Applying Maximum Entropy to Known-Item Email Retrieval

Sirvan Yahyaei and Christof Monz

Department of Computer Science
Queen Mary, University of London
London E1 4NS, UK
{sirvan,christof}@dcs.qmul.ac.uk

Abstract. It is becoming increasingly common in information retrieval to combine evidence from multiple resources to compute the retrieval status value of documents. Although this has led to considerable improvements in several retrieval tasks, one of the outstanding issues is estimation of the respective weights that should be associated with the different sources of evidence. In this paper we propose to use maximum entropy in combination with the limited memory LBFG algorithm to estimate feature weights. Examining the effectiveness of our approach with respect to the known-item finding task of enterprise track of TREC shows that it significantly outperforms a standard retrieval baseline and leads to competitive performance.

1 Introduction

In several information retrieval tasks, such as web retrieval [15], structured document retrieval [7] and email retrieval [3], a number of approaches combine evidence from multiple resources to compute the retrieval status values.

Typically the different sources of evidence include term frequencies within different fields of a document (e.g., body and anchor text), different ways to compute within-document and collection term frequencies or the combination of different document similarity functions as a whole. Zobel and Moffat [16] show that it is very difficult to find a similarity measure which is best in all cases, but at the same time they show that there is still a lot of room for improvement by varying retrieval strategies.

Unfortunately, most of the evidence formulas and combining functions that have been developed were tuned by heuristic approaches. Thus, an approach which combines evidence from different representations of the documents and automatically estimates the importance of each component in the retrieval ranking function can be very useful. For this purpose, we have adapted maximum entropy, a statistical machine learning method, to perform the retrieval task. This paper contains a description of the method and a number of experiments to verify its effectiveness.

The remainder of this paper is organized as follows: In the next section the problem of combining evidence and document representations is introduced. The maximum entropy method and its adaptation to IR are provided in sections 3 and 4. Related work is discussed in section 5. In section 6 we discuss the experimental set-up and results. Section 7 concludes the paper.

2 Combining Evidence

The problem of evidence combination can be re-formulated as the problem of finding a ranking function $W(\mathbf{d}, q, \mathbf{C})$, where collection $\mathbf{C}$ contains a set of documents $\mathbf{d}$ with k fields $\{f_1, f_2, ..., f_k\}$. As mentioned, most of the work in this area is in the form of combining scores, particularly, linear combination of scores. However, Ogilvie and Callan [11] have shown that their mixture language model approach outperforms various meta-search methods in almost all cases. Moreover, Robertson et al. [14] have discussed the dangers of linear combination of entire document similarity scores and criticized it in detail.

To deal with the problem of combining evidence, we propose a method that addresses most of the issues in previous approaches. This method, was designed to have following features:

1. Automatically learn different features from different sources of evidence
2. Learn complex features such as term proximity or user preferences in a manner similar to simple features
3. Do not make assumptions which are not realistic, for example, term independence assumption due to mathematical convenience by many methods
4. Deal with documents with both single and multiple representations in a unified manner

3 Maximum Entropy Modeling

Statistical modeling is used to build a model to predict the behavior of a process. A labeled training set is employed to learn a model predict future behavior of the process [1]. The first modeling task is feature selection and the second one is model selection. Firstly, a set of statistics is determined and then these statistics will be employed to construct an accurate model of the desired process.

One of the approaches to build that model is through maximum entropy modeling. The idea behind maximum entropy method is very simple: model all that is known and assume nothing about that which is unknown [1]. It means, choose a model consistent with all the facts, but otherwise as uniform as possible.

The probability distribution for the process based on maximum entropy has two characteristics: Firstly, it is in accordance with the constraints, secondly it is as uniform as possible.

It can be shown that there is a unique distribution that satisfies these constraints and it is always of the exponential form. Berger et al. [1] have shown that the solution has the following parametric form:

$$p_\lambda(y|x) = \frac{1}{Z_\lambda(x)} \exp \sum_{i=1}^{n} \lambda_i f_i(x, y) \tag{1}$$

$$Z_\lambda(x) = \sum_y \exp \sum_{i=1}^{n} \lambda_i f_i(x, y) \tag{2}$$

where $Z_\lambda(x)$ is a constraint to satisfy the requirement that $\sum_y p_\lambda(y|x) = 1$ for all x, because it is a probability distribution.

Except for simple problems, equation 1 cannot be solved analytically and numerical methods have to be used to find the optimal weights of the features. We decided to use Nocedal's limited-memory BFGS optimization algorithm [10] which is a very efficient and robust method to solve large scale optimization problems and significantly outperforms the other two optimization approaches we experimented with.

4 Maximum Entropy in IR

By viewing the IR problem as a classification task, it is possible to apply discriminative classifiers to it, such as a classifier based on maximum entropy modeling. The retrieval process output values are $r \in \mathcal{R} = \{R, \bar{R}\}$ which are affected by the contextual information from the collection, documents, and queries. The parametric form of the distribution, which is mentioned in section 3, can be expressed as the conditional probability $p(r|d, q)$ as follows:

$$p(r|d, q) = \frac{1}{Z_\lambda(d, q)} \exp \sum_{i=1}^{n} \lambda_i f_i(d, q, r) \tag{3}$$

$$Z_\lambda(d, q) = \sum_{r \in \{R, \bar{R}\}} \exp \sum_{i=1}^{n} \lambda_i f_i(d, q, r) \tag{4}$$

There are two classes of features: Firstly, *atomic features* for documents with a single representation, which are functions of different term frequency statistics in the collection, documents and queries. Secondly, statistics for various *representations* of documents which in our case amounts to the different sections of the text. Representations are combined with atomic functions to have real-valued numbers as value. Table 1 shows some of the atomic and complex features. As we evaluate our approach in the context of email retrieval, our documents are e-mail message and each of the features is applied to the different fields of an e-mail: the subject, body and the body of the replied messages.

For training the maximum entropy model we normally use a set of queries for each of which we take a number of relevant and non-relevant documents. However, in the known-item finding task, there is exactly one relevant document for each query and the remainder of the collection is considered non-relevant with respect to this query. Therefore, we have to repeat the relevant constraints as much as non-relevant examples or choose a small portion of non-relevant examples. Due to the large number of documents we decided to under-sample a set of non-relevant documents, also repeating the relevant examples to balance out the training set.

Table 1. Functions used as features in our maximum-entropy retrieval approach.

Name	Atomic Feature	Description
NTF	$\sum_{t \in Q \cap D} \log{(1 + \frac{tf(D,t)}{\vert D \vert})}$	Normalized term frequency
IDF	$\sum_{t \in Q \cap D} \log{\frac{N}{df(t)}}$	Inverse document frequency
CT	$\vert q_i \in Q \cap D \vert$	Number of common terms
ICF	$\sum_{t \in Q \cap D} \log{\frac{\vert C \vert}{tf(C,t)}}$	Inverse collection frequency
Name	Complex Feature	Description
NTF-ICF	$\sum_{t \in Q \cap D} \log{(1 + \frac{tf(D,t)}{\vert D \vert} \frac{C}{tf(C,t)})}$	Normalized $tf \times icf$
NTF-IDF	$\sum_{t \in Q \cap D} \log{(1 + \frac{tf(D,t)}{\vert D \vert} \frac{N}{df(t)})}$	Normalized $tf \times idf$
BM25	$\sum_{t \in Q \cap D} \frac{(k1+1) \cdot tf(D,t)}{tf(D,t) + k1 \cdot (1 - b + b \cdot \frac{\vert D \vert}{avgdl(C)})} \cdot \log{\frac{N - df(t) + 0.5}{df(t) + 0.5}}$	Okapi BM25
TP	$\sum_{t_i, t_j \in \mathcal{P}} \log(1 + (\min{\{distance(t_i, t_j)\}})^{-2} \cdot \frac{\vert C \vert}{tf(C,t_i) tf(C,t_j)})$	Term proximity
FO	$\sum_{t \in Q \cap D} \log{(1 + \frac{\vert D \vert}{firstOccurrencePos(t)} \frac{\vert C \vert}{tf(C,t)})}$	First occurrence position
TD	$\log{(1 + \frac{4}{td(D)})}$	Depth in thread

5 Related Work

Ogilvie and Callan [11] compare the effectiveness of meta-search methods for combining document representations with their language modeling retrieval approach. In particular, they compared rank-based and score-based meta-searching with a mixture language model approach, showing that the latter slightly outperforms the best meta-search algorithms.

Their mixture method uses a unigram language model where the language model θ_D is specified by $p(w|\theta)$. During retrieval, documents are ranked by $p(Q|\theta_D) = \prod_{|Q|}^{i=1} p(q_i|\theta_D)$, where θ_D is the language model estimated for document D, $|.|$ is the length function and q_i is the ith term in the query Q.

Their approach is very similar to ours, except that we use an exponential model. Moreover, Ogilvie and Callan's mixture language model is based on unigram language modeling assuming term independence.

Robertson et al. [14] use BM25 as scoring function, but they have mentioned that this is a general method that can be used for many other scoring functions. The linear combination of frequencies method, similar to mixture language model by Ogilvie and Callan, uses a linear combination of a single scoring function over the representations. However, we have shown that our proposed maximum entropy method can combine any scoring function in a unified manner. On the other hand, using advanced features in a linear combination of frequencies will not be as easy as integrating them into a maximum entropy approach.

Another difference between maximum entropy and the above approaches is the estimation of the optimal weights or parameters of the ranking functions. Ogilvie and Callan [12] did not mention any optimization algorithms for finding the appropriate feature weights. Robertson et al. [14] used grid search for finding the parameters of their function. On the other hand, there are a number of well-studied optimization algorithms such as IIS, and L-BFGS for maximum entropy.

There have been a few attempts to explore maximum entropy in IR. Cooper [2] applied maximum entropy to information retrieval. Kantor and Lee [5] explored the application of maximum entropy, but more recently ([6]) they reported low performance on large document collections.

Greiff and Ponte [4] showed that ranking formulas of the Binary Independence Model (BIR) and Combination Match Model (CMM) can be derived from the maximum entropy principle with suitable features. Nallapati [9] explored discriminative models for IR and applied maximum entropy and support vector machines to several ad-hoc retrieval test sets. However, because of the rather discouraging results in these tasks, he did not examine maximum entropy in other tasks such as web or email retrieval.

6 Experiments

As mentioned before, one of the benefits of using maximum entropy is its ability to automatically learn arbitrary features. Thus to demonstrate the effectiveness of our approach in the context of information retrieval, we evaluate it with respect to email retrieval as defined in the Known-Item Finding Task of TREC 2005's Enterprise Track [3]. As emails are structured documents containing a number of fields (subject line, body, quoted text, etc.) this task is well-suited to evaluate the effectiveness of retrieval approaches that combine different sources of evidence. The collection is the W3C corpus which contains 174,311 documents. 25 queries are provided for training purposes and 125 additional queries are set aside for testing only. In our experiment, only the 25 training queries are used to generate the training data and learn the weights of the features.

There are three official measures for evaluating TREC's known-item finding task. The primary measure is the *Mean Reciprocal Rank (MRR)* and the other two are *success at 10* ($S@10$), indicating whether a relevant document is ranked among the top 10 retrieved documents, and *success at infinity* ($S@inf$) indicating whether a relevant document had been retrieved at any rank [3]. The Okapi BM25 ranking function has been used as one of the baselines for this experiment. The best results for the parameters after several attempts were $b = 0.25$ and $k1 = 1.2$. For this baseline, documents are treated as they have only one representation, i.e. all fields are merged and documents are indexed with one field which contains the whole text of the e-mail message.

6.1 Features

Three categories of features are used in this experiment: Firstly, features based on term frequencies of different representation of e-mail messages such as NTF-ICF-S= $\sum_{t \in Q \cap D_s} \log \left(1 + \frac{tf(D_s,t)}{|D_s|} \frac{C_s}{tf(C_s,t)}\right)$. Secondly, we use position based features such as term proximity, phrase match and first occurrence position features. Lastly, we use query independent features such as message depth in the thread.

There are many different methods to calculate term proximities [13, 8]. Our term proximity feature computes the sum of minimum distances between term pairs. We chose this method of calculating proximity to avoid using features which carry similar information. For example, this term proximity metrics does not contain information about term frequency in the document.

$$\sum_{t_i, t_j \in \mathcal{P}} \log(1 + (\min\{distance(t_i, t_j)\})^{-2} \frac{|C|}{tf(C, t_i) tf(C, t_j)}) \tag{5}$$

where, $distance(t_i, t_j)$ returns the set of distances between terms t_i and t_j, $tf(C, t_i)$ is the collection frequency of term i and $\mathcal{P}$ is the set of all possible pairs of query terms.

Similar to term proximity, the phrase match feature computes the maximum length of an exact match between the query and the document. Thus, a phrase match of 3 terms has a greater value than three matches of length 2.

The position of the first query term in the document is another feature that is used in the experiment:

$$\sum_{t \in Q \cap D} \log\,(1 + \frac{|D|}{firstOccurrencePos(t)} \frac{|C|}{tf(C, t)}) \tag{6}$$

Thread depth feature is a query independent feature that is computed as follows:

$$\log 1 + \frac{4}{td(D)} \tag{7}$$

where, $td(D)$ is the depth of document D in the thread. The depth of emails is capped at level 4.

6.2 Results

Table 2 shows the best runs of the maximum entropy system, compared to the baseline systems. For statistical significance testing we used the two-sided Wilcoxon signed-rank test. All the fields are stemmed by the first two steps of Porter stemmer after stop-word removal. We use three baselines: a standard BM25 run, where all fields are merged (run 1), a maximum entropy run with a BM25 feature applied separately to the body and subject (run 2), and a maximum entropy run, using term frequency statistics only (run 3).

As the results show, the best maximum entropy based system significantly outperforms all baselines. Although the first occurrence position feature is somewhat unstable, it improved overall performance. In accordance with earlier approaches, our experiments show the importance of the subject field. Runs using only the subject field outperform runs using only the body and thread fields.

In general, the results show that our maximum entropy approach leads to strong results, substantially outperforming competitive baselines. Comparing runs 5 and 3 shows that the new, term-position based features, such as the term proximity and first occurrence features described above, lead to the best results.

Table 2. E-mail search results. $\cdot^*$ indicates whether the improvement with respect to each of the three baselines (runs 1–3) statistically significant at level $\alpha = 0.05$. S, B and T indicate the field that the function is applied to, which are subject, body and replies of the message in the thread, respectively.

	Run	Features	MRR	S@10	S@inf
1	Baseline	BM25 on S+B+T, b=0.25, k1=1.2	$0.483^{-,*,-}$	0.696	**0.976**
2	Baseline	BM25-S, BM25-B	$\mathbf{0.557}^{*,-,*}$	**0.728**	0.968
3	Baseline	NTF-ICF-S, NTF-ICF-B, NTF-ICF-T	$0.520^{-,*,-}$	0.68	0.968
4	MaxEnt	NTF-ICF-S, NTF-ICF-B, FO-B, PM-B, TP-S, TP-B, TD	$0.603^{*,-,*}$	**0.816**	0.944
5	MaxEnt	NTF-ICF-S, NTF-ICF-B, FO-B, PM-B, TP-S, TP-B	$\mathbf{0.609}^{*,*,*}$	0.800	**0.976**
6	MaxEnt	BM25-S, BM25-B, BM25-T, FO-B, PM-B, TP-S, TP-B	$0.587^{*,-,*}$	0.76	0.960
7	MaxEnt	BM25-S, BM25-B, FO-B, PM-B, TP-S, TP-B	$0.603^{*,-,*}$	0.808	**0.976**
8	MaxEnt	NTF-ICF-S, NTF-ICF-B, PM-B, TP-S, TP-B	$0.565^{-,-,*}$	0.768	**0.976**
9	MaxEnt	NTF-S, NTF-B, IDF-S, IDF-B, FO-B, PM-B, TP-S, TP-B	$0.554^{-,-,-}$	0.736	0.952

7 Conclusions

We have shown that maximum entropy can be applied successfully to known-item email retrieval leading to statistically significant improvements over various baselines.

The advantages of using maximum entropy are twofold: Firstly, it is easy to integrate and experiment with additional features that are more tailored towards the retrieval task at hand. Here, we used three additional term-position based feature functions, term proximity, first term occurrence, and phrase matching. Using these additional features resulted in the highest performance, and led to statistically significant improvements over a maximum entropy based retrieval system that did not use these features.

The second advantage of maximum entropy over a number of related evidence combination approaches concerns the problem of estimating the appropriate feature weights. Earlier work often estimated the feature ways in a rather ad-hoc way by just experimenting with a number of weight combinations or by applying grid search. Both approaches are likely to miss the optimal weights and therefore leading to sub-optimal performance. On the other hand, there are a number of well-established and well-studied feature weight optimization algorithms for maximum entropy.

Acknowledgments. This work has been supported, in part, by the Nuffield Foundation, Grant No. NAL/32720.

References

1. Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, 1996.
2. William S. Cooper. Exploiting the maximum entropy principle to increase retrieval effectiveness. *Journal of the American Society for Information Science*, 34(1):31–39, January 1983.
3. Nick Craswell, Arjen de Vries, and Ian Soboroff. Overview of the trec-2005 enterprise track. In *Proceedings of the 14th Text REtrieval Conference*, 2006.
4. Warren R. Greiff and Jay M. Ponte. The maximum entropy approach and probabilistic ir models. *ACM Trans. Inf. Syst.*, 18(3):246–287, 2000.
5. Paul B. Kantor and Jung Jin Lee. The maximum entropy principle in information retrieval. In *SIGIR '86: Proceedings of the 9th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 269–274, New York, NY, USA, 1986. ACM Press.
6. Paul B. Kantor and Jung Jin Lee. Testing the maximum entropy principle for information retrieval. *J. Am. Soc. Inf. Sci.*, 49(6):557–566, 1998.
7. Mounia Lalmas. Uniform representation of content and structure for structured document retrieval. In *20th SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence*, 2000.
8. Christof Monz. *From Document Retrieval to Question Answering.* PhD thesis, University of Amsterdam, 2003.
9. Ramesh Nallapati. Discriminative models for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 64–71, 2004.
10. Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980.
11. Paul Ogilvie and Jamie Callan. Combining document representations for known-item search. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 143–150, New York, NY, USA, 2003. ACM Press.
12. Paul Ogilvie and Jamie Callan. Experiments with language models for known-item finding of e-mail messages. In *Proceedings of the Fourteenth Text Retrieval Conference (TREC-14)*, 2005.
13. Yves Rasolofo and Jacques Savoy. Term proximity scoring for keyword-based retrieval systems. In *Advances in Information Retrieval: 25th European Conference on IR Research, ECIR 2003, Pisa, Italy, April 14-16, 2003. Proceedings*, page 79. Springer Berlin / Heidelberg, 2003.
14. Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple bm25 extension to multiple weighted fields. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49, New York, NY, USA, 2004. ACM Press.
15. Theodora Tsikrika and Mounia Lalmas. Combining evidence from web retrieval using the inference network model - an experimental study. *Information Processing & Management, Special Issue in Bayesian Networks and Information Retrieval*, 40(5):751–772, September 2004.
16. Justin Zobel and Alistair Moffat. Exploring the similarity space. *SIGIR Forum*, 32(1):18–34, 1998.