FUB at TREC-10 Web Track: A probabilistic framework for topic relevance term weighting

Gianni Amati [*] Claudio Carpineto
Giovanni Romano
Fondazione Ugo Bordoni, Roma, Italy

1 Introduction

The main investigation of our participation in the WEB track of TREC-10 concerns the effectiveness of a novel probabilistic framework [1] for generating term weighting models of topic relevance retrieval. This approach endeavours to determine the weight of a word within a document in a purely theoretic way as a combination of different probability distributions, with the goal of reducing as much as possible the number of parameters which must be learned and tuned from relevance assessments on training test collections.

The framework is based on discounting the probability of terms in the whole collection, modeled as deviation from randomness, with a notion of information gain related to the probabilty of terms in single documents. The framework is made up of three components: the "information content" component relative to the entire data collection, the "information gain normalization factor" component relative to a subset of the data collection (the elite set of the observed term), and the "term frequency normalization function" component relative to the document length and to other collection statistics. Each component is obtained by computing a suitable probability density function.

One advantage of the framework is that we may easily compare and test the behaviour of different basic models of Information Retrieval under the same experimental conditions and normalization factors and functions. At the same time, we may test and compare different term frequency normalization factors and functions.

In addition to testing the effectiveness of the term weighting framework, we were interested in evaluating the utility of query expansion on the WT10g collection. We used information theoretic query expansion and focused on careful paremeter selection.

In our experiments, we did not use link information, partly because of tight scheduling

[*] The author has been carrying out his work also at the Computing Science Department, University of Glasgow, Scotland

- the WT10g collection was made available to us as late as late May 2001 - and partly because it has been shown at TREC-9 that it is not beneficial to topic relevance retrieval.

In the rest of the paper, we first introduce the term weighting framework. Then we describe how collection indexing was performed, which probability functions were used in the experiments to instantiate the term weighting framework, and which parameters were chosen for query expansion. A discussion of the main results concludes the paper.

2 The term weighting framework

The framework presented here can be found in [1].

The fundamental weighting Formula is the product of two *information content* functions:

$$w = \quad Inf_1 \cdot Inf_2 \tag{1}$$

Both Inf_1 and Inf_2 are decreasing functions of two probabilities P_1 and P_2, respectively. The function Inf_1 is related to the whole document collection D, whilst Inf_2 to the elite set E_t (this notion roots back to Harter's work [5]) of the term t, namely the set of all documents in which the term t occurs.

P_1 is obtained as follows. We assume that words which bring little information are randomly distributed on the whole set of documents. By contrast, informative words diverge from the randomic behaviour and therefore they receive little probability according to a suitable model of randomness for Information Retrieval. This is the *the inverse document frequency "component"* of our model, in the sense that similar to the standard IR models based on the idf measure, the informative words have a small probability to occur within a document. We provide different basic models which defines such a notion of *randomness in the context of Information Retrieval.* A model of randomness is derived by a suitable interpretation of the probabilistic urn models of Types I and II [4] into the context of Information Retrieval. Basically, a model of Type I is a model where balls (tokens) are randomly extracted from an urn, whilst in Type II models balls are randomly extracted from an urn belonging to a collection of urns (documents). Among type I models there is the Poisson model, the Bose-Einstein statistics, the Geometric distribution, whilst a type II model is the inverse document frequency model. Therefore, the frequency of a word within a document with the lowest probability P_1 as predicted by such models of randomness or, equivalently, the words whose probability is less expected by the chosen model of urns, are "highly informative" words.

$$Inf_1 = -\log P_1 \quad \texttt{large for informative words in the collection}$$

If we observe the elite set E_t of the term, then we may derive a second conditional probability P_2 of term occurrence within a document in its elite set. The information content of a highly informative term, as obtained by means of Inf_1, will be tuned according to its elite set.

This weight tuning process corresponds to *the information gain "component"* of our model. *We will take as weighting formula only a fraction Inf_2 of Inf_1.* This fraction of the information content corresponds to the "gain" associated to the decision of accepting the term as an informative descriptor of the document. We assume, as in decision theory, that this information gain and thus inf_2 is inversely related to its odds P_2.

Differently from P_1, which is in general very small, P_2 should be in general close to certainty, especially when tf is large. If we observe many occurrences of the term t, then the observed term should have a very high probability P_2 of being a descriptor of the document. We assume that P_2 is the conditional probability of observing, within an *arbitrary document* of the elite set, $tf + 1$ occurrences of a given word in the hypothesis that one has already observed tf occurrences. The higher the term frequency tf, the higher the conditional P_2. Since the gain is inversely related to its odds:[1]

$$Inf_2 = 1 - P_2 \qquad \texttt{rate of the information content gained with } t$$

The weight of a term in a document is thus a function of two probabilities P_1 and P_2 which are related by the following relation:

$$w \quad = Inf_1 \cdot Inf_2 \quad = (-\log_2 \, P_1) \ \cdot (1 - P_2) \tag{2}$$

The term weight w of Formula 2 can be seen as a function of 4 random variables:

$$w = w(F, tf, n, N)$$

where
tf is the within document term frequency
N is the size of the collection
n is the size of the elite set E_t of the term,
F is the term frequency in its elite set

However, the size of tf depends on the document length: we have to derive the expected term frequency in a document when the document is compared to a fixed length (typically the average document length). We should determine what is the distribution that the tokens of a term follow in the documents of a collection at different document lengths. Once this distribution is obtained, the normalized term frequency tfn is used in the Formula 2 instead of the non-normalized tf.

One formula we have formally derived and successfully tested on previous TREC collections is:

$$tfn = \quad tf \cdot \log_2 \left(1 + \frac{c \cdot avg\_l}{l}\right) \quad \texttt{(with } c \geq 1\texttt{)} \tag{3}$$

[1] We also used an alternative monotone decreasing function, namely $Inf_2 = -\log_2 P_2$. Experimentally, this decreasing function seems to be a little less effective than $Inf_2 = 1 - P_2$. Also, the function $1 - P_2$ will be easily generalized below to the increment rate of two Bernoulli's trials, whilst a similar generalization with $Inf_2 = -\log_2 P_2$ is problematic.

where avg_l and l are the average length of the document collection and the length of the observed document respectively.

Our term weight w of Formula 2 will be thus a function of 6 random variables:

$$w = w(F, tfn, n, N) = w(F, tf, n, N, l, avg\_l)$$

where l is the document length
avg_l is the length mean

We postpone the discussion about the probability functions used to instantiate this framework and the choice of parameter c to Section 4.2. We first describe, in the next section, how collection indexing was performed.

3 Test collection indexing

Text segmentation. Our system first identified the individual terms occurring in the test collection, ignoring punctuation and case. The whole body of each document was indexed except for HTML tags, which were removed from documents. *Pure single keyword indexing was performed, and link information was not used.*

Document pruning. As we had very limited storage capabilities, we performed some document pruning. We removed very long or short documents as well as documents which were deemed to be nontextual or nonenglish textual. Specifically, we pruned the documents with more than 10,000 words (2,897) or less than 10 words (57,031); also, we removed the documents that contained more than 50% of unrecognized English word (86,146), according to a large morphological lexicon for English (Karp et al, 1992). In all, we removed 118.087 documents (this is not the exact sum of the three categories due to document overlap). The price we paid for this computational gain is that some relevant documents were lost. More exactly, we removed 162 out of 3363 relevant documents (4.81%). Thus, it should be emphasized that our actual performance retrieval was probably lower than the performance that we would have obtained by considering the whole set of documents.

Word pruning. Incorrect words affect collection statistics and query expansion. In order to reduce the inherent web word noise, we removed very rare, ill-formed or exceedingly long words. Specifically, the words contained in no more than 10 documents, which were apparently exclusively mispelled words, were dropped from the document descriptions. The words containing more than three consecutive equal characters or longer than 20 characters were also deleted. In this way, the number of distint words in the collection decreased dramatically, from 1,602,447 (after steps 1 and 2) to only 293,484.

Stop wording and word stemming. As we were primarily interested in early precision, we used a very limited stop list and did not peform word stemming at all.

The system has been implemented in ESL, a Lisp-like language that is automatically translated into ANSI C and then compiled by gcc compiler. The system indexes two gigabytes of documents per hour and allows sub-seconds searches on a 550 MHz Pentium III with 256 megabytes of RAM running Linux.

4 Term weighting models

The term-weighting framework described above was instantiated to a number of models using the Bose-Eistein statistics and the inverse document frequency (expected and non) combined with the weight normalization factor Inf_2 and frequency normalization function tfn. We first describe the basic models and then the 2 normalization factors L and B for Inf_2.

Bose-Einstein statistics

The operational model of the Bose-Einstein statistics is constructed by approximating the factorials by Stirling's formula. The model B_E (B_E stands for Bose-Einstein) is:

$$Inf_1(tf) = \quad \log_2 \frac{(N+F-tf-2)!F!(N-1)}{(F-tf)!(N+F-1)!} \tag{4}$$

Let $\lambda = \frac{F}{N}$ be the mean of the frequency of the term t in the collection D, then the Bose-Einstein probability that a term occurs tf times in a document can be approximated by $P_1(tf) = \left(\frac{1}{1+\lambda}\right) \cdot \left(\frac{\lambda}{1+\lambda}\right)^{tf}$. The right hand side is known as the *geometric distribution* with probability $p = \dfrac{1}{1+\lambda}$. Hence:

$$Inf_1(tf) = \quad -\log_2 \left(\frac{1}{1+\lambda}\right) - tf \cdot \log_2 \left(\frac{\lambda}{1+\lambda}\right) \tag{5}$$

The approximations of Equation 4 by Stirling's formula and by Equation 5 were indistinguishable in the experiments, therefore Equation 5 is preferred to Equation 4 for its simplicity.

The inverse document frequency model $I(n)$

We use a standard tf-idf probability distribution. The probability $P_1(tf)$ is obtained by first computing the probability of choosing a document containing the given term at random and then computing the probability of having tf occurrences of the same term in a document:

$$Inf_1(tf) = tf \cdot \log_2 \frac{N+1}{n+0.5} \tag{6}$$

The inverse expected document frequency model $I(n_{exp})$

A different model can be obtained by Bernoulli's law. Let n_{exp} the expected number of documents containing the term under the assumption that there are F tokens in the collection. Then

$$n_{exp} = N \cdot Prob(tf \neq 0) = N \cdot (1 - B(N, F, 0)) = N \cdot (1 - \left(\frac{N-1}{N}\right)^F)$$

The third basic model is the tf-Expected_idf model $I(n_{exp})$:

$$Inf(tf) = tf \cdot \log_2 \frac{N+1}{n_{exp} + 0.5} \tag{7}$$

4.1 Term frequency normalizations: the probability P_2

We assume that the probability that an observed term contributes to select a relevant document is high if the probability of counting one more token of the same term in a relevant document is similarly high. This probability approaches 1 for high values of tf.

Laplace's normalization L

The first model of $P_2(tf)$ is obtained by the conditional probability $p(tf+1|tf, d)$ of Laplace's Law of Succession: $P_2(tf) = \dfrac{tf}{tf+1}$

The normalization L (for Laplace) is:

$$Inf_2 = \frac{1}{tf+1} \tag{8}$$

Bernoulli's normalization B

To obtain an alternative estimate of P_2 with Bernoulli's trials we use the following urn model. Let $B(n, F, tf)$ be

$$B(n, F, tf) \quad = \left(\begin{array}{c} F \\ tf \end{array} \right) p^{tf} q^{F-tf}$$

where $p = \frac{1}{n}$ and $q = \frac{n-1}{n}$.

We add a new token of the term to the collection, thus having $F+1$ tokens instead of F. We then compute the probability $B(n, F+1, tf+1)$ that this new token falls into the observed document, thus having a within document term frequency $tf+1$ instead tf. The process $B(n, F+1, tf+1)$ computes the probability of obtaining one more token of the term t in the document d out of all n documents in which t occurs when a new token is added to the elite set. The ratio

$$\frac{B(n, F+1, tf+1)}{B(n, F, tf)} = \frac{F+1}{n \cdot (tf+1)}$$

of the new probability $B(n, F+1, tf+1)$ to the previous one $B(n, F, tf)$ tells us whether the probability of encountering a new occurrence by chance is increased or diminished.

Instead of using P_2 we normalize with the probability increment rate

$$IncrementRate = 1 - \frac{B(n, F+1, tf+1)}{B(n, F, tf)}$$

that is the normalization B is:

$$Inf_2(tf) = \frac{F+1}{n \cdot (tf+1)} \tag{9}$$

4.2 The parameter c for the baseline models

Independently from the model used, namely independently from the probability distributions P_1 and P_2 chosen, in TREC-9 and TREC-10 the best matching value for c was 7. The parameter c seems to be proportional to the size of the collection and inversely proportional to the size of the indexing vocabulary. A similar observation held also for the TREC-1 to TREC-8 collections.

We conjecture that the parameter c is connected to the Zipfian law which relates the size of vocabulary to the size of the collection. This relationships which is not linear affects the size of term frequency in the collection and thus the term frequency in the document.

5 Query expansion

For TREC-9, the results about the use of query expansion were not as good as with previous TRECs. Several groups reported that expansion did not improve or even hurt retrieval performance [6]. As groups participating in TREC-9 web track had little opportunity for parameter tuning and the WT10g collection is very different from the previous collections, these result may have been influenced by poor choice of query expansion parameters.

We encountered a similar problem with our own information theoretic-based expansion method [3, 2]. The weight of a term of the expanded query q^* of the original query q is obtained as follows:

$$weight(t \in q^*) = (\alpha \cdot tfq_n + \beta \cdot tfn_{KL}) \cdot Inf_1 \cdot Inf_2$$

where

- tfq_n is the normalized term frequency within the original query q (i.e. $\frac{tfq}{max_{t \in q} tfq}$)
- tfn_{KL} is a term frequency in the expanded query induced by using a normalized Kullback-Leibler measure

  $$tfn_{KL} = \frac{tf_{KL}}{\max\limits_{t \in q^*} tf_{KL}} \tag{10}$$

  $$tf_{KL} = P_R(t) \cdot log \frac{P_R(t)}{P_C(t)}$$

  where $P_X(t)$, with $X = R, C$, is the probability of occurrence of term t in the set of documents X (estimated by the relative frequency of the term in X), R indicates the pseudo-relevant set, C indicates the whole collection.
- $\alpha = 1$, $\beta = 0.2$
- $|R| = 3$ with the number of terms of the expanded query equal to 10.
- Inf_1 and inf_2 as defined in Relation 1

This method was used with good results on TREC-8; however, when we ran it with the TREC-8 parameters against the TREC-9 collection the retrieval performance was badly affected, whether using the new weighting functions discussed above or the Okapi formula. Thus, we focused on better selection of the values used for query expansion parameters for the WT10g document set, by performing parameter tuning on the TREC-9 test collection. We considered three parameters, namely the number of pseudo relevant documents, the number of expansion terms and the ratio between α and β in Rocchio's formula.

One of the most striking characteristics of the WT10g collection is that the quality of baseline retrieval is lower than that obtained for past TREC collections. In an attempt to reduce the chance to select terms from mostly nonrelevant documents we chose fewer pseudo-relevant documents than typically used for query expansion. We set the number of pseudo-relevant documents at 3. In order to compensate for the lower quality of the terms used for expansion, we also adjusted the values of α and β. Since the original query should become more important as the quality of the expansion terms and their weights diminishes, we set the ratio between α and β to 5 (i.e., $\alpha = 1$, $\beta = 0.2$) and reduced the number of terms used for query expansion to 10. In this way, it should be easier for the expanded query to keep the focus on the original topic, even in the presence of bad term suggestions.

Finally, it should be noted that the removal of bad words performed at indexing time (see discussion above) may have considerably reduced the number of typographical errors in documents, which was pointed out as one of the causes for poor query expansion.

6 Runs at TREC 10

We submitted 4 runs, 2 of them with our query expansion technique.

Runs fub01ne and fub01ne2: $I(n_{exp})L$

The baseline model fub01ne for Inf_1 is $I(n_{exp})$ and the normalization formula for Inf_2 is Laplace's law L namely the term weight is:

$$w = \frac{1}{tfn+1} \cdot \left(tfn \cdot \log_2 \frac{N+1}{n_{exp}+0.5} \right) \tag{11}$$

tfn is defined as in Equation 3 with $c = 7$. Run fub01ne was performed without query expansion, whilst run fub01ne2 with.

Run fub01be2: $B_E L$**.** This was the best performing run at TREC-10. The baseline model fub01be for Inf_1 is B_E and the normalization formula for Inf_2 is Laplace's law L namely the term weight is:

$$w = \frac{1}{tfn+1} \cdot \left(-\log_2 \left(\frac{1}{1+\lambda} \right) - tfn \cdot \log_2 \left(\frac{\lambda}{1+\lambda} \right) \right) \tag{12}$$

tfn is defined as in Equation 3 with $c = 7$. The automatic query expansion was performed.

Run fub01idf: $I(n)B$

The baseline model fub01idf for Inf_1 is $I(n)$ and the normalization formula for Inf_2 is Bernoulli's rate B namely the term weight is:

$$w = \frac{F+1}{n(tfn+1)} \cdot tfn \log_2 \frac{N+1}{n+0.5} \tag{13}$$

tfn is defined as in Equation 3 with $c = 7$. The automatic query expansion was not performed.

7 Results and conclusions

In Table 1 we show the retrieval performance of all possible models that can be generated by the term weighting framework using the probability functions introduced above, without and with query expansion.

The main conclusions that can be drawn from the experimental results are the following.

- On the whole, the term weighting framework was effective, with very good absolute and comparative retrieval performance (run **fub01be2** achieved the best performance of all official submissions in the title-only, automatic topic relevance task),

Method	Official run	AvPrec	Prec-at-10	Prec-at-20	Prec-at-30
Model performance without query expansion					
$B_E L$		0.1788	0.3180	0.2730	0.2413
$I(n)L$		0.1725	0.3180	0.2740	0.2353
$I(n_{exp})L$	fub01ne	0.1790	0.3240	0.2720	0.2440
$B_E B$		0.1881	0.3280	0.2980	0.2487
$I(n)B$	fub01idf	0.1900	0.3360	0.2880	0.2580
$I(n_{exp})B$		0.1902	0.3340	0.2860	0.2580
Model performance with query expansion					
$B_E L$	fub01be2	0.2225	0.3440	0.2860	0.2513
$I(n)L$		0.1973	0.3200	0.2730	0.2380
$I(n_{exp})L$	fub01ne2	0.1962	0.3280	0.2760	0.2507
$B_E B$		0.2152	0.3400	0.2870	0.2527
$I(n)B$		0.2052	0.3380	0.2970	0.2680
$I(n_{exp})B$		0.2041	0.3360	0.2990	0.2660

Table 1: Comparison of performance of models and normalization factors.

although noteworthy differences in performance were observed depending on which combination of probabilistic distributions and normalization techniques was used.

- Query expansion with the chosen parameters improved performance for almost all term weighting models and evaluation measures, with more tangible benefits for average precision.

More work is necessary to investigate the relative strengths and weaknesses of each model as well as to study the relationships to other term weighting approaches. Moreover, further experiments should be performed to control the effect on performance of a wider range of factors, including word stemming, document pruning, and word pruning.

References

[1] Gianni Amati and Cornelis Joost van Rijsbergen. Probabilistic models of information retrieval based on measuring divergence from randomness. *Manuscript*, 2001.

[2] C. Carpineto, R. De Mori, G. Romano, and B. Bigi. An information theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, 19(1):1–27, 2001.

[3] C. Carpineto and G. Romano. Trec-8 automatic ad-hoc experiments at fub. In *In Proceedings of the 8th Text REtrieval Conference (TREC-8), NIST Special Publication 500-246*, pages 377–380, Gaithersburg, MD, 2000.

[4] Irving John Good. *The Estimation of Probabilities: an Essay on Modern Bayesian Methods*, volume 30. The M.I.T. Press, Cambrige, Massachusetts, 1968.

[5] Stephen Paul Harter. A probabilistic approach to automatic keyword indexing. part I: On the distribution of specialty words words in a technical literature. *Journal of the ASIS*, 26:197–216, 1975.

[6] D Hawking. Overview of the trec-9 web track. In *In Proceedings of the 9th Text Retrieval Conference (TREC-9)*, Gaithersburg, MD, 2001.