A Proximity Probabilistic Model for Information Retrieval

Ruihua Song, Liqian Yu, Ji-Rong Wen, and Hsiao-Wuen Hon

Microsoft Research Asia,
Beijing, 100190, China
{rsong,liqy,jrwen,hon}@microsoft.com

Abstract. We propose a proximity probabilistic model (PPM) that advances a bag-of-words probabilistic retrieval model. In our proposed model, a document is transformed to a pseudo document, in which a term count is propagated to other nearby terms. Then we consider three heuristics, i.e., the distance of two query term occurrences, their order, and term weights, and try four kernel functions in measuring a position-dependent term count, which can be viewed as a pseudo term frequency. Finally, we integrate term proximity into the probabilistic model BM25 by using the pseudo term frequency to replace term frequency. Experimental results on TREC data sets indicate that the proximity probabilistic model with the reverse kernel function consistently improves the BM25 model by 5% - 11%, in terms of Mean Average Precision.

1 Introduction

Bag-of-words models are criticized in Information Retrieval (IR) because they ignore the dependency of terms. It is intuitive that the document where the matched query terms occur closely to each other is more likely to be relevant than the document where the matched query terms are isolated. However, it is not trivial to integrate term dependency or term proximity into IR models and achieve consistent improvements over the bag-of-words models.

Some previous works obtained promising results [6][9][1][10]. The recent work [4][11] internally integrated proximity factors into language models. However, few studies have been conducted based on the state-of-the-art probabilistic models.

In this paper, we propose a proximity probabilistic model (PPM) that integrates term proximity into the state-of-the-art probabilistic model BM25. We allow a term to propagate its term count to its surrounding terms and transforms a document into a pseudo document. In estimating a position-dependent term count, we model three heuristics, i.e., the distance between two terms, whether the order is the same as they occur in the query, and term weights, and propose four proximity kernel functions, i.e., the Gaussian, linear, parabola, and reverse functions. Finally, we integrate position-dependent term counts into the BM25 model by replacing term frequency.

We evaluate the proposed proximity probabilistic models and the baseline BM25 model on public data sets from Text Retrieval Conference (TREC). Experimental results show that the proximity probabilistic model is effective to

improve the bag-of-words model in terms of Mean Average Precision. In particular, the reverse kernel function, which is new, performs better than the other kernel functions that have been experimented in the previous work [4].

The remaining part of this paper is organized as follows. We briefly introduce related works in Section 2. Then we describe our proposed proximity model in Section 3 and report experimental results in Section 4. Finally, we conclude our work in Section 5.

2 Related Work

2.1 Probabilistic IR Models

The classic probabilistic model introduced in 1976 by Robertson and Sparck Jones [7], a.k.a. the binary independence retrieval (BIR) model, is based on Probabilistic Principle. Let $P(R|d)$ be the probability that the document d is relevant to the query q and $P(\overline{R}|d)$ be the probability that d is non-relevant to q. Documents are ranked by the following ratio:

$$\frac{P(R|d)}{P(\overline{R}|d)}$$

BM25 is the state-of-the-art implementation of the BIR model [8]. Given a query q, the relevance score of a document d with respect to the query is calculated as follows:

$$RS_b = \sum_{q_i \in q} w_i * \frac{tf_i}{K + tf_i}$$

$$K = k_1((1-b) + b\frac{l}{avdl})$$

Here, q_i denotes a key term in the query q; tf_i is term frequency of q_i; l is document length, and $avdl$ is average document length; k_1 and b are parameters. w_i is q_i's term weight, usually estimated by the formula proposed by Robertson and Sparck-Jones [7]:

$$w_i = \log \frac{N - df_i + 0.5}{df_i + 0.5}.$$

where N is the number of documents within a collection, and df_i is document frequency of q_i. The BM25 retrieval model performs robustly and effectively as shown in previous TREC experiments. We choose it as the reference model in this study.

2.2 Proximity Related IR Models

When we integrate term proximity in IR models, two key issues should be considered: One is how to measure term proximity, and another is how to integrate the proximity measures into a retrieval model.

As Tao and Zhai [10] summarized, there are two kinds of approach on measuring proximity: (1) Pair-based approaches, such as [6][1][11], measure the proximity based on the distance between two terms. (2) Span-based approaches, such as [2][3][9][5], measure the proximity based on text segments, i.e. spans, that cover more than one query term. Lv and Zhai's approach is somehow in between [4]. They define a language model for each position of a document, which is estimated based on the propagated word counts from the words at all other positions. Thus, the score for a position conveys the compactness of surrounding query terms that compose a soft span.

On integrating the proximity measures into retrieval, three strategies have been proposed: (1) Best score strategy, such as [2][3], uses the maximum score for spans. (2) Score combination strategy, such as [6][1][10], combines the score based on proximity measures with the score based on term independent models. (3) Internal integration strategy, such as [9][4][11], transforms a document into a pseudo bag-of-words document, which combines the original document model with the document's proximity model with respect to a query at term level. As a result, term frequency is transformed into a pseudo term frequency.

Our work considers the influence from a term's surrounding terms in counting term frequency and integrates proximity heuristics into probabilistic models. In some sense, our work is close to [9] in that our proximity model also integrates proximity measures into probabilistic models by pseudo term frequency. However, our model is different in that we do not detect spans nor calculate pseudo term frequency by span scores. Our proximity model also differs from [11][4] because we integrate proximity factors into a probabilistic model, rather than a language model.

3 A Proximity Probabilistic Model

3.1 Integrating Proximity into the BM25 Model

In this section, we discuss how to integrate the proximity of query terms in a document into the BM25 retrieval model. Similar to the ideas proposed in [9][4][11], the proximity information could be seen as being transformed to word count information, which is the primary factor that the unigram probabilistic model takes advantage of. Consider a document d represented by:

$$d = (t_1, t_2, \cdots, t_l)$$

where t_i is the ith term occurrence in document d. In the bag-of-words model, the terms are assumed independent to each other. Thus any term occurrence counts one in its corresponding term frequency, independent of other terms. For example, term frequency of a query term q_j is calculated as:

$$tf_j = \sum_{t_i = q_j} 1$$

The main idea of a proximity probabilistic model is to estimate a position-dependent term count $f(t_i)$, which is related not only to the number of occurrences of t_i but also to the term counts propagated from other terms, as defined in Section 3.2. Therefore, a document is transformed to a pseudo bag-of-words document represented as follows:

$$d_p = (f(t_1), f(t_2), \cdots, f(t_l))$$

Based upon such a proximity based document representation, the problem of ranking d with respect to q is changed to ranking d_p. Our proximity probabilistic model estimates the relevance score as follows:

$$RS_p = \sum_{q_i \in q} w_i * \frac{tfp_i}{K + tfp_i}.$$

where tfp_i is a pseudo term frequency based on d_p calculated as follows:

$$tfp_i = \sum_{t_j = q_i} f(t_j)$$

3.2 Position-Dependent Term Count

The key problem in the proximity probabilistic model is how to estimate the position-dependent term count $f(\cdot)$. Since in the retrieval model a term other than a query term does not influence relevance scores, we ignore non-query terms in the document representation. For example, suppose a query "ABC" and a document as follows:

$$B_1 A_1 X X X A_2 B_2 C_2 X X X X X X X X X X X X A_3 X$$

Here, X denotes a term other than a query term, and A_i represents that the ith occurrence of the query term A.

In this paper, we consider the following heuristics in measuring term proximity:

1. Distance: The closer two query terms are, the more a term's count propagates to another term. For example, A receives more propagated word counts from B in "AB" than from B in "AXXXXXB".
2. Order: A receives more propagated word counts from B in a pair "AXXB" which preserves the same order between A and B as in the query, than in a pair "BXXA" which does not preserve the order.
3. Term Weight: Stronger terms propagate higher term counts. For example, if $w_B > w_C$, A receives more word counts from B in "AB" than from C in "AC".

Now we present how we integrate these proximity intuitions in $f(\cdot)$. When calculating f for a term occurrence t_i, we regard the term as the central term. Then we search forwards and backwards respectively for another query term. In practice, we stop searching when the distance is larger than $MaxDist$. Whenever we find another different key term, denoted as t_j, we define the distance between t_i and t_j as the number of steps one has to move t_j in order to obtain the same relative positions for t_i and t_j as in the query. Specifically, we define the distance as follows:

$$dist(t_i, t_j) = |(p(t_i, d) - p(t_j, d)) - (p(t_i, q) - p(t_j, q))|.$$

where $p(t, d)$ is the position where term t occurs in the document d, and $p(t, q)$ is the position where t occurs in the query q. For example,

$$\begin{aligned} dist(A_2, B_2) &= |(-1) - (-1)| = 0; \\ dist(A_2, C_2) &= |(-2) - (-2)| = 0; \\ dist(A_1, B_1) &= |1 - (-1)| = 2. \end{aligned}$$

As a result, we punish the unordered term pair in a natural way. e.g., $dist(A_1, B_1)$ is larger by two than $dist(A_2, B_2)$.

Next, we define a proximity kernel function g to transform $dist$ into a score. Four different functions g are used:

1. Gaussian:
$$g(x) = e^{-\frac{x^2}{2a^2}};$$
2. Linear:
$$g(x) = a * x + 1, a < 0;$$
3. Parabola:
$$g(x) = a * x^2 + 1, a < 0;$$
4. Reverse:
$$g(x) = 1/(a * x + 1);$$

The four kernel functions represent different ways in which the propagated term count decreases when $dist$ increases. For example, the linear function means that the propagated term count from term t_j decreases linearly. The reverse function means that the propagated term count from t_j decreases rapidly when $dist(t_i, t_j)$ is small while it decreases slowly when $dist(t_i, t_j)$ gets large. The first two kernels have been used by Lv and Zhai [4], and the parabola kernel is also similar to the circle kernel in their work. The unique kernel function that we try is the reverse function. It is different from the other three because it is concave.

Finally, by considering term weights and handling multiple occurrences of the same term, we obtain the final f as follows:

$$\begin{aligned} f(t_i) &= c + \textstyle\sum_{j \neq i} h(t_i, t_j) \\ h(t_i, t_j) &= w_1 * w_2 * g(\min\{dist(t_i, t_j)\}). \end{aligned}$$

Table 1. Basic statistics on query sets from TREC Terabyte Tracks

	04 adhoc	05 adhoc	06 adhoc
queries	701-750	751-800	801-850
#qry (with qrel)	49	50	50
#total qrels	10617	10407	5893
#qry of two terms	13	13	12

where c is a constant term count that t_i contributes to itself. When no other query terms appear around t_i, the default $f(t_i)$ is c, instead of zero. When more than one t_j occurs near t_i, we use the one with the smallest $dist$ to calculate h. For t_i, f sums the term count of t_i and those propagated from other key terms. If $MaxDist = 10$, for the earlier example, we have:

$$f(A_1) = c + h(A_1, B_1)$$
$$f(A_2) = c + h(A_2, B_2) + h(A_2, C_2)$$
$$f(A_3) = c.$$

and the proximity based term frequency for A is:

$$tfp_A = f(A_1) + f(A_2) + f(A_3)$$

This new term frequency is used to replace the raw term frequency in BM25. According to the function g used, four proximity probabilistic models (PPM) are thus defined: PPM-G, PPM-L, PPM-P, and PPM-R.

4 Experiments

We use the query sets of TREC Terabyte tracks to evaluate our proposed PPMs and the baseline BM25. Only the title fields of TREC topics are used as queries. Table 1 shows some basic information on these data sets. The document collection is GOV2 consisting of about 25 million pages. To avoid the uncertainty of combining different fields, we do not index the title, URL, or anchors of a page. We conduct retrieval on the text parsed from an HTML page's body or a PDF file. We perform stemming with Porter stemmer, but do not remove stop words. In each experiment, we first use the BM25 retrieval model to retrieve 3,000 documents for each query, and then apply the PPM or the BM25 with new parameters to re-rank them. The top 1,000 documents are finally used to evaluate all runs using Mean Average Precision (MAP). To tune the parameters of BM25 and PPM, we use the Terabyte'04 set for training and the other two sets for testing. The results are shown in Table 2. Results marked with "*" mean that the differences with the baseline are statistically significant in t-test at the level of $p < 0.05$.

The results show that proximity information is useful to improve the bag-of-words model in terms of overall MAP. For example, the PPM with the Gaussian kernel (PPM-G) performs well on the data set of 04 query set, and the PPM

Table 2. Comparing BM25 and our proposed PPM

	04 adhoc	05 adhoc	06 adhoc
BM25	0.262	0.314	0.288
PPM-G	0.278*	0.326	0.309*
PPM-L	0.269	0.329	0.317*
PPM-P	0.274*	0.326	0.309*
PPM-R	**0.280***	**0.329**	**0.319***
PPM-R over BM25	6.87%	4.76%	11.0%

with the linear kernel (PPM-L) performs well over 05 and 06 query sets. The best kernel is the reverse function (PPM-R) that we propose in this study (which has not been used in the previous work [4]). It consistently outperforms BM25 on three query sets, and the improvements over 04 and 06 query sets are statistically significant. This indicates that the reverse function we propose better captures the relationship between proximity and relevance than the other functions. More specifically, the positions nearby a key term have the highest impact on the estimation of the probability of relevance, whereas the influence from more distant positions is low and its form vs. distance tends to be flat.

5 Conclusion

In this paper, we propose a proximity probabilistic model (PPM) based on the BM25 retrieval model. The model estimates a position-dependent term count by considering heuristics on the distance between the query term and its surrounding query terms, their order, and term weights. In transforming the distance into a measure of relevance, we have tried four kernel functions. Among them, the reverse function has not been experimented with yet in previous works. Finally, we replace term frequency by pseudo term frequency based on position-dependent term counts and update the BM25 model to the PPM models. Experimental results show that the PPM model with the reverse kernel consistently outperforms the BM25 model across different TREC data sets. The reverse function performs better than the other three functions proposed in previous works. This indicates that the impact of a surrounding query term to the query term decays rapidly when the distance is small and then goes flat when the distance is large.

References

1. S. Buttcher, C. L. A. Clarke, and B. Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. In *Proceedings of SIGIR'06*, pages 621–622, 2006.
2. C. L. A. Clarke, G. V. Cormack, and F. J. Burkowski. Shortest substring ranking (multitext experiments for trec-4). In *Proceedings of TREC-4*, pages 295–304, 1995.
3. D. Hawking and P. B. Thistlewaite. Proximity operators - so near and yet so far. In *Proceedings of TREC-4*, 1995.

4. Y. Lv and C. Zhai. Positional language model for information retrieval. In *Proceedings of SIGIR'09*, pages 299–306, 2009.
5. D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of SIGIR'05*, pages 472–479, 2005.
6. Y. Rasolofo and J. Savoy. Term proximity scoring for keyword-based retrieval systems. *F. Sebastiani (Eds.): ECIR 2003, LNCS 2633*, pages 207–218, 2003.
7. S. E. Robertson and K. S. Jones. Relevance weighting for search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.
8. S. E. Robertson, S. Walker, and M. Beaulieu. Experimentation as a way of life: Okapi at trec. *Information Processing and Management*, 26(1):95–108, 2000.
9. R. Song, M. J. Taylor, J.-R. Wen, H.-W. Hon, and Y. Yu. Viewing term proximity from a different perspective. In *C. Macdonald et al. (Eds.): ECIR 2008, LNCS 4956*, pages 346–357, 2008.
10. T. Tao and C. Zhai. An exploration of proximity measures in information retrieval. In *Proceedings of SIGIR'07*, pages 295–302, 2007.
11. J. Zhao and Y. Yun. A proximity language model for information retrieval. In *Proceedings of SIGIR'09*, pages 291–298, 2009.