A New Probabilistic Ranking Model

Richard Connor
Department of Computer and Information Sciences
University of Strathclyde
Glasgow G1 1XH Scotland UK
richard.connor@strath.ac.uk

Robert Moss
Department of Computer and Information Sciences
University of Strathclyde
Glasgow G1 1XH Scotland UK
robert.moss@strath.ac.uk

Morgan Harvey
Faculty of Informatics
University of Lugano (USI)
CH-6900, Lugano, CH
morgan.harvey@usi.ch

ABSTRACT

Over the years a number of models have been introduced as solutions to the central IR problem of ranking documents given textual queries. Here we define another new model. It is a probabilistic model and has no term inter-dependencies, thus allowing calculation from inverted indices. It is based upon a simple core hypothesis, directly calculating a ranking score in terms of probability theory. Early results show that its performance is credible, even in the absence of parameters or heuristics. Its semantic basis gives absolute results, allowing different rankings to be compared with each other. The investigation of this model is at a very early stage; here, we simply propose the model for further investigation.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

Keywords

information retrieval, ranking, probabilistic retrieval, Jensen-Shannon Divergence

1. INTRODUCTION

An idealised Information Retrieval (IR) system should, given all the information available, rank documents in descending order of their expected relevance to an information need, usually expressed as a short keyword-based query [8].

In this work, we introduce a new ranking model, the Uniform Probability Model (UPM). In common with many other models, it assumes term independence, representing documents as so-called "bags of words", and is based on the notion of probabilistic unigram document generators [5].

Good empirical performance does not usually come directly from a well-motivated and parsimonious model, but rather from a number of heuristic modifications to such a model. The most successful models – e.g. BM25, the Language Model (LM) and TF/IDF – all use a combination of common heuristics, such as Inverse Document Frequency (IDF) weighting, document length normalisation, and term smoothing, and have parameters that must be individually tuned for each collection in order to perform well.

As defined here, in the absence of any parameters or heuristics, initial results show the performance of UPM to be comparable with much more sophisticated ranking functions. More complex models, specifically adapted to the ad hoc ranking problem or other IR tasks, can presumably be derived from the foundations of UPM; we propose UPM as a worthy foundation for further investigation.

2. RELATED WORK

The problem of ranking documents from a large collection, given a short textual query, has been long studied and is at the core of the IR field. Unigram IR systems regard each document (and the queries) as a set of distinct tokens, representing each unique term in the collection, where each token may have a count associated with it or merely a note of its presence or absence; the pragmatic value of such systems is that they can be evaluated using an inverted index, essential for fast results over huge collections.

The essence of such systems is a scoring or ranking function, which operates over tuples of values representing some aspect of the keys terms' presence in each document. An overview of methods using this model is given in [9].

Within this context, the most notable probabilistic models are the Binary Independence Model (BIM) and the Query Likelihood Model (LM). BIM, whilst theoretically sound, performs poorly on standard IR test collections unless large numbers of explicit relevance judgements are available to train upon. LM also has a principled basis, and estimates the probability of a query being relevant to a document, rather than vice versa. LM works well with adjunct smoothing functions, notably Dirichlet.

UPM approaches the problem of document ranking in a similar manner to that of the Divergence From Randomness model [1], an extension of Harter's 2-Possion model in which it is also assumed that there are a set of "elite" terms. Term weights are proportional to the divergence of the frequency of a term in a given document and its frequency over the entire collection, this divergence is determined by choosing a suitable underlying model from a number of available options. While it is possible to define a parameter-free version of this model, in order to obtain competitive levels of performance it is necessary to apply a number of parameterised normalisation steps.

ICTIR '13 September 29 - October 02 2013, Copenhagen, Denmark
Copyright 2013 ACM 978-1-4503-2107-5/13/09 ...$15.00
http://dx.doi.org/10.1145/2499178.2499185.

3. THE MODEL

The model is defined in the context of unigram generative models, based upon a notion of probabilistic generation of documents. We assume an invariant mapping from terms in a corpus to dimensions in the vector space, and all vectors are transformed into probability vectors.

$\mathcal{P}(t|v)$ denotes the probability assigned to term t in vector v, and $\sum_t \mathcal{P}(t|v) = 1$ for all vectors we consider. Vectors are used to model a number of different entities, including: the corpus, a document within the corpus, and term-generation oracles.

We define a term generator $\mathcal{G}^v$, driven by probability vector v, such that $\mathcal{G}^v$ randomly returns a term t with probability $\mathcal{P}(t|v)$. A document of length n can be notionally generated by making n successive calls to $\mathcal{G}^v$ and normalising the result into a new probability vector.

3.1 Probability Estimation

Consider a distance function $Dist$, bounded in $[0, 1]$, over probability vectors. We will consider the value $1 - Dist(d, v)$ to represent an estimate of the probability that generator $\mathcal{G}^v$ was used to produce document d. This will allow us to compare a document against different generators, in order to find the one most likely to have generated it.

Consider a notional set of *null* documents, within the context of a corpus $\mathbb{C}$, which are not relevant to *any* particular subject. We hypothesise that the terms of such documents are most likely to be drawn from the terms of the language with probabilities in proportion to the distribution over $\mathbb{C}$; that is, as the length of document d increases, then $Dist(d, \mathbb{C}) \to 0$.

We will use $\mathcal{G}^{\mathbb{C}}$ as a baseline for comparing actual documents drawn from the corpus. Notice that, if d_n is a randomly selected null document and d_1 is generated from any other $\mathcal{G}^v$, then $Dist(d_1, \mathbb{C})$ is likely to be greater than $Dist(d_n, \mathbb{C})$.

3.2 Documents and Key Terms

Now consider documents which are relevant to a given set of key terms k_1 to k_n. We hypothesise only that the frequency with which these terms appear within each such document is likely to be greater than the frequency with which the same term appears within the corpus.

The number of occurrences of these terms is most likely to increase as a single additive value over and above its frequency within the corpus: for example, a document about *red aardvark*s may well have extra instances of that phrase, but a significantly long document is still likely to have the terms in other contexts as well. As a consequence, although a less obvious effect, all other terms in the document will have a decreased frequency.

These observations allow us to construct a model generator $\mathcal{G}^{v_{\mathbb{K}}}$for documents about a subject characterised by a set of key terms $\mathbb{K}$. The vector $v_{\mathbb{K}}$ created to drive the generator is given by:

$$\mathcal{P}(t|v_{\mathbb{K}}) = \begin{cases} \mathcal{P}(t|\mathbb{C}) + \varepsilon & \text{if } t \in \{\mathbb{K}\} \\ \epsilon \cdot \mathcal{P}(t|\mathbb{C}) & \text{otherwise} \end{cases}$$

where ε is a constant used to represent an increase in each probability deriving from the extra occurrences of each key term, and ϵ simply rebalances the probabilities of the non-key terms so that all probabilities sum to one.

Note that ε is *not* an adjustable parameter. As we are interested only in comparisons of $Dist(d, \mathbb{C})$ and $Dist(d, v_{\mathbb{K}})$, the correct notional value for ε is infinitesimal, to minimise $Dist(\mathbb{C}, v_{\mathbb{K}})$. An analogy in two dimensions would be to model whether $P = (x, y)$ lies above the X-axis by comparing $Dist(P, (0, 0))$ with $Dist(P, (0, \varepsilon))$ - the smaller the value of ε, the more correct the model.

3.3 Relative Probability Ranking

If the distance we are considering gives a good estimate of generation probability, it is clear that the inequality

$$Dist(d, v_{\mathbb{K}}) < Dist(d, \mathbb{C})$$

is more likely to be true than false for documents generated from $\mathcal{G}^{v_{\mathbb{K}}}$; whereas, if document d is about a different subject, then the same calculation is more likely to be false.

As a corollary, we can consider the function

$$R_{\mathbb{K}}(d) = Dist(d, \mathbb{C}) - Dist(d, v_{\mathbb{K}})$$

as a ranking function for documents d over the key terms $\mathbb{K}$.

3.4 Non-key terms and noise

We make one simplification to the document model, to remove noise from the document collection: for all terms in each document which are *not* stated as key terms in a search, we assume that the term frequency is in ratio with the corpus term frequency. This assumption is made in other probabilistic models, such as the BIM model [10]. Thus, we consider document vectors which maintain their term frequencies for terms in the query, but otherwise effectively lose all other information. For document d and set of key terms $\mathbb{K}$, we denote the version with noise removed as $d^{\mathbb{K}}$.

The ranking function $R_{\mathbb{K}}$ for the set of key terms $\mathbb{K}$ is now:

$$R_{\mathbb{K}}(d) = Dist(d^{\mathbb{K}}, \mathbb{C}) - Dist(d^{\mathbb{K}}, v_{\mathbb{K}}) \qquad (1)$$

All that remains is to find a suitable distance function, ideally one that can be efficiently calculated from only the document term frequencies of the given search terms. For this we use Jensen-Shannon divergence.

4. EVALUATION OF JENSEN-SHANNON

Jensen-Shannon (JS) is the name given in [4] to a divergence function identified in [7]. In essence it is a smoothed, symmetrised version of the Küllback-Leibler divergence [3]. More recent analysis e.g. [2] has shown important semantic properties, and it is being increasingly investigated. We believe this is its first application to IR ranking.

JS is defined in terms of Küllback-Leibler divergence:

$$JS(v, w) = \tfrac{1}{2}KL(v, m) + \tfrac{1}{2}KL(w, m)$$

where m is the vector mean of v and w. If logs are taken to base two, then the outcome is bounded in [0,1].

Some simple algebra gives two other forms of interest for the same function:

$$JS(v, w) = H(m) - \tfrac{1}{2}H(v) - \tfrac{1}{2}H(w) \qquad (2)$$

where H is Shannon's entropy function; and, from this:

$$JS(v, w) = \tfrac{1}{2}\sum_i \mathcal{F}(v_i, w_i)$$

for a kernel function $\mathcal{F}$ defined by

$$\mathcal{F}(x, y) = h(x + y) - h(x) - h(y)$$

where $h(x) = -x \log_2(x)$.

An important property of $\mathcal{F}$ in our context is that, when summing over terms which are in ratio, then a single evaluation over the sum of the terms gives the same outcome: that is, if $\frac{a}{b} = \frac{c}{d}$, then $\mathcal{F}(a, b) + \mathcal{F}(c, d) = \mathcal{F}(a + c, b + d)$.

The importance of this property is that, as they are all in the same ratio, all of the residual (non-key) terms in both $\mathbb{C}$ and $d_i^{\mathbb{K}}$ can be treated as a *single* term in the calculation, by summing their values in both vectors. The value of this sum is given by $1 - \sum_{t \in \mathbb{K}} \mathcal{P}(t|v)$, meaning that only the values of $\mathcal{P}(t|v)$ where $t \in \mathbb{K}$ need to be accessed during the distance calculation. This is the property that allows calculation of the apparently complex distances using only the terms available from an inverted index.

We now return to the task of calculating the ranking function (Equation 1). The distance between $d^{\mathbb{K}}$ and $\mathbb{C}$ is:

$$\sum_{t \in \mathbb{K}} \mathcal{F}(\mathcal{P}(t|d^{\mathbb{K}}), \mathcal{P}(t|\mathbb{C})) + \mathcal{F}(1 - \sum_{t \in \mathbb{K}} \mathcal{P}(t|d^{\mathbb{K}}), 1 - \sum_{t \in \mathbb{K}} \mathcal{P}(t|\mathbb{C}))$$

and the distance between $d_i^{\mathbb{K}}$ and $v_{\mathbb{K}}$ is:

$$\sum_{t \in \mathbb{K}} \mathcal{F}(\mathcal{P}(t|d^{\mathbb{K}}), \mathcal{P}(t|v_{\mathbb{K}})) + \mathcal{F}(1 - \sum_{t \in \mathbb{K}} \mathcal{P}(t|d^{\mathbb{K}}), 1 - \sum_{t \in \mathbb{K}} \mathcal{P}(t|v_{\mathbb{K}}))$$

and therefore our ranking function is the difference between these terms.

Finally, it can be noted that the *difference* between the final terms of these distances, i.e. the non-key terms, is constant for any given set of keywords, and does not therefore affect the order of results for a given query. This allows the ranking to be simplified to:

$$R(d, \mathbb{K}) = \sum_{t \in \mathbb{K}} \mathcal{F}(\mathcal{P}(t|d^{\mathbb{K}}), \mathcal{P}(t|\mathbb{C})) - \mathcal{F}(\mathcal{P}(t|d^{\mathbb{K}}), \mathcal{P}(t|v_{\mathbb{K}}))$$

For each term in the query, we therefore require only the term frequency in the corpus, from which the term frequency in the notional subject generator is calculated, and the term frequency in the document.

If an absolute value is required, to allow comparison of results among different queries, then the residual constants can be calculated at little extra cost.

5. EVALUATION

5.1 Method

Performance evaluation was carried out using the Terrier IR platform [6], and its implementations of: TF/IDF with Robertson's TF (TFIDF), BM25, and LM with Bayesian smoothing and Dirichlet prior (DirLM). We used default parameters for each of the models found in the literature, which are quite well optimised for TREC experimentation; BM25 $k_1 = 1.2$, $k_2 = 8$, $b = 0.75$; Robertson TF/IDF $k_1 = 1.2$, $b = 0.75$; Dirichlet LM $\mu = 2500$.

We used the Text Research Collection Volumes 4 & 5 from the Text REtrieval Conference (TREC), with topics 301-400 from TREC-6 and TREC-7. Each topic provides three fields. A title, a description, and a narrative; we ran short queries (where only the title was used) and long queries (where all three fields were used).

5.2 Results

	Trec 6-7			Trec 6-7 Long		
Metric	mAP	mRR	nDCG	mAP	mRR	nDCG
UPM	0.280	0.575	0.603	0.219	0.626	0.560
BM25	0.292	0.603	0.621	0.237	0.677	0.587
DirLM	0.275	0.543	0.597	0.196	0.536	0.543
TFIDF	0.294	0.604	0.623	0.232	0.672	0.582

Table 1: mAP, mRR and nDCG values for TREC

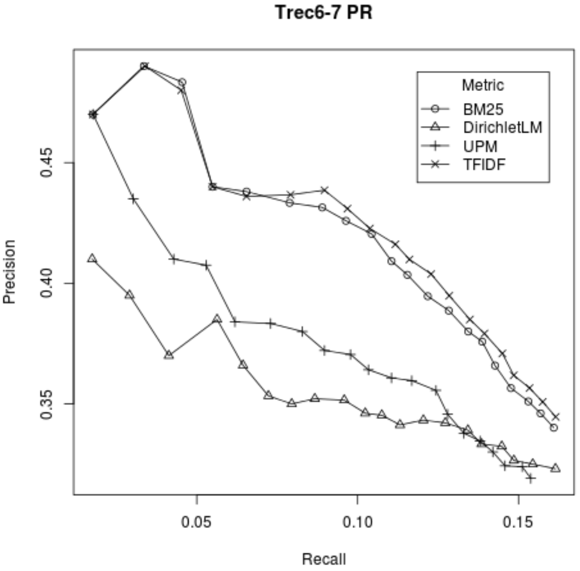

Figure 1: Precision/recall, short queries, Trec6-7

Table 1 shows the performance of all of the models for both short and long queries. We give three standard metrics: mean average precision (mAP), mean reciprocal rank (mRR), and mean normalised discounted cumulative gain (nDCG) over the first 1,000 documents returned for each query. Figures 1 and 2 give the corresponding precision/recall charts for the first 20 results returned over short and long queries respectively; after this point, there is increasingly little discernible difference among the models.

An easy observation is that UPM outperforms DirichetLM, and is outperformed by the other functions. However over a single collection and a relatively small set of queries this is not a statistically valid observation; furthermore, in practice, there would be little to choose among these models given the absolute values reported. However the purpose of these measurements, in this context, is only to show that the performance of the initial form of UPM, before the development of any appropriate heuristics or parameters, is entirely credible even in competition with functions which have been highly refined from their original, conceptual definitions.

We also performed tests using some others of the more accessible collections, including the WT10G and Blogs06. The relative performance was similar, but UPM fared relatively less well with the shorter documents in these collections.

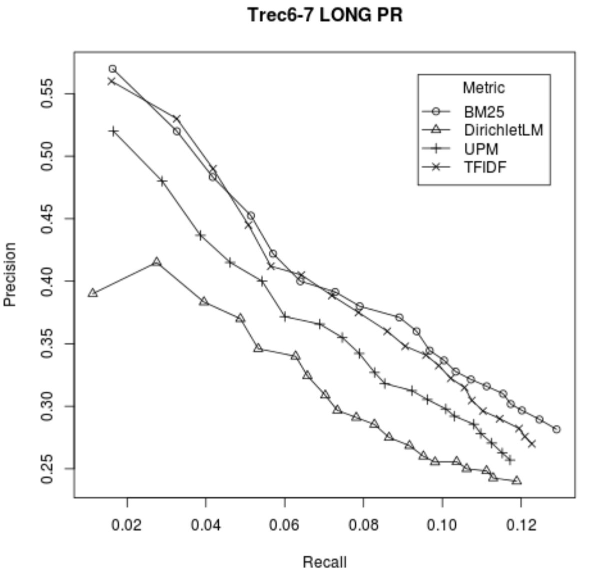

Figure 2: Precision/recall, long queries, Trec6-7

6. CONCLUSIONS AND FUTURE WORK

In this work we have described a new retrieval model, UPM, based on a probabilistic semantics. We have shown that the pure model, in the absence of further refinement, has performance comparable to three highly competitive baselines, all of which have significant heuristic refinements over their conceptual definitions. However we regard this work as being at a very early stage of development, with much investigation remaining to be performed.

The performance of the unrefined UPM model is stronger for long queries and long documents. This is not surprising: the semantics upon which the model is defined rely upon documents being represented by probability vectors. When documents are very long, then dividing a term count by the document length does represent a reasonable approximation of probability; however for shorter documents, the confidence with which this probability may be assigned is much lower. This is of course the basis of applying smoothing techniques [11] over the core semantics of the language model; we have not yet tried this approach with UPM and would expect to see a significant improvement especially with shorter documents.

It is interesting to compare the underlying hypothesis of the UPM model with the LM model; the former is based on the probability of query terms occurring within different documents, while the latter is based directly on the probability of a query being relevant to a document. This would imply directly that UPM may be a better model over longer documents and queries, and that the LM may be better over shorter documents and queries. Our results so far support this, and it would be interesting to test UPM in the context of tasks such as patent retrieval, where the queries are often entire documents.

Although shown here as a ranking function, absolute values can also be calculated, allowing the relevance of different query results to be compared with each other. If relevance values are absolute then they can be used to weight terms in candidate documents for purposes such as Pseudo Relevance Feedback.

Finally, looking at the mathematics beyond the intuition, our method effectively constructs a Voronoi hyperplane between $\mathbb{C}$ and $v_{\mathbb{K}}$ and then awards the highest scores to those points which are closest aligned with a perpendicular through $v_{\mathbb{K}}$, not taking distance from the plane into account. This is significantly at odds with most other models, where greater distances would give greater ranking scores, and closer inspection of the particular documents retrieved and not retrieved in comparison with other models may allow useful refinement.

7. ACKNOWLEDGEMENTS

We would like to thank the anonymous referees for their enthusiasm and for some very helpful suggestions.

We would like to acknowledge financial support from the UK Research Councils (EPSRC EP/G012407) and EU FP7 (D4SCIENCE, 212488 and D4S-II, 239019.)

8. REFERENCES

[1] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, Oct. 2002.

[2] B. Fuglede and F. Topsøe. Jensen-shannon divergence and hilbert space embedding. In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, pages 31–, 2004.

[3] S. Küllback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 1951.

[4] J. Lin. Divergence measures based on the shannon entropy. *Information Theory, IEEE Transactions on*, 37(1):145–151, 1991.

[5] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing.* MIT Press, Cambridge, MA, USA, 1999.

[6] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, 2006.

[7] C. R. Rao. Diversity: Its measurement, decomposition, apportionment and analysis. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 44(1):pp. 1–22, 1982.

[8] S. E. Robertson. Readings in information retrieval. chapter The probability ranking principle in IR, pages 281–286. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[9] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA, 1986.

[10] C. Zhai. Statistical language models for information retrieval a critical review. *Found. Trends Inf. Retr.*, 2(3):137–213, Mar. 2008.

[11] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, Apr. 2004.