Improved Retrieval Effectiveness Through Impact Transformation

Vo Ngoc Anh **Alistair Moffat**

Department of Computer Science and Software Engineering
The University of Melbourne
Victoria 3010, Australia
http://www.cs.mu.oz.au/~vo
http://www.cs.mu.oz.au/~alistair

Abstract

Users of web search engines are notoriously parsimonious in their use of search terms, and search effectiveness has tended to be relatively poor on the resulting short queries, especially when compared against the good performance attained by recent systems when working with long *TREC*-like queries. In this paper we examine the reasons why short queries are hard to deal with, and propose a modification to the vector-space paradigm, which we call *impact transformation*. Experiments with a collection of web data and web queries shows that impact transformation significantly boosts retrieval effectiveness. Moreover, the use of quantised values in the transformation allows extremely fast query processing.

Keywords: Information retrieval, web searching, vector space model, normalisation.

1 Introduction

Web-searching is now taken for granted. We type a few words into a query box at a location such as `www.google.com`, and expect to be shown a wealth of matching web pages. When the search engine provides us with links to the information we seek, we move on without much thought; and if the search engine fails to show us useful pages, we scoff, call it "useless", and try either a different query or a different search engine. So ubiquitous is this process that tens of millions of people make daily use of free search services, and billions of pages are indexed.

To provide this service, the search engine must determine, for a set of query terms, a list of pages to be presented. If the search service is to be useful, the pages returned must be *relevant* to the query – that is, we require the underlying computation to be effective. And if the search engine is to be economical, the answer pages must be calculated quickly. That is, we also require the underlying computation to be *efficient*.

Users of web search engines are notoriously parsimonious in their use of search terms [Jansen et al., 1998]. Because of this economy, web search effectiveness has tended to be relatively poor, especially when compared against the good performance attained by recent systems when working with long *TREC*-like queries. (See `trec.nist.gov` for details of the *TREC* Text Retrieval Conference.) On the other hand, considerable care has been paid to the efficiency of web search engines, and query response times are measured in fractions of seconds, even when searching across hundreds of millions of pages.

	Exp.1	*Exp.2*
Collection	*WEB1*	*TREC12*
TREC query set	451–500	051–200 (title)
Collection size (MB)	2,458.09	2,070.29
Number of documents	383,429	741,856
Number of queries	50	150
Terms per query	3.2	3.8
Relevant docs per query	14.2	252.1

Table 1: Parameters and statistics of some sets of resources used for testing retrieval systems.

In this paper we examine the reasons why short queries are hard to deal with, and propose a modification to the vector-space paradigm, which we call *impact transformation*. In short, we "tweak" the similarity function so as to prefer documents that contain multiple query terms, and boost their similarity score compared to documents that contain a smaller range of the query terms. We justify this approach by a retro-analysis of the *TREC* relevance judgements.

Experiments with a collection of web data and web queries shows that the new technique of impact transformation significantly boosts retrieval effectiveness. Moreover, the use of quantised values in the transformation allows extremely fast query processing.

2 Experimental regime

Use of an appropriate test regime is critical to an investigation of retrieval mechanisms. Table 1 lists the two experimental frameworks employed in this paper. Both make use of the *TREC* resources established by the National Institute of Standards and Technology in the United States [Harman, 1995, Hawking et al., 1999].

Each of the two experiments consists of a collection of unstructured *documents*; a set of queries; and a corresponding set of *relevance judgements* indicating which of the documents in the collection are relevant to each of the queries. In the case of *Exp.2*, we took the "Title" of each of the *TREC* topics as the query.

To compare the relative effectiveness of systems, an *effectiveness metric* is used. Of the many metrics that have been proposed [Salton, 1989], we believe that for the web environment – where users tend to peruse the first page of retrieved documents, but rarely move to the second and almost never look at the third – that *precision at 10 documents retrieved* (*Prec.10*), and *reciprocal rank of the first relevant document* (*Recp.Rank*) are the most meaningful

indicators. However, Buckley and Voorhees [2000] suggest that these indicators are relatively unstable, and recommend instead that for accurate assessment the *average precision over all relevant documents* (*Av.Prec*) should be employed. Hence in this study the latter, which is calculated based on the top $r = 1{,}000$ documents retrieved for each query (with precision taken to be zero for any relevant documents not retrieved within the top $r = 1{,}000$), is used as the primary effectiveness metric, with *Prec.10* and *Recp.Rank* used for corroboration. Note also that the use of these latter two metrics is less susceptible to the problem of relevant documents not being located [Cormack et al., 1998, Zobel, 1998]. All of there metrics take on values between zero and one, with a score of 1.0 indicating "perfection".

3 Vector-space similarity

The *vector space model* is widely used in information retrieval systems [Salton, 1989]. In this model, documents and queries are represented as bags of *terms*, and statistics concerning these terms and the documents they appear in are gathered together into an *index*. In the index, each distinct term t has an associated *document frequency*, denoted f_t, which indicates the number of documents it appears in. In addition, each term is associated with an *inverted list* of *pointers* $\langle d, f_{d,t} \rangle$ recording that term t appears in document d a total of $f_{d,t}$ times. Moreover, each document d has a corresponding value W_d associated with it, its *document length*, which is calculated as a function of f_t and $f_{d,t}$ for the terms t in that document. Generally speaking, W_d is greater when a document becomes physically longer, but W_d usually depends also upon the relative scarcity of the terms in the document.

In ranking a query q against the database, the vector space model employs a *similarity heuristic* to calculate a score $S_{q,d}$ between q and each document d of the database. For our purposes, $S_{q,d}$ can be described as

$$S_{q,d} = \sum_{t \in q \cap d} w_{d,t} \cdot w_{q,t} \qquad (1)$$

where the values of $w_{d,t}$ and $w_{q,t}$, called *term impacts* or simply *impacts* [Anh et al., 2001], represent the degree of "importance" of term t in document d and query q respectively, and are calculated from $f_{d,t}$, $f_{q,t}$, f_t, W_d, and W_q. It should be noted that we employ a common notation for *document impacts* and *query impacts* (that is, the impact values of document terms and query terms respectively) for simplicity, and that they can in fact have different formulations in terms of the underlying values f_t, $f_{d,t}$, $f_{q,t}$, W_d, and W_q.

It is clear that there is a huge number of different formulations for $S_{q,d}$ [Zobel and Moffat, 1998]. Nevertheless, in the notation of formula (1), $w_{d,t}$ is usually a product of functions monotonic in three factors: the document-term frequency $f_{d,t}$; the *inverse document frequency* $1/f_t$; and the *inverse document length* $1/W_d$. The combination of the first two factors has been frequently referred to as *TF* $\times$ *IDF* [Salton, 1989]. The third factor comprising $w_{d,t}$, which we similarly denote by *IDL*, acts to penalise long documents, which tend to have large values for $\sum$ *TF* $\cdot$ *IDF*.

Four possible cosine measures, labelled from *Cos.1* to *Cos.4* for later reference, are detailed in Table 2. In the nomenclature of Zobel and Moffat [1998], these four mechanisms are denoted as *BB-BBB-BBB*, *BD-ABB-BBB*, *BD-ACB-BCB*, and *BD-ACI-BCA*, respectively. The first measure is perhaps the simplest "standard" cosine measure; the last includes pivoted document length normalisation [Singhal et al., 1996] and is more complex, but performs best in the experiments undertaken by Zobel and Moffat. The two other measures in Table 2 serve as transitional steps from the first to the last.

Although we commenced this paper by noting the need for effective retrieval from search engines when faced with short queries, this need has been dominant only for the last half-dozen years. Early investigation into information retrieval techniques – including most of the *TREC* work in the first few years – centred on queries that are dramatically longer than those used by Internet surfers. That is, much of the prior research work is, explicitly or implicitly, based on the use of long queries for testing. Only recently have realistic web-sized test collections and web-sized test queries been available.

Hence our hypothesis – that the development of cosine similarity functions has been oriented towards improving effectiveness for long queries, and the resulting modifications may not be as applicable to short queries.

4 Short queries

We start with an investigation of cosine measure *Cos.1* in the context of *Exp.1*, and analyse the distribution of document impacts as follows. First, the maximal value U of the impacts is calculated, and a series of breakpoints whose value represent a portion of U are introduced. Then, for each breakpoint, the number of impacts that are smaller is counted. The resultant cumulative distribution is plotted in Figure 1. In *Exp.1* there are approximately 101 million document impact values, varying between $L \approx 4.6 \times 10^{-6}$ and $U \approx 1.0$.

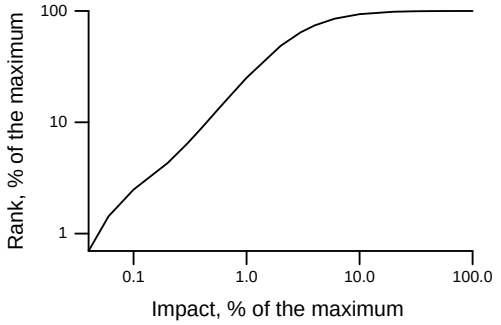

Figure 1: Distribution of impacts of *Cos.1* in *Exp.1*. For example, approximately 25% of the impact values in the collection are smaller than 0.01 (1% of the maximum impact score of 1.0).

The distribution graph shows that the majority of the impacts have low values. For example, 85% of the impacts are smaller than 5% of U, and 95% are less than 10%.

This skewed distribution does not create any particular problems for long queries. The score for a document is given by the sum of the impacts of the query terms, and with many query terms, no single term can dominate the scoring. Indeed, some retrieval systems even discard low (or potentially low) impacts to improve retrieval efficiency without loss on effectiveness. For example, Persin et al. [1996] successfully set a lower-bound threshold for the products of *TF* $\cdot$ *IDF* and discard all the values below that threshold. Note also that Moffat and Zobel [1996] found that pruning based upon *IDF* only did impact upon retrieval effectiveness, and that terms with low *IDF* val-

Metrics Label	Factors of $w_{d,t}$			Factors of $w_{q,t}$		
	TF	*IDF*	*IDL*	*TF*	*IDF*	*IQL*
Cos.1	$f_{d,t}$	$\log_e(1+N/f_t)$	$1/W_d$	$f_{q,t}$	$\log_e(1+N/f_t)$	$1/W_q$
Cos.2	$f_{d,t}$	1	$1/W_d$	$f_{q,t}$	$\log_e(1+f^m/f_t)$	$1/W_q$
Cos.3	$1+\log_e f_{d,t}$	1	$1/W_d$	$1+\log_e f_{q,t}$	$\log_e(1+f^m/f_t)$	$1/W_q$
Cos.4	$1+\log_e f_{d,t}$	1	$1/((1-s)+s\cdot W_d/W^a))$	$1+\log_e f_{q,t}$	$\log_e(1+f^m/f_t)$	1

Table 2: Elements of some similarity heuristics based upon the cosine rule. Here W_x (x can be either d or q) is the document or query length and calculated as $\sqrt{(\sum_{t\in x}(TF\cdot IDF)^2)}$; W^a is the average value of W_d over the documents in the collection; f^m is the greatest value of f_t in the collection; and s is a constant with typical value of 0.7 [Singhal et al., 1996]. Each of the four listed similarity computations is calculated as the summation over the terms common to the query and the document of the product of the six listed factors.

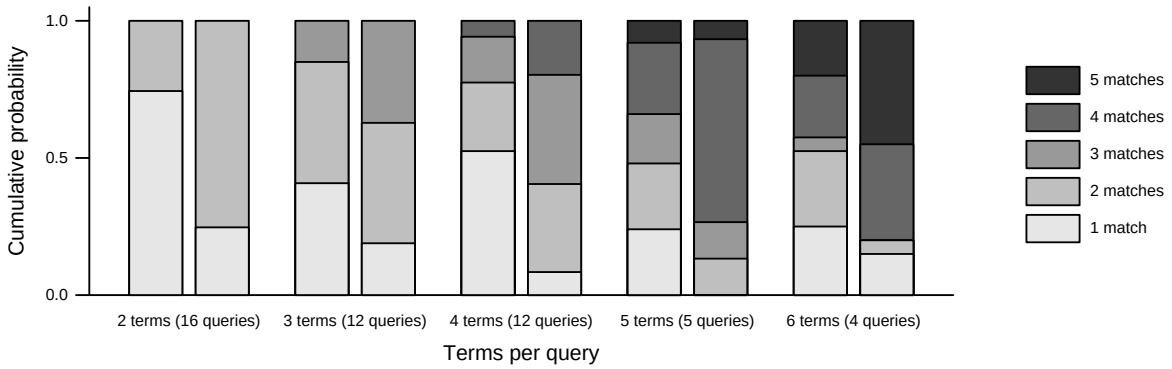

Figure 2: Probability of a document being retrieved compared with its probability of being relevant, categorised as a function of the number of terms in the query. The left bar in each pair shows the distribution of those query terms in the documents retrieved from *Exp.1* using *Cos.1*, while the second bar shows the distribution of the query terms in the documents judged to be relevant by the *TREC* assessors. Note that none of the six-term queries corresponded to relevant documents that matched the query in either six or three terms.

ues should be allowed to permute the selected documents, even if they did not select any.

On the other hand, the skew distribution of impact values creates problems for short queries. For example, consider a query of just two terms. Intuitively we would presume that a document containing both of those terms, even if just once each, should probably rank ahead of documents containing only one of the terms, even if many times. But this is not what vector-space mechanisms actually do – in these mechanisms a single appearance of one of the query terms in a short document can create a high impact value, as can multiple appearances of one of the terms in a document of more typical length.

To draw out this effect, we studied the *TREC* relevance judgements supplied for *Exp.1*, and compared them to the documents selected by *Cos.1* for those queries. First, the queries were partitioned into sets determined by the number of terms in each, with the single query containing only one term excluded. The remaining 49 queries were then classified by their length, ranging from 2 terms (16 queries) to 6 terms (4 queries). For the queries in each group the relevance judgements were used to determine the set of relevant documents from the collection, and those documents examined to determine the distribution of query terms in them. From this data it was then possible to calculate the probability that an "average" relevant document has a certain number of matches to its corresponding query.

We also separately processed the queries using *Cos.1*, retrieving the top 10 ranked documents for each query. These top 10 documents were then examined in the same way, to calculate the probability that an "average" returned document matches its query in a certain number of terms.

Figure 2 plots the outcomes, and confirms our suspicion that the actual scoring mechanism of *Cos.1* places an inordinate importance upon documents matching the query in relatively few of the query terms, and that the relevant documents contain a disproportionate (at least, according to the scoring mechanism) number of the query terms. For example, in the two-term queries there is a quite large probability that a retrieved document shares only one common term with the query, but the majority of the relevant documents in fact share two terms with it. Indeed, the shorter the query, the greater the apparent difference between the retrieved and relevant distributions.

5 Normalisation

Figure 2 suggests that the cosine measure overstates the role of high impacts and dampens the collective role of the lower ones. That is, the scoring heuristic probably needs to somehow moderate high impact values and boost low impact values. That is, some kind of impact normalisation is required.

Normalisation is not a new technique in information retrieval, and there have been several successful attempts to normalise individual factors of the impacts. The employment of *IDL* mentioned in Section 1 is one example, and there are many others. Indeed, the transition from *Cos.1* to *Cos.2* (in Table 2) is brought about by the normalisation of the *TF* factor, and from *Cos.2* to *Cos.3* by the normalisation of the *IDF* factor.

Further, the SMART system normalises the *TF* factor to $(0.5+0.5\cdot TF/\max(TF))$ as a replacement for both *TF* and *IDL* factors. A similar approach is employed by the INQUERY system with a slightly different formula of

$(0.4 + 0.6 \cdot TF/\max(TF))$ [Callan et al., 1992, Salton and Buckley, 1988]. These normalisations turn to be in favour of long documents [Broglio et al., 1994]. The OKAPI system uses another kind of normalisation, namely based on the length in bytes of documents [Robertson et al., 1994].

In a similar vein, Singhal et al. exploit the fact that the cosine measure tends to be biased in favour of short documents. They compare the probability of documents being relevant with the probability of them being retrieved, for different groups of documents parametrised by document length W_d, and conclude that it would be better to normalise W_d by boosting low values. They then introduce *pivoted document length normalisation* – a technique in which the document length W_d is normalised to $(1-s)+s \cdot W_d/W^a$. This is the technique listed as *Cos.4* in Table 2. The constant s is referred to as *slope*, with the typical value of 0.7, largely independent of the collection; and so a document of raw length close to zero will have pivoted length of $1-s$; a document of length W^a will have a pivoted length of 1; and a document of length $2W^a$ has pivoted length $1+s$. It is interesting to note that the normalisations of the SMART and the INQUERY systems can also be classified as pivoted, with the pivot point taken as the maximum value f^m.

In their experiments, Zobel and Moffat [1998] determined that *Cos.4*, with pivoted document normalisation, performs well compared to other similarity heuristics. This measure includes most of the previous normalisation techniques, and in fact all three factors of document impacts have been separately normalised.

We do not attempt to normalise further any of the individual factors. Rather, we address the problem directly by normalising the complete impact value arising from those three factors, regardless of whether the factors have been pre-normalised. That is, whatever the formulation chosen as the similarity measure, the impacts can be considered to be a set of independent values and are normalised to dampen the high values and promote the low ones.

With this purpose, pivoted normalisation with the maximum impact as the pivot value is an obvious candidate. In this case, all the impacts, except the maximum value, are promoted to (relatively) higher values. Also, the smaller the impact, the greater the relative gain.

But while this kind of pivoted normalisation satisfies our criteria, there is at least one drawback – the fact that the smallest impacts are promoted by the highest absolute amounts directly contradicts the long-held belief in information retrieval that very small impacts have no value in retrieval, and should usually be considered to be noise.

To avoid this difficulty we use a normalisation scheme that has two fixpoints: one at the minimum value L, and the other at the maximum value U. To fit to our criteria, the scheme needs to allocate large relative increases to the small values, and small relative increases to the large impacts. There are a number of ways to deploy the required normalisation. One, which we denote as *two-fixpoint normalisation*, is included in the listing below.

As a counter point to our exploration of impact normalisation, we also included a third mechanism in our experiments – a *demotion* mechanism that reduced small impact values by a large amount, and large impact values by a small amount.

In detail, the methods we explored are:

1. *Two-fixpoint promotion*: All impacts are increased, with higher relative increments for small values, but with the minimum and maximum values unchanged. The normalised shape is a logarithmic curve. The normalised value of w is $L + L \cdot \log_B(w/L)$ where $B = (U/L)^{L/(U-L)}$.

2. *One-fixpoint promotion*: The same as pivoted normalisation with the maximum impact as the fixpoint. Recall that for this method a slope value s must be chosen.

3. *Demotion*: This method is introduced as a complement of the two-fixpoint promotion method. It has an exponential curve, normalising w to $L \cdot B^{(w-L)/L}$ where all constants are defined as in the promotion case.

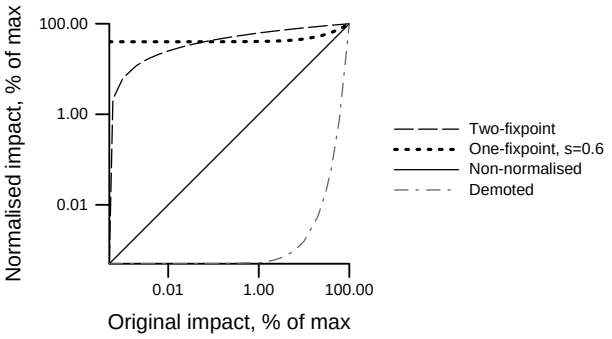

Figure 3: Behaviour of different normalisation methods compared to original impact. The graph shows the transformations used for *Exp.1* and the cosine measure *Cos.1*.

Method	*Recp.Rank*	*Prec.10*	*Av.Prec*
Standard	100	100	100
Two-fixpoint	161	162	203
One-fixpoint	153	168	196
Demoted	83	97	94

Table 3: Relative effectiveness of normalised impacts, as a percentage of the baseline unnormalised score for each of three underlying effectiveness metrics, using *Cos.1* on *Exp.1*. The unnormalised *Recp.Rank*, *Prec.10* and *Av.Prec* values are 0.3480, 0.1500 and 0.1369 respectively.

The behaviour of these normalisation methods is illustrated in Figure 3 for the chosen resource and cosine measure. In this graph, one-fixpoint promotion has been plotted with slope $s = 0.6$.

To see the effect of normalisation upon retrieval effectiveness, experiments were conducted for *Exp.1* and *Cos.1*. The result is shown in Table 3. The one-fixpoint normalisation was done with slope $s = 0.6$ – the value which gave the best performance in preliminary tests.

In general the results in Table 3 confirm our hypothesis. The two-fixpoint method outperforms the one-fixpoint method in two of the three evaluation metrics, but not the third. On the other hand, both are considerably better than the unnormalised computation, and the *demotion* method is worse than doing nothing.

Similar results have also been obtained for a slightly different version, where the value of *L* and *U* are selected to exclude 0.1% of the impact population at each extreme end of the range of values. This possibility emerged from the fact that there are very few values located at the long tail of the distribution curve, and hence it can be argued that the use of the original maximal value *U* for determining the normalisation parameter might not be representative. Marginally better effectiveness results were obtained.

In the experiments below we make use of two-fixpoint promotion (using the original U and L), partly because, of the two promotion methods tested, it yielded the higher *Av.Prec* score, but also because there is no need to fix a parameter s.

6 Quantisation

Before committing fully to impact transformation we must also consider the issue of efficiency. Some non-trivial amount of time might be needed to normalise each of the impacts, making query processing more expensive. One obvious way to ameliorate this cost is to pre-calculate all normalised impacts and store them in the inverted list instead of within-document frequencies $f_{d,t}$. This saves on later computation costs, but adds – possibly considerably – to the space required to store the index.

Anh et al. [2001] sidestep the problem of index space by approximating the impact values by small integers, each in b bits for some small value b. Anh et al. further *impact sort* each of the index lists according to these quantised values. Sorting ensures that each index list is composed of not more than 2^b segments, with each segment containing pointers of approximately equal impact value, sorted by document number. During querying, as index list segments are selected for processing, the inverse transformation is applied, and (approximate) inverse impact scores accumulated as similarity contributions. Note that when impact values are stored in the index, there is no need for subsequent normalisation by W_d values; and pruning techniques of the type described by [Persin et al., 1996] can be applied [Anh et al., 2001].

One consequence of normalisation – which aims to narrow the gap between moderately-small and moderately-large impact values – is that it might be possible to employ the quantised integer values to represent the impacts themselves, and use those integers directly in calculation of similarity scores without referring to the original or normalised impacts.

To demonstrate this point, we experiment with some small values of b. The normalised impacts are quantised into linear buckets over the interval $(0, 2^b)$ by division and truncation. That is, a normalised impact w is quantised to the value $\lfloor 2^b \cdot w/(U + \epsilon) \rfloor$, where ϵ is used to force the value U to map to $2^b - 1$.

We have a range of possibilities for the inverse transformation. Of these, three have been considered in our experiments:

1. *GroupNo*: With buckets numbered from 0 to $2^b - 1$, the bucket number itself is used as the inverse impact value and accumulated against the similarity score for corresponding documents.

2. *NextGroup*: A value one greater than the bucket number is used as the inverse impact value, to avoid the zero quantised values of the *GroupNo* approach.

3. *Middle*: In the non-transformed scheme of Anh et al. [2001], the arithmetic mid-point of the range spanned by each original bucket is taken as the inverse value for the values quantised into that bucket. Because Anh et al. work with a non-uniform quantisation into buckets, the inverse transform cannot be described as a simple set of integer values.

Figure 4 shows the functional composition of the two-fixpoint normalisation mechanism and the *GroupNo* quantisation mechanism with $b = 3$.

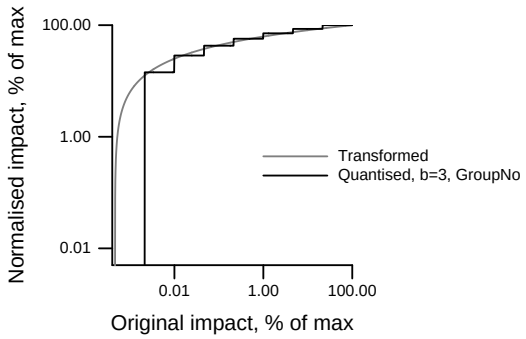

Figure 4: Final transformation of documents impacts, showing two-fixpoint promotion and then quantisation using $b = 3$, with inverse transform accomplished using *GroupNo* mechanism. The displayed data refers to *Exp.1* and *Cos.1*.

Retrieval effectiveness as a function of b, the number of bits used for the approximations, is plotted in Figure 5 for both transformed and non-transformed impacts, using *Cos.1* applied to *Exp.1*, and taking *Av.Prec* as the effectiveness metric. Even with values of b as small as four – and hence with just 16 different impact levels – retrieval effectiveness is good. Moreover, there is little apparent difference between the *GroupNo* and *NextGroup* reverse transformations. Note that one advantage of the former compared with the latter is that the index can be made smaller, since the segment in each index list corresponding to bucket 0 need not be stored (unless Boolean queries must also be supported), as there is no net effect when adding an impact of zero. But even without this saving, with $b = 5$ the cost of storing quantised impact values in an impact sorted index is small [Anh et al., 2001].

The bottom two lines in Figure 5 show the retrieval effectiveness attained by the mechanism described by Anh et al.Again, effectiveness converges to its final value for small values of b. But the final effectiveness value is markedly smaller than that achieved when the impacts are transformed, further confirmation of the gains accruing to the impact transformation described in Section 5.

In the further experiments described below, we fix $b = 5$ and use the *GroupNo* inverse transformation.

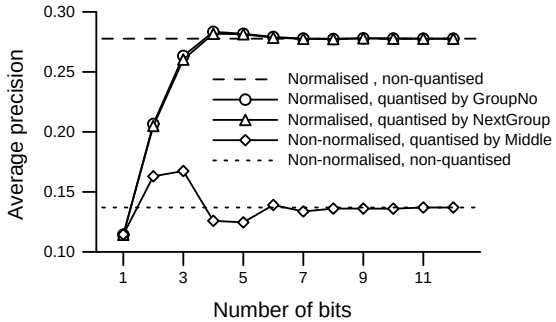

Figure 5: Impact of quantisation: *Av.Prec* as a function of the number of bits b used in the quantisation process, for situations with and without impact normalisation, and using three different inverse quantisation strategies. All results are for *Exp.1* and *Cos.1*.

7 Query impact transformation

The retrieval effectiveness results presented in the previous section presumed that whatever transformation was applied to the document impacts should also be applied to query impacts. That is, the normalisation and quantisation formulae for the document collection were fixed prior to any query being considered, and then those formulae used for both the document impacts and query impacts, with the latter clipped to ensure that they were within the defined range of the document impact transformation. Hence, the final similarity score for a document in these experiments was the sum of the products of integer-valued document impacts and integer-valued query impacts.

But we have no especial basis for presuming that query impacts should be treated identically to document impacts. It might be that query impacts are best used unnormalised, or unquantised, or both.

Table 4 evaluates these alternatives, using *Av.Prec* as a retrieval effectiveness metric, for *Cos.1* and *Exp.1*. The values in the table are percentages relative to the "both document and query impacts transformed, and both query and document impacts quantised" presumed in the results of the previous section. In all cases, the document impacts are normalised, and quantised using the *GroupNo* inverse transformation and $b = 5$.

As can be seen from the table, the presence or absence of query impact normalisation or quantisation has little further impact upon the effectiveness of the retrieval process, and we can allow efficiency concerns to influence the choice of mechanism. In particular, quantisation using the *GroupNo* mechanism (which can generate query impacts of zero) may allow whole query terms to be dispensed with, thereby saving computation time.

8 Experimental results

The results of the previous sections – with experiments on *Exp.1* using *Cos.1* – have made it clear that term impact transformation is a powerful technique, and achieves a dramatic improvement in retrieval effectiveness compared to an untransformed computation.

One obvious question to address is the extent to which this improved performance applies to vector space formulations in which one or more of the factors is already transformed or normalised. That is, to what extent have we simply found an alternative way of boosting the effectiveness of an inferior retrieval mechanism.

Figure 6 dispels this concern. For the four different vector space formulations given in Table 2, the figure shows the "before" and "after" effect of impact normalisation and quantisation. Pleasingly, there is a marked improvement in all four mechanisms, no matter how effectiveness is measured.

Query impact normalisation	Query impact quantisation		
	None	*GroupNo*	*NextGroup*
None	102	n/a	n/a
Two-fixpoint	100	100	99
One-fixpoint	98	98	98
Demoted	99	97	101

Table 4: Different combinations of normalisation and quantisation for query impacts, scored using *Av.Prec* as a percentage of the base effectiveness obtained when both document and query impacts are two-fixpoint promoted, and then both quantised by *GroupNo* with $b = 5$, using *Cos.1* on *Exp.1*. In all cases document impacts were two-fixpoint promoted, and quantised with $b = 5$ using *GroupNo*.

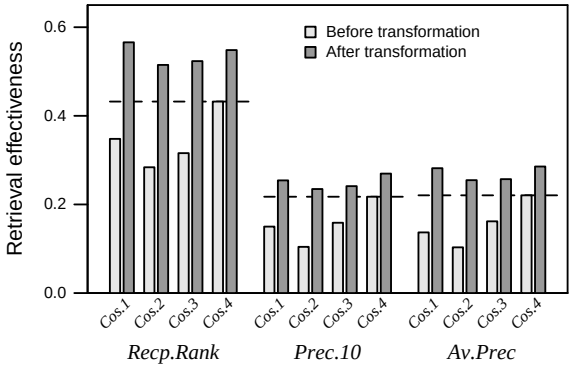

Figure 6: Effect of transformation on different cosine measures for the resource *Exp.1*. The dotted horizontal lines show the "before transformation" effectiveness attained by *Cos.4* for that metric.

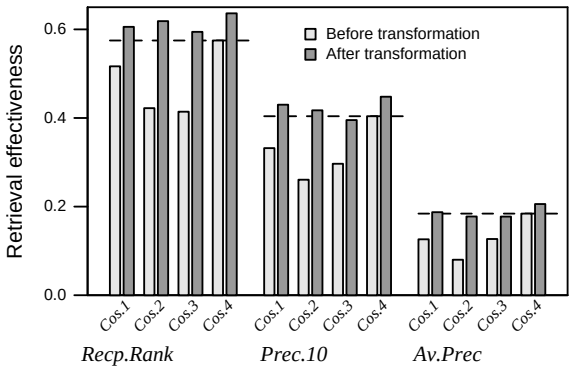

Figure 7: Effect of transformation on different cosine measures for the resource *Exp.2*. The dotted horizontal lines show the "before transformation" effectiveness attained by *Cos.4* for that metric.

Even more interesting is that, although the effectiveness (in terms of all the three indicators) of the untransformed impacts vary over a wide range when different measures are employed, it is not true for the transformed impacts. That is, the selection of a particular cosine measure has little effect on the final absolute outcome once impact transformation has been applied.

The second interesting outcome is that despite the more modest relative improvement for *Cos.4*, the result is still material. For this "competitive" method the improvement on *Recp.Rank*, *Prec.10* and *Av.Prec* are approximately 31%, 24% and 30% respectively when "before" and "after" are compared.

To further test the robustness of the transformation technique, we applied it to the resource *Exp.2*, in which the term statistics are quite different, the queries are slightly longer, and measure *Cos.4* gives better baseline effectiveness than it does in *Exp.1* for two of the three effectiveness metrics. Figure 7 plots the outcomes.

It is clear that for this second set of experiments the advantage of the transformation is retained, even though the gain is not of the same magnitude as for the previous experiments using *Exp.1*. On this second collection and query set the improvement on *Recp.Rank*, *Prec.10* and

Av.Prec for *Cos.4* is 11%, 11%, and 12% respectively. For both sets of experiments the transformation on impacts gives retrieval effectiveness not worse than that of the untransformed *Cos.4*, shown by the horizontal dotted line in Figures 6 and 7.

9 Future directions

Impact transformation provides considerably improved retrieval effectiveness, particularly for web data and web-style short queries.

The transformation is composed of two stages. The first is an impact normalisation process, which boosts low – but not small – impact scores. The second stage is then a quantisation of normalised values to a small set of integers used in all subsequent computations. The first stage is responsible for the gain in retrieval effectiveness; the second provides for fast execution and efficient use of disk space [Anh et al., 2001]. Indeed, while this paper has concentrated solely upon retrieval effectiveness, retrieval efficiency has not been far from our thoughts, and experiments to validate our expectations as to speed are currently being planned, both with and without the use of pruning heuristics.

We still have several avenues to explore in search of better, or more consistently good, retrieval effectiveness.

One outstanding question is about the applicability of the method to long queries. We expect that direct application of the method to long queries (for example, to the resource *Exp.3* used as a third experimental environment by Anh et al.) would not yield the same improvement, since a relatively modest number of originally small impacts might dominate over a smaller number of originally high impacts. Hence, schemes in which the "amount" of transformation applied is governed by the length of the query are probably also worth investigating. When dealing with long queries we must also verify that pruning techniques – when some or all of the pointers in some or all of the index lists are not processed – can be applied without loss of retrieval effectiveness, since it is the use of pruning that allows long queries to be executed quickly.

Another area for further investigation is the possibility of undertaking local transformations. In the present work, the transformation parameters are evaluated and applied globally. But these parameters might also be defined locally over a single inverted list, or a cluster of lists.

Another question that must be resolved before we can conclude our investigation is this: we have improved four vector space mechanisms, but in *TREC* terms, those four scoring mechanisms are not especially competitive. That is, we must also compare our best method against *TREC*-best methods in appropriate experiments, and possibly also add impact transformation to implementations of *TREC*-best methods to see if transformation remains useful, or is merely an alternative enhancement that allows a similar effectiveness gain.

Last, but not least, we make no claim that the normalisation functions described above are in any way optimal. In our experiments both the one-fixpoint and two-fixpoint methods worked similarly well, but it is clear from Figure 3 that both transformations are quite aggressive, and there are plenty of milder functions with the requisite general properties that might sensibly be experimented with.

Acknowledgement This work was supported by the Victorian Partnership for Advanced Computing.

References

V. N. Anh, O. de Kretser, and A. Moffat. Vector-space ranking with effective early termination. In *Proc. 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, LA, Sept. 2001. ACM Press, New York. To appear.

J. Broglio, J. P. Callan, W. B. Croft, and D. W. Nachbar. Document retrieval and routing using the INQUERY system. In Harman [1994], pages 29–38.

C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In N. J. Belkin, P. Ingwersen, and M.-K. Leong, editors, *Proc. 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40, Athens, Greece, Sept. 2000. ACM Press, New York.

J. P. Callan, W. B. Croft, and S. M. Harding. The INQUERY retrieval system. In *Proc. Int. Conf. Database and Expert Systems Applications*, pages 78–83, Dublin, Ireland, 1992. Boole Press.

G. V. Cormack, C. R. Palmer, and C. L. A. Clarke. Efficient construction of large test collections. In Croft et al. [1998], pages 282–289.

W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors. *Proc. 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, Aug. 1998. ACM Press, New York.

D. K. Harman, editor. *Proc. Third Text REtrieval Conference (TREC–3)*, Gaithersburg, MD, Nov. 1994. National Institute of Standards and Technology Special Publication 500-225.

D. K. Harman. Overview of the second text retrieval conference (TREC-2). *Information Processing & Management*, 31(3):271–289, May 1995.

D. Hawking, E. Voorhees, N. Craswell, and P. Bailey. Overview of the TREC-8 web track. In E. M. Voorhees and D. K. Harman, editors, *Proc. Eighth Text REtrieval Conference (TREC–8)*, Gaithersburg, MD, Nov. 1999. National Institute of Standards and Technology Special Publication 500-246.

B. P. Jansen, A. Spink, J. Bateman, and T. Saracevic. Real life information retrieval: A study of user queries on the web. *ACM SIGIR Forum*, 32(1):5–17, Spring 1998.

A. Moffat and J. Zobel. Self-indexing inverted files for fast text retrieval. *ACM Transactions on Information Systems*, 14(4):349–379, Oct. 1996.

M. Persin, J. Zobel, and R. Sacks-Davis. Filtered document retrieval with frequency-sorted indexes. *Journal of the American Society for Information Science*, 47(10):749–764, Oct. 1996.

S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC–3. In Harman [1994].

G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, Massachusetts, 1989.

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In H.-P. Frei, D. Harman, P. Schäuble, and R. Wilkinson, editors, *Proc. 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29. ACM Press, New York, Aug. 1996.

J. Zobel. How reliable are the results of large-scale information retrieval experiments? In Croft et al. [1998], pages 307–314.

J. Zobel and A. Moffat. Exploring the similarity space. *ACM SIGIR Forum*, 32(1):18–34, Spring 1998.