Robust and Web Retrieval with Document-Centric Integral Impacts

Vo Ngoc Anh *Alistair Moffat*

Department of Computer Science and Software Engineering
The University of Melbourne
Victoria 3010, Australia

{*vo,alistair*}@*cs.mu.oz.au*

Abstract: *This paper reports the experiments done at The University of Melbourne for the Robust and Web tracks of* TREC-2003. *We explore the idea of determining the impact of a term locally within the document and in a qualitative manner instead of a quantitative one. The impact of each distinct term in a document or query text is defined to be a small integer. The scalar product of the impact vector for a document and the impact vector for a query is taken to be the similarity score between them, an arrangement that allows very fast query evaluation.*

1 Document-Centric Integral Impacts

Consider a document collection with N documents and n distinct terms. Use is often made of the TF-IDF rule to assess the similarity degree between a query q and any document d of the collection. An n-dimensional vector space is constructed, with each dimension representing a term that appears in the collection. In the space each document d is represented as

$$d = (w_{d,t_1}, w_{d,t_2}, \ldots, w_{d,t_n}),$$

and the query q as

$$q = (w_{q,t_1}, w_{q,t_2}, \ldots, w_{q,t_n}).$$

In this framework, the t_i are the distinct terms of the collection, and $w_{x,t}$ is the projection of document or query x in dimension t. That is, $w_{x,t}$ is the "importance" of t in x, and can be calculated by any formulation obeying the TF-IDF requirement. A similarity score $S(d,q)$ between d and q calculated by the cosine measure is of the form:

$$S(d,q) = \frac{\sum(w_{d,t} \cdot w_{q,t})}{\sqrt{\sum w_{d,t}^2} \cdot \sqrt{\sum w_{q,t}^2}}, \tag{1}$$

where the three summations are over all n terms.

Anh et al. [Anh et al., 2001, Anh and Moffat, 2002a] consider $w_{d,t}/\sqrt{\sum w_{d,t}^2}$ as the *impact* of the term t in d and propose a transformation that maps the impacts into small integers. The transformation is done in the context of the whole document collection. They obtained better effectiveness than using non-transformed impacts even when only $k = 32$ different values are used for transformed impacts.

In our TREC-2003 experiments we define the same mapping locally within each document rather than globally in the whole collection. The rationale for this change is to give all the documents an equal spread of high impact terms and low impact ones.

Because the transformation is done locally within each document, we do not need to rely on the document length factor in Equation 1. We can choose a step further by not to be bound to any particular formulation of $w_{d,t}$, which is often very diverse. In its original form, TF-IDF is not a precise definition – it is a philosophy. It is our human desire for precision in computation that has led to overly precise formulations, with their various tuning factors. Adopting a philosophy that "rank is more important than value" allows us to reduce the effect that numerical constants have on the effectiveness of the ranking [Anh and Moffat, 2002b].

Ideally, we would like to create the sorted list of terms corresponding to each document without doing any detailed computation. The surrogate weights of the n_d terms in document d are then computed for use in the query processing regime, without any further recourse to collection or document statistics.

To achieve this independence, we focus on one document at a time, and divide the process of determining the transformed impacts of terms in a document d into two phases. The first *sorting* phase orders the list of n_d terms d in decreasing order of term contribution. In the second *mapping* phase, each value in the term list is converted to an integer in range 1 to k. As the end of the process, these integer values serve as document term impacts.

The same process can be done to define query term impacts for the terms in any query q.

Finally, the similarity score between d and q is computed as the sum of the products of document term impact and query term impact of the terms shared by d and q.

The question to be faced is how to specify the sorting and the mapping phases. We sorted the terms in decreasing term frequency $f_{d,t}$, with ties broken using increasing f_t as a secondary key. This two-part sort key implicitly imports the TF-IDF rule into our scoring system. Note that the reverse sorting order, where f_t is used as the primary key, and $f_{d,t}$ as the secondary can not be expected to work as well, as the large number of different f_t values means that the TF factor would rarely be brought into play.

In the mapping phase, we follow the main idea of Fibonacci that in a structure, the number of "important" elements should be less than the number of less important ones. Hence we choose the mapping so that the number of elements in a document corresponding to each surrogate weight, in decreasing order of the weights, forms a geometric subsequence. That is, $x_i \approx x_{i+1} \cdot B$, where $B = k^{1/k}$ and $x_0 \approx n_d \cdot (B-1)/(k-1)$ is the number of elements with the highest impact value.

As an example, when a document contains $n_d = 250$ distinct terms and $k = 10$, the most frequent (within that document) 7 terms are assigned the highest impact of ten, and the least frequent 57 terms in that document are assigned the lowest impact value of one.

2 Task Setting and Performance

Robust Track

In the robust track, the collection `TREC45-CR` (that is, disk 4 and disk 5 of the TREC corpus, minus the *Congressional Record*) is employed with 50 old topics and 50 new topics. The opportunity of using the 50 old topics for training was not exploited.

Our goal in this track is to measure the effectiveness of the retrieval methods respective a range of query length. The following type of queries, that are automatically generated from the original topics, are used as input: title-only queries, set `T`; description-only, `D`; title-and-description, `TD`; and the whole topic, query set `TDN`.

The standard settings described in the previous section were applied directly for the Robust track. The number of different impacts was set to $k = 10$, a value found to be appropriate based upon experiments carried out on the TREC-2002 data [Anh and Moffat, 2002b].

Our preliminary experiments with other collection show that using document meta data to compute quasi term frequencies, and using them as surrogate to the raw term frequencies in ordering, leads to improvement on effectiveness. The quasi frequencies can be calculated, for example, as weighted sum

Query type	Precision at 10	No of topics not found in top 10	Area underneath MAP(X) for worst 25 topics
For 50 old topics			
T	0.2800	**8**	0.0029
D	0.2680	**9**	0.0041
TD	**0.3340**	**7**	**0.0093**
TDN	0.2980	8	0.0079
For 50 new topics			
T	**0.4200**	**4**	0.0365
D	0.4480	**5**	**0.0205**
TD	**0.5060**	**1**	**0.0456**
TDN	0.4880	4	0.0383
For all 100 topics			
T	**0.3500**	**12**	0.0087
D	0.3580	**14**	**0.0090**
TD	**0.4200**	**8**	**0.0175**
TDN	0.3930	12	0.0155

Table 1: Effectiveness performance on different type of queries. Figures in bold represent values that are not inferior to the median performance of the 2003 TREC runs for the same category of query.

of frequencies of the terms in different text components. Unfortunately, the different text structures in the sub collections of TREC45-CR meant that we did not assign a weight to each component, and only raw term frequencies were used in our runs.

Table 1 presents the performance of our system with respect to different types of queries. Overall, the TD queries obtain the best performance, and TDN is the worst. The poor performance of the TDN queries is presumed to be a consequence of the noise information in the narrative fields of the original topics. The same observation explains the relative weakness of the D in comparison with the query set T. Another possible reason for the relative good performance of the short queries is that we do not apply any pruning technique to prevent the low impact terms in long queries adding their contributions to the final score.

Web Track

In the Web track, the corpus .GOV is used, with 300 topics for the named page finding task, and another 50 for the topic distillation task.

Meta data is used consistently in .GOV, and we made use of its availability for ordering terms in documents. Incoming and outgoing anchor text is also employed. The way to use these extra information is to count the frequency of terms separately in different component of the text, and use a combination of the partial frequencies to determine a primary key for sorting, prior to impact assignment. The second sorting key is either not used or bound to the IDF factor as in the standard settings.

As the first step, we divide each text into the following components:

- *URL text*. The words in the URL can be valuable for both content search and web search, as it is indicated by many of the last year participants. One possible problem is that in many cases acronyms (such as nist) are used. Content management systems that emit alphanumeric strings of gibberish are also an issue. As an initial remedy to this problem, we preprocess the URL text before indexing. The details of this preprocessing are given in the next subsection.

Run name	Description
W3	Weights of $(1, 2, 2, 4, 4, 8)$ are assigned to components (content, meta text, outgoing anchor, incoming anchor, title, URL); then the weighted sum of these components is used as the sorting key. The IDF factor is not employed.
W4	As for W2, but the component weight vector is $(1, 2, 2, 2, 4, 16)$; and IDF is used as a secondary sort key.
S5	The primary sort key is defined as the complex key that represents the term frequency in the sequence (URL, incoming anchor, outgoing anchor, title, meta, content); with IDF used as a secondary sort key.
W1	As for W3, but only entry pages are indexed. Non-entry pages are ignored.

Table 2: Different sorting methods used to define impacts. The left column is a method identification, with the digit being used as the last digit in the names of our two sets of official runs.

- *Titles*. The title field represent a concise summary of the document, and is usually provided by the author of the document. Other visible markup – for example, headings and bold terms – can also be used with some confidence.
- *Incoming anchor text*. Incoming anchor text can be regarded as being an assessment of the content of this document that is provided by a reader of it, rather than the author of it. The ubiquitous "click here" is not helpful in this regard, and nor are clickable images.
- *Outgoing anchor text*. Outgoing anchor text (provided in links in this document that lead to other documents) do not normally reflect the content of this document. It can, however, be valuable in inferring site structure and page connectivities for the topic distillation task.
- *Metadata*. Other meta text, such as keywords, subject fields, and explicit metadata, may also be of assistance. But there is also a risk that it will have been inherited from a previous document and not edited when the content was revised. Even if it has been edited, it will not have been proofread with the same care as the visible text is.
- *Content text*. The plain text of the page is also a reflection of its content, but not of its importance. Content is the baseline resource of all document retrieval systems.

For each document, after counting the term frequency in each text component, different methods can be used to order the terms in a document based on the number of occurrences of each term in each of these categories. The methods used for the TREC runs are listed in Table 2. In all of the methods, the number of different impact values is set to $k = 10$ (as was also the case for the Robust track).

Manipulating URL text

To facilitate use of the URL text, we have employed a simple process that attempts to de-acronymize it. The two parts considered for expansion are the host name, and the last component of the path name of the document (not counting the default page specification such as `index.html`).

The expansion is based on the title of the page and incoming anchor text, to look for possible variants or abbreviations that might give rise to the words in the URL for that page. If the metadata field "Subject" is available, it is also used as part of this process.

Method	Topic Distillation		Home Page Finding		
	P10	AVP	MRR	F10	NF
W3	0.0660	0.0537	**0.508**	**76.7%**	10.3%
W4	0.0400	0.0559	**0.530**	**75.0%**	10.7%
S5	0.0580	0.0728	**0.527**	**76.0%**	11.0%
W1	**0.0920**	**0.1096**	0.288	40.0%	58.0%

Table 3: Effectiveness of the configurations described in Table 2 for the topic distillation and home page finding tasks of the TREC-2003 Web track. The metric P10 is the precision at 10 documents retrieved; AVP is the average precision over all relevant documents; MRR is the mean reciprocal rank of the first relevant document; F10 is the percentage of topics for which the named page found in top 10; and NF is the percentage of topics for which no named page was found in the top 50 documents retrieved. Figures in bold represent values that are higher than the median performance of the 2003 TREC runs for the same task. Comparative figures are not available for the last column.

For example, document `G00-00-0000000` in `.GOV` was crawled from the URL `http://www.aspe.hhs.gov`. Many of the incoming links that lead to it use variants of "Office Assistant Secretary Planning Evaluation" as their anchor text. The algorithm assigns the words "aspe assistant secretary evaluation planning evaluation hhs" by sorting the set of title, subject, and incoming anchor text words into decreasing frequency, and then choosing for inclusion one matching word for each letter in the URL. Similarly, with `http://www.cs.mu.oz.au`, "computer science cs mu oz" should be identified from anchor text and title text, and used subsequently as URL-related terms.

For document `G00-00-0114013`, the original URL was `http://wwwcalfed.water.ca.gov`, and the URL-based terms are just the words "wwwcalfed water ca". This example is not based upon an acronym, and is correspondingly more complex. It is not satisfactorily handled by the ad-hoc process, and the small number of anchors containing "CALFED Bay Delta Program" were overlooked. The URL `www.unimelb.edu.au` is also hard to handle using the simple mechanism.

As a final example, incoming anchor text for document `G00-00-0497574` from the URL `http://wrf.fsl.noaa.gov` is all "WRF", but the title words "Weather Research and Forecasting (WRF) Model" contribute to the expansion, and allow "wrf weather forecasting research fsl noaa" as URL-derived index terms.

Topic Distillation and Home Page Finding Tasks

This year's topic distillation task seems biased in favor of entry pages. While this is generally true, there are also exceptions – for example, a person with a particular hobby might build an excellent resource page which can be the target of a topic distillation query, but is probably within that person's web site. Consequently, we treat all pages equally in three of our four runs for this task. In the other run only entry pages are indexed, as listed in Table 2. The same configurations were used for the home page finding task. The performance for both of the Web tasks is shown in Table 3.

As can be expected, for the topic distillation, W1 is the best, since it indexes only entry pages. However, this policy dramatically degrades the effectiveness of the named page finding task. A clear conclusion from here is that, if the topic distillation keeps its present definition we need to find another way to address it in order to get a compromise with the home page finding task.

For both of the tasks, all of the three remaining methods give about the same level of effectiveness. That means that the weighting scheme we applied is not effective enough. There should be another, better, way to combine the term frequencies in different components.

3 Practical Considerations

Our experiments were conducted in a 933 MHz Intel Pentium III with 1 GB RAM, a 9 GB SCSI disk for system needs, and four 36 GB SCSI disks in a RAID-5 configuration for data. The computer is also a server for a Beowulf cluster of 16 nodes, each an 800 MHz Intel Pentium III with 256 MB RAM and local hard disk of 40 GB. There is a link with the capacity of 100 Mbits/second to and from each node as well as the server with a network switch. The system runs Debian GNU Linux, and all the programs are written on ANSI C, compiled with gcc, with the optimization flag -O3. For the Robust task, the experiments are done in the server, while all the Web tasks are done by the cluster in a parallel manner.

Except for the methods used determining the impacts, all the index structure and query processing are as described by Anh et al. [2001], and are based on the *MG* software (available at `http://www.cs.mu.oz.au/mg/`, and described by Witten et al. [1999]).

Index construction using the integer impacts is straight forward. After an additional pass to link the incoming anchor text to the target documents (this pass also computes the document frequencies for each term, if needed) the process of assigning impacts can be done locally within a document.

Parallelism is implemented as follow. The data collection is roughly divided between the nodes of the cluster, and each part transferred to the local disk of the corresponding node. The server plays the role of interface – it receives requests for indexing or querying, and sends them to all the nodes. In the case of indexing, the nodes work independently. In the case of querying, after receiving a query, each node produces a list of answers of its own based on its sub-collection, then the results are merged between pairs of node until one of them finally has the complete results. That node then send the results to the server.

Acknowledgement This work was supported by the Australian Research Council.

References

V. N. Anh, O. de Kretser, and A. Moffat. Vector-space ranking with effective early termination. In W. B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel, editors, *Proc. 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 35–42, New Orleans, LA, September 2001. ACM Press, New York.

V. N. Anh and A. Moffat. Impact transformation: Effective and efficient web retrieval. In M. Beaulieu, R. Baeza-Yates, S. H. Myaeng, and K. Järvelin, editors, *Proc. 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–10, Tampere, Finland, August 2002a. ACM Press, New York.

V. N. Anh and A. Moffat. Vector space ranking: Can we keep it simple? In J. Kay, editor, *Proc. Australian Document Computing Symposium*, pages 7–12, Sydney, December 2002b. University of Sydney, Australia.

I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, San Francisco, second edition, 1999.