TERM-WEIGHTING APPROACHES
IN AUTOMATIC TEXT RETRIEVAL

GERARD SALTON and CHRISTOPHER BUCKLEY
Department of Computer Science, Cornell University, Ithaca, NY 14853, USA

(Received 19 November 1987; accepted in final form 26 January 1988)

Abstract—The experimental evidence accumulated over the past 20 years indicates that
text indexing systems based on the assignment of appropriately weighted single terms
produce retrieval results that are superior to those obtainable with other more elaborate
text representations. These results depend crucially on the choice of effective term-
weighting systems. This article summarizes the insights gained in automatic term weight-
ing, and provides baseline single-term-indexing models with which other more elaborate
content analysis procedures can be compared.

1. AUTOMATIC TEXT ANALYSIS

In the late 1950s, Luhn [1] first suggested that automatic text retrieval systems could be
designed based on a comparison of content identifiers attached both to the stored texts and
to the users' information queries. Typically, certain words extracted from the texts of
documents and queries would be used for content identification; alternatively, the content
representations could be chosen manually by trained indexers familiar with the subject
areas under consideration and with the contents of the document collections. In either case,
the documents would be represented by term vectors of the form

\[ D = (t_1, t_2, \ldots, t_p) \]  

where each \( t_i \) identifies a content term assigned to some sample document \( D \). Analog-
ously, the information requests, or queries, would be represented either in vector form,
or in the form of Boolean statements. Thus, a typical query \( Q \) might be formulated as

\[ Q = (q_1, q_2, \ldots, q_q) \]  

or

\[ Q = (q_1, q_2, \ldots, q_q) \text{ or } (q_1, q_2, \ldots, q_q) \text{ or } \ldots \]  

where \( q_i \) once again represents a term assigned to query \( Q \).

A more formal representation of the term vectors of eqns (1) and (2) is obtained by
including in each vector all possible content terms allowed in the system and adding term
weight assignments to provide distinctions among the terms. Thus, if \( w_{tk} \) (or \( w_{qk} \)) re-
prents the weight of term \( t_k \) in document \( D \) (or query \( Q \)), and \( t \) terms in all are available
for content representation, the term vectors for document \( D \) and query \( Q \) can be written as

\[ D = (t_1, w_{10}; t_1, w_{11}; \ldots; t_1, w_{1t}) \]  

and

\[ Q = (q_1, w_{q0}; q_1, w_{q1}; \ldots; q_1, w_{qt}). \]  

In the foregoing formulation, the assumption is that \( w_{tk} \) (or \( w_{qk} \)) is equal to 0 when term
\( k \) is not assigned to document \( D \) (or query \( Q \)), and that \( w_{tk} \) (or \( w_{qk} \)) equals 1 for the
assigned terms.

Given the vector representations of eqn (4), a query-document similarity value may be
obtained by comparing the corresponding vectors, using for example the conventional
vector product formula

\[ \text{similarity}(Q, D) = \sum_{k=1}^{t} w_{qk} \cdot w_{tk}. \]  

(5)

When the term weights are restricted to 0 and 1 as previously suggested, the vector prod-
uct of eqn (5) measures the number of terms that are jointly assigned to query \( Q \) and docu-
ment \( D \).

In practice, it has proven useful to provide a greater degree of discrimination among
terms assigned for content representation than is possible with weights of 0 and 1 alone.
In particular, term weights in decreasing term importance order could be assigned, in
which case the weights \( w_{tk} \) (or \( w_{qk} \)) could be allowed to vary continuously between 0
and 1, the higher weight assignments near 1 being used for the most important terms,
whereas lower weights near 0 would characterize the less important terms. In some cir-
cumstances, it may also be useful to use normalized weight assignments, where the indi-
vidual term weights depend to some extent on the weights of other terms in the same
vector. A typical term weight using a vector length normalization factor is

\[ w_{tk} = \frac{w_{tk}}{\sqrt{\sum_{v} (w_{vk})^2}} \]  

for documents \( \frac{w_{tk}}{\sum_{v} (w_{vk})^2} \) for queries. When a length normalized term-weight-
ing system is used with the vector similarity function of eqn (5), one obtains the well-known
cosine vector similarity formula that has been used extensively with the experimental Smart
retrieval system [2,3]:

\[ \text{similarity}(Q, D) = \frac{\sum_{k=1}^{t} w_{qk} \cdot w_{tk}}{\sqrt{\sum_{k=1}^{t} (w_{qk})^2 \cdot \sum_{k=1}^{t} (w_{tk})^2}}. \]  

(6)

A vector matching system performing global comparisons between query and docu-
ment vectors provides ranked retrieval output in decreasing order of the computed similar-
ities between \( Q \) and \( D \). Such a ranked output is useful because controls are now available
over the size of the retrieved document set, and iterative retrieval strategies based on suc-
cessive query reformulations are simplified. A system that first retrieves those items
thought to be of main interest to the users will necessarily prove helpful in interactive infor-
mation retrieval.

In designing automatic text retrieval systems, two main questions must be faced. First,
what appropriate content units are to be included in the document and query representa-
tions? Second, is the determination of the term weights capable of distinguishing the
important terms from those less crucial for content identification?

Concerning first the choice of content terms, various possibilities must be considered.
In most of the early experiments, single terms alone were used for content representation,
often consisting of words extracted from the texts of documents and from natural language
query formulations. [3–7] In many cases, quite effective retrieval output has been obtained
using single-term content representations. Ultimately, however, sets of single terms can
not provide complete identifications of document content. For this reason, many enhance-
ments in content analysis and text indexing procedures have been proposed over the years.
in an effort to generate complex text representations. The following possibilities have been considered in this connection:

1. The generation of sets of related terms based on the statistical cooccurrence characteristics of the words in certain contexts within the document collection. The assumption normally made is that words that cooccur with sufficient frequency in the documents of a collection are in fact related to each other [8–11].

2. The formation of term phrases consisting of one or more governing terms (the phrase heads) together with corresponding dependent terms (the phrase components). Phrases are often chosen by using word frequency counts and other statistical methods, possibly supplemented by syntactic procedures designed to detect syntactic relationships between governing and dependent phrase components [12–17].

3. The use of word grouping methods of the kind provided by thesauruses, where classes of related words are grouped under common headings; these class headings can then be assigned for content identification instead of the individual terms contained in the classes [18–20]. Alternatively, term relationships useful for content identification may also be obtainable by using existing machine-readable dictionaries and lexicons [21–24].

4. The construction of knowledge bases and related artificial intelligence structures designed to represent the content of the subject area under consideration; entries from the knowledge base are then used to represent the content of documents and queries [25–30].

From the beginning, it was evident that the construction and identification of complex text representations was inordinately difficult. In particular, it became clear that most automatically derived term dependencies were valid only locally in the documents from which the dependent term groups were originally extracted; this implies that dependent term groups could not be counted upon to produce useful content identifiers in new document contexts different from those originally used [11]. The experiences gained with the use of automatically generated term phrases proved similarly discouraging: for some collections, improvements in retrieval effectiveness of up to 20% (in search recall and precision) were obtainable by using phrase identifiers instead of single terms; but for other collections these same phrase procedures did not furnish any improvements at all. Moreover, even more sophisticated syntactic analysis programs could not be relied upon to produce useful complex content identifiers [16].

As for the use of preconstructed vocabulary schedules and term classifications, the problem is that viable procedures for the construction of effective vocabulary tools covering subject areas of reasonable scope appear to be completely lacking. The same goes for the construction of knowledge bases designed to reflect the structure of disclosure areas. Until more becomes known about the desired form and content of dictionaries and thesauruses, little gain should be expected from these tools in text analysis and document indexing.

In reviewing the extensive literature accumulated during the past 25 years in the area of retrieval system evaluation, the overwhelming evidence is that the judicious use of single-term identifiers is preferable to the incorporation of more complex entities extracted from the texts themselves or obtained from available vocabulary schedules [31–37]. Two main problems appear in producing complex text identifiers:

1. When stringent conditions are used for the construction of complex identifiers, typified by the use of restrictive frequency criteria and limited cooccurrence contexts for the recognition of term phrases, then few new identifiers are likely to become available, and the performance of the retrieval system with complex identifiers will differ only marginally from the results obtainable with single term indexing.

2. On the other hand, when the construction criteria for the complex entities are relaxed, then some good identifiers are obtained, but also many marginal ones that do not prove useful. Overall, the single-term indexing will generally be preferred.

When single terms are used for content identification, distinctions must be introduced between individual terms, based on their presumed value as document descriptors. This leads to the use of term weights attached to the item identifiers. The considerations controlling the generation of effective weighting factors are outlined briefly in the next section.

2. TERM-WEIGHT SPECIFICATION

The main function of a term-weighting system is the enhancement of retrieval effectiveness. Effective retrieval depends on two main factors: one, items likely to be relevant to the user's needs must be retrieved; two, items likely to be extraneous must be rejected. Two measures are normally used to assess the ability of a system to retrieve the relevant and reject the nonrelevant items of a collection, known as recall and precision, respectively. Recall is the proportion of relevant items retrieved, measured by the ratio of the number of relevant retrieved items to the total number of relevant items in the collection; precision, on the other hand, is the proportion of retrieved items that are relevant, measured by the ratio of the number of relevant retrieved items to the total number of retrieved items.

In principle, a system is preferred that produces both high recall by retrieving everything that is relevant, and also high precision by rejecting all items that are extraneous. The recall function of retrieval appears to be best served by using broad, high-frequency terms that occur in many documents of the collection. Such terms may be expected to pull out many documents, including many of the relevant documents. The precision factor, however, may be best served by using narrow, highly specific terms that are capable of isolating the few relevant items from the mass of nonrelevant ones. In practice, compromises are normally made by using terms that are broad enough to achieve a reasonable recall level without at the same time producing unreasonably low precision.

The differing recall and precision requirements favor the use of composite term weighting factors that contain both recall- and precision-enhancing components. Three main considerations appear important in this connection. First, terms that are frequently mentioned in individual documents, or document excerpts, appear to be useful as recall-enhancing devices. This suggests that a term frequency (tf) factor be used as part of the term-weighting system measuring the frequency of occurrence of the terms in the document or query texts. Term-frequency weights have been used for many years in automatic indexing environments [1–4].

Second, term frequency factors alone cannot ensure acceptable retrieval performance. Specifically, when the high frequency terms are not concentrated in a few particular documents, but instead are prevalent in the whole collection, all documents tend to be retrieved, and this affects the search precision. Hence a new collection-dependent factor must be introduced that favors terms concentrated in a few documents of a collection. The well-known inverse document frequency (idf) (or inverse collection frequency) factor performs this function. The idf factor varies inversely with the number of documents to which a term is assigned in a collection of N documents. A typical idf factor may be computed as \(\log N/n \) [38].

Term discrimination considerations suggest that the best terms for document content identification are those able to distinguish certain individual documents from the remainder of the collection. This implies that the best terms should have high term frequencies but low overall collection frequencies. A reasonable measure of term importance may then be obtained by using the product of the term frequency and the inverse document frequency (tf \(\times\) idf) [39–41].

The term discrimination model has been criticized because it does not exhibit well substantiated theoretical properties. This is in contrast with the probabilistic model of information retrieval where the relevance properties of the documents are taken into account, and a theoretically valid term relevance weight is derived [42–44]. The term relevance
weight—defined as the proportion of relevant documents in which a term occurs divided by the proportion of nonrelevant items in which the term occurs—is, however, not immediately computable without knowledge of the occurrence properties of the terms in the relevant and nonrelevant parts of the document collection. A number of methods have been proposed for estimating the term relevance factor in the absence of complete relevance information, and these have shown that under well-defined conditions the term relevance can be reduced to an inverse document frequency factor of the form log ((N - n)/n) [45-46]. The composite (tf x idf) term-weighting system is thus directly relatable to other theoretically attractive retrieval models.

A third term-weighting factor, in addition to the term frequency and the inverse document frequency, appears useful in systems with widely varying vector lengths. In many situations, short documents tend to be represented by short-term vectors, whereas much larger-term sets are assigned to the longer documents. When a large number of terms are used for document representation, the chance of term matches between queries and documents is high, and hence the longer documents have a better chance of being retrieved than the short ones. Normally, all relevant documents should be treated as equally important for retrieval purposes. This suggests that a normalization factor be incorporated into the term-weighting formula to equalize the length of the document vectors. Assuming that w represents the weight of term t, the final term weight might then be defined as \[ w' = \frac{w}{\sqrt{\sum \text{wt}}}, \]

In the preceding discussion of term-weighting systems, both documents and queries were assumed to be represented by sets, or vectors, of weighted terms. Term-weighting systems have also been applied to Boolean query statements, and extended Boolean systems have been devised in which Boolean query statements are effectively reduced to vector form [47-54]. The previous considerations regarding term weighting thus apply to some extent also to Boolean query processing.

3. TERM-WEIGHTING EXPERIMENTS

A number of term-weighting experiments are described in the remainder of this note in which combinations of term frequency, collection frequency, and length normalization components are used with six document collections of varying size, covering different subject areas. In each case, collections of user queries are used for retrieval purposes and the performance is averaged over the number of available user queries. For each experiment, the average search precision is computed for three different recall points, including a low recall of 0.25, an average recall of 0.50, and a high recall of 0.75. This average search precision is then further averaged for all available user queries. In addition, to the precision measure, the rank of the weighting methods in decreasing performance order is used as an evaluation criterion. A total of 1800 different combinations of term-weight assignments were used experimentally, of which 287 were found to be distinct. A rank of 1 thus designates the best performance, and 287 the worst.

In the present experiments, each term-weight combination is described by using two triples, representing, respectively, the term frequency, collection frequency, and vector normalization factors for document terms (first triple), and query terms (second triple). The principal weighting components are defined in Table 1. Three different term-frequency components are used, including a binary weight (b), the normal term frequency (t), and a normalized term frequency (n) that lies between 0.5 and 1.0. The three collection frequency components represent multipliers of (x) that disregards the collection frequency, a conventional inverse collection frequency factor (c), and a probabilistic inverse collection frequency factor (p). Finally, the length normalization factor may be absent (x as the third component) or present (c). (In the previously mentioned full set of 1800 different term-weight assignments, additional weighting components not included in Table 1 were also tried. These additional components did not supply any fundamentally new insights or advantages.)

Table 2 shows actual formulas for some well-known term-weighting systems. The

<table>
<thead>
<tr>
<th>Table 1. Term-weighting components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term Frequency Component</td>
</tr>
<tr>
<td>b 1.0</td>
</tr>
<tr>
<td>t tf</td>
</tr>
<tr>
<td>n 0.5 + 0.5 ( \frac{tf}{\max tf} )</td>
</tr>
<tr>
<td>Collection Frequency Component</td>
</tr>
<tr>
<td>x 1.0</td>
</tr>
<tr>
<td>( \log \frac{N-N}{n} )</td>
</tr>
<tr>
<td>Normalization Component</td>
</tr>
<tr>
<td>c 1/ ( \sqrt{\sum \text{wt}} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. Typical term-weighting formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weighting System</td>
</tr>
<tr>
<td>--------------------</td>
</tr>
<tr>
<td>Best fully weighted system (fc-nfc) ( \sqrt{\sum (c, \log N_{x})^{2}} )</td>
</tr>
<tr>
<td>Best weighted probabilistic weight axx-bpx ( 0.5 + 0.5 \frac{tf}{\max tf} )</td>
</tr>
<tr>
<td>Classical idf weight bfx-bfx ( \log \frac{N}{N_{x}} )</td>
</tr>
<tr>
<td>Binary term independence bxx-bxx ( \log \frac{N-N}{n} )</td>
</tr>
<tr>
<td>Standard tf weight: txc-tcx ( \sqrt{\sum (tf)^{2}} )</td>
</tr>
<tr>
<td>Coordination level bxx-bxx</td>
</tr>
</tbody>
</table>

A coordinate-level match, which simply reflects the number of matching terms present in documents and queries, respectively, is described by the sextuple bxx-bxx. Similarly, the probabilistic binary term independence system that uses binary document terms, but a probabilistic inverse collection frequency weight for the query terms, is represented as bxx-bpx. A typical complex term-weighting scheme, described as fc-nfc, uses a normalized tf x idf weight for document terms, and an enhanced, but unnormalized tf x idf factor for the queries. (Since the query vectors remain constant for all documents of a collection, a query normalization simply adds a constant factor to all query-document similarity measurements, which leaves the final document ranking unaffected.)

The six collections used experimentally are characterized by the statistics of Table 3. The smallest collection is a biomedical (MED) collection, consisting of 1033 documents and
Table 3. Collection statistics (including average vector length and standard deviation of vector lengths)

<table>
<thead>
<tr>
<th>Collection</th>
<th>Number of vectors (documents or queries)</th>
<th>Average vector length (number of terms)</th>
<th>Standard deviation of vector length</th>
<th>Average frequency of terms in vectors</th>
<th>Percentage of terms in vectors with frequency 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CACM</td>
<td>3,204</td>
<td>24.52</td>
<td>21.21</td>
<td>1.35</td>
<td>80.93</td>
</tr>
<tr>
<td>CRAN</td>
<td>1,460</td>
<td>46.55</td>
<td>19.28</td>
<td>1.37</td>
<td>80.27</td>
</tr>
<tr>
<td>INSPEC</td>
<td>7,788</td>
<td>9.17</td>
<td>3.19</td>
<td>1.04</td>
<td>95.69</td>
</tr>
<tr>
<td>MED</td>
<td>1,033</td>
<td>51.60</td>
<td>22.78</td>
<td>1.54</td>
<td>72.70</td>
</tr>
<tr>
<td>NPL</td>
<td>11,429</td>
<td>19.96</td>
<td>10.84</td>
<td>1.21</td>
<td>84.03</td>
</tr>
</tbody>
</table>

30 queries, whereas the largest collection (INSPEC) comprises 12684 documents and 84 queries, covering the computer engineering areas. In all cases, the query vectors are much shorter than the corresponding document vectors.

The National Physical Laboratory collection of 11429 documents and 100 queries was available in indexed form only (i.e., in the form of document and query vectors) and not in original natural language form. This may explain its somewhat peculiar makeup. Both the document and the query vector are much shorter in the NPL collection than in the other collections, and the variation in query length (2.36 for a mean number of 7.16 query terms) is very small. Furthermore, the term frequencies are especially low for the NPL collection: each query term appears precisely once in a query, and the average frequency of the terms in the documents is only 1.21. In these circumstances, the term frequency weighting and length normalization operations cannot perform their intended function. One may conjecture that the NPL index terms are carefully chosen and may in fact represent specially controlled terms rather than freely chosen natural language entries.

Typical evaluation output is shown in Tables 4 and 5. With a few minor exceptions, the results for the five collections of Table 4 are homogeneous, in the sense that the best results are produced by the same term-weighting systems for all collections, and the same holds also for the poorest results. The results of Table 4 do however differ substantially from those obtained for the NPL collection in Table 5. Considering first the results of Table 4, the following conclusions are evident:

1. Methods 1 and 2 produce comparable performances for all collections, the length normalization is important for the documents, and the enhanced query weighting is effective for the queries. These methods are recommended for conventional natural language texts and text abstracts.
2. Method 3 does not include the normalization operation for vector length, nor the enhanced query weights. This unnormalized $(t_f \times idf)$ weighting method is poor for collections such as CRAN and MED where very short query vectors are used with little deviation in the query length. In such cases, enhanced query weights $(n$ factor) prove important.
3. Method 4 represents the best of the probabilistic weighting systems. This method is less effective than the enhanced weighting schemes of methods 1 and 2. It fails especially for collections such as CACI and INSPEC where long query vectors are used and the term discrimination afforded by query term weighting is essential.

The results of Table 5 for the NPL collection differs markedly from those of Table 4. Here the probabilistic schemes using binary query weights and unnormalized document vectors are preferred. This is a direct result of the special nature of the queries and documents for that collection: the very short queries with little length deviation require fully weighted query terms $(b = 1)$, and the normally effective term frequency weights should be avoided because many important terms will then be downgraded in the short document vectors. An enhanced term frequency weight $(n$ factor), or a full weight $(b = 1)$ is therefore preferred. Retrieval results obtained for NPL were used earlier to claim superiority for the probabilistic term weighting system [55]. The results of Tables 4 and 5 do not support this contention for conventional natural language documents and queries.

### Table 4. Performance results for eight term-weighting methods averaged over 5 collections

<table>
<thead>
<tr>
<th>Term-weighting methods</th>
<th>Rank</th>
<th>CACM</th>
<th>CISI</th>
<th>CRAN</th>
<th>INSPEC</th>
<th>MED</th>
<th>Averages for 5 collections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best fully weighted</td>
<td>1</td>
<td>14</td>
<td>19</td>
<td>3</td>
<td>19</td>
<td>11.2</td>
<td></td>
</tr>
<tr>
<td>$(t_f/n_f)$</td>
<td>$P$</td>
<td>0.3630</td>
<td>0.2189</td>
<td>0.3841</td>
<td>0.2626</td>
<td>0.5628</td>
<td></td>
</tr>
<tr>
<td>Weighted with inverse frequency $f/n_f$ used for docs $(t_xe/n_x)$</td>
<td>2</td>
<td>14</td>
<td>3</td>
<td>4</td>
<td>32</td>
<td>16.4</td>
<td></td>
</tr>
<tr>
<td>$(t_f/n_f)$</td>
<td>$P$</td>
<td>0.3232</td>
<td>0.2189</td>
<td>0.3950</td>
<td>0.2626</td>
<td>0.5542</td>
<td></td>
</tr>
<tr>
<td>Classical $(t_f \times idf)$ No normalization $(t_f/x)$</td>
<td>3</td>
<td>22</td>
<td>219</td>
<td>45</td>
<td>122</td>
<td>84.4</td>
<td></td>
</tr>
<tr>
<td>$(t_f/x)$</td>
<td>$P$</td>
<td>0.3248</td>
<td>0.2166</td>
<td>0.3991</td>
<td>0.2365</td>
<td>0.5877</td>
<td></td>
</tr>
<tr>
<td>Best weighted probabilistic $(b_xe/b_x)$</td>
<td>4</td>
<td>208</td>
<td>7</td>
<td>97</td>
<td>60</td>
<td>86.2</td>
<td></td>
</tr>
<tr>
<td>$(t_xe/b_x)$</td>
<td>$P$</td>
<td>0.2909</td>
<td>0.1441</td>
<td>0.3999</td>
<td>0.2093</td>
<td>0.5449</td>
<td></td>
</tr>
<tr>
<td>Classical idf without normalization $(b_xe/b_x)$</td>
<td>5</td>
<td>247</td>
<td>183</td>
<td>160</td>
<td>178</td>
<td>82.8</td>
<td></td>
</tr>
<tr>
<td>$(t_xe/b_x)$</td>
<td>$P$</td>
<td>0.2353</td>
<td>0.1410</td>
<td>0.3184</td>
<td>0.1781</td>
<td>0.5062</td>
<td></td>
</tr>
<tr>
<td>Binary independence probabilistic $(b_xe/b_x)$</td>
<td>6</td>
<td>262</td>
<td>154</td>
<td>195</td>
<td>147</td>
<td>159</td>
<td></td>
</tr>
<tr>
<td>$(b_xe/b_x)$</td>
<td>$P$</td>
<td>0.2276</td>
<td>0.1323</td>
<td>0.3056</td>
<td>0.1563</td>
<td>0.5116</td>
<td></td>
</tr>
<tr>
<td>Standard weights cosine normalization $(t_xe/n_f)$</td>
<td>7</td>
<td>173</td>
<td>137</td>
<td>187</td>
<td>246</td>
<td>184</td>
<td></td>
</tr>
<tr>
<td>$(t_xe/n_f)$</td>
<td>$P$</td>
<td>0.2102</td>
<td>0.1539</td>
<td>0.3408</td>
<td>0.1620</td>
<td>0.4641</td>
<td></td>
</tr>
<tr>
<td>Coordination level binary vectors $(b_xe/b_x)$</td>
<td>8</td>
<td>284</td>
<td>280</td>
<td>238</td>
<td>281</td>
<td>260</td>
<td></td>
</tr>
<tr>
<td>$(b_xe/b_x)$</td>
<td>$P$</td>
<td>0.1848</td>
<td>0.1033</td>
<td>0.2414</td>
<td>0.0944</td>
<td>0.4132</td>
<td></td>
</tr>
</tbody>
</table>

### Table 5. Performance results for NPL collection (11429 docs, 100 queries)

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Best fully weighted $(t_f/n_f)$</th>
<th>Weighted restricted $(t_xe/n_f)$</th>
<th>Classical $(t_f \times idf)$ $(b_xe/b_x)$</th>
<th>Classical $(t_f/txe)$ $(b_xe/b_x)$</th>
<th>Binary independence $(t_xe/b_x)$</th>
<th>Standard weight $(t_xe/n_f)$</th>
<th>Coordination level $(b_xe/b_x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank</td>
<td>116</td>
<td>62</td>
<td>149</td>
<td>2</td>
<td>23</td>
<td>8</td>
<td>172</td>
</tr>
<tr>
<td>Average precision</td>
<td>0.1933</td>
<td>0.2170</td>
<td>0.1846</td>
<td>0.2752</td>
<td>0.2406</td>
<td>0.2596</td>
<td>0.1750</td>
</tr>
</tbody>
</table>
4. RECOMMENDATIONS

The following conclusions may be drawn from the experimental evidence reported in this study:

4.1 Query vectors

1. Term-frequency component
   • For short query vectors, each term is important; enhanced query term weights are thus preferred: first component $n$.
   • Long query vectors require a greater discrimination among query terms based on term occurrence frequencies: first component $f$.
   • The term-frequency factor can be disregarded when all query terms have occurrence frequencies equal to 1.

2. Collection-frequency component
   • Inverse collection frequency factor $f$ is very similar to the probabilistic term independence factor $p$: best methods use $f$.

3. Normalization component
   • Query normalization does not affect query-document ranking or overall performance; use $x$.

4.2 Document vectors

1. Term-frequency component
   • For technical vocabulary and meaningful terms (CRAN, MED collections), use enhanced frequency weights: first component $n$.
   • For more varied vocabulary, distinguish terms by conventional frequency weights: first component $f$.
   • For short document vectors possibly based on controlled vocabulary, use fully weighted terms: first component $b = 1$.

2. Collection-frequency component
   • Inverse document-frequency factor $f$ is similar to probabilistic term independence factor $p$: normally use $f$.
   • For dynamic collections with many changes in the document collection makeup, the $f$ factor requires updating; in that case disregard second component: use $x$.

3. Length-normalization component
   • When the deviation in vector lengths is large, as it normally is in text indexing systems, use length normalization factor $c$.
   • For short document vectors of homogeneous length, the normalization factor may be disregarded; in that case use $x$.

The following single-term weighting systems should be used as a standard for comparison with enhanced text analysis systems using thesauruses and other knowledge tools to produce complete multimodal content identifications:

- Best document weighting $tfc$, $nfc$ (or $tpc$, $npc$)
- Best query weighting $nfx$, $fzx$, $bxz$ (or $npxz$, $ipxz$, $bpzx$)

REFERENCES

1. Luhn, H.P. A statistical approach to the mechanized encoding and searching of literary information. IBM Journal of Research and Development 1:4; 399-409; 1957.
34. Lancaster, F.W. Evaluation of the Medlars demand search service, National Library of Medicine, Bethesda, MD; January 1968.
51. Wong, S.K.M.; Ziarko, W.; Raghavan, V.V.; Wong, P.C.N. Extended Boolean query processing in the generalized vector space model, Report, Department of Computer Science, University of Regina, Regina, Canada; 1986.