 2000.

[7] M. Agosti, F. Crestani, and M. Melucci. Design and implementation of a tool for the automatic construction of hypertexts for Information Retrieval. *Information Processing & Management*, 32(4):459–476, July 1996.

[8] GNU Database Manager (GDBM). `http://www.gnu.org/software/gdbm/gdbm.html`, October 2000. Last visited: October 25th, 2000.