Beyond Bags of Words: Modeling Implicit User Preferences in Information Retrieval

Donald Metzler and **W. Bruce Croft**
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{metzler,croft}@cs.umass.edu

Abstract

This paper reports on recent work in the field of information retrieval that attempts to go beyond the overly simplified approach of representing documents and queries as bags of words. Simple models make it difficult to accurately model a user's information need. The model presented in the paper is based on Markov random fields and allows almost arbitrary features to be encoded. This provides a powerful mechanism for modeling many of the implicit constraints a user has in mind when formulating a query. Simple instantiations of the model that consider dependencies between the terms in a query have shown to significantly outperform bag of words models. Further extensions of the model are possible to incorporate even more complex constraints based other domain knowledge. Finally, we describe what place our model has within the broader realm of artificial intelligence and propose several open questions that may be of general interest to the field.

Introduction

Information retrieval, broadly defined, is the task of retrieving relevant information from a collection of items in response to a user's query. Depending on the task, the information items may come in the form of text documents, web pages, images, videos, music files, or some mixture of the aforementioned. Queries can also be expressed in many different forms, such as Boolean queries or those expressed in natural language. With the advent, and subsequent popularity of web search, users have become accustomed to generating short natural language queries. However, such queries are often ambiguous or poor approximations of what the user has in mind. This abstract mental representation is often referred to as the information need.

Information needs are often complex. They include what the user already knows and a set of constraints about the types of documents that are likely to be relevant. Examples of such constraints are: all query terms should appear within a close proximity to each other, documents matching subphrases within a query are likely to be more relevant than those that do not match any, and more recent documents should be ranked higher than older documents. These are just a small sample of all the constraints a user has in mind before formulating a query. Unfortunately, users are unable to explicitly express their full mental state using an interface that only accepts short natural language queries. Therefore, the retrieval system is burdened with taking the user's short query and implicitly infusing it with preferences the user did not explicitly request. Therefore, the better the system is at representing these implicit preferences, the more likely the user's information need will be satisfied.

Inferring user preferences from a few keywords is a difficult task. In fact, most state of the art retrieval models ignore this problem altogether and simply treat queries and documents as a *bag of words*. For example, consider the query *white house rose garden*. To a bag of words model, this query returns the same results as the query *white rose house garden*, which is not even closely semantically related to the original query. It would be desirable for a model to accurately represent that the user implicitly prefers documents that match *white house* and *rose garden* as exact phrases, which is not possible in a bag of words model. Due to their very nature, it is either not possible or not easy to represent many types of user preferences within such models.

In this paper, we review recent work that has been done to address the issue of representing various implicit preferences that are meaningful in information retrieval (Metzler & Croft 2005). We present a formally motivated statistical model based on Markov random fields that goes beyond bag of words approaches. Although the model was primarily developed to robustly model dependencies between query terms, it is general enough to allow for a wide range of constraints to be easily modeled.

The remainder of this paper is laid out as follows. We first describes the details of our model. We then provide a high level discussion of how our work can be broadly interpreted in the field of artificial intelligence. Finally, we summarize our results and conclude the paper.

The Model

In this section we detail our Markov random field model for information retrieval. Markov random fields (MRF), also known as undirected graphical models, are commonly used in machine learning to succinctly model a joint distribution over a collection of random variables. We use MRFs to model the joint distribution $P_\Lambda(Q, D)$ over queries Q and

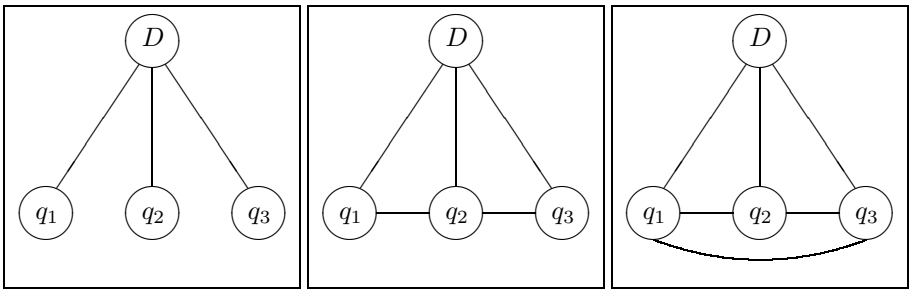

Figure 1: Example Markov random fields for three query terms constructed under various independence assumptions, including full independence (left), sequential dependence (middle), and full dependence (right).

documents D, parameterized by Λ.

Description

A Markov random field is constructed from a graph G. The nodes in the graph represent random variables, and the edges define the independence semantics between the random variables. In particular, a random variable in the graph is independent of its non-neighbors given observed values for its neighbors. Therefore, different edge configurations impose different independence assumptions. In this model, we assume G consists of query nodes q_i and a document node D, such as the graphs in Figure 1. Then, the joint distribution over the random variables in G is defined by:

$$P_\Lambda(Q, D) = \frac{1}{Z_\Lambda} \prod_{c \in C(G)} \psi(c; \Lambda)$$

where $Q = q_1 \dots q_n$, $C(G)$ is the set of cliques in G, each $\psi(\cdot; \Lambda)$ is a non-negative *potential function* over clique configurations parameterized by Λ and $Z_\Lambda = \sum_{Q,D} \prod_{c \in C(G)} \psi(c; \Lambda)$ normalizes the distribution. Note that it is generally infeasible to compute Z_Λ because of the exponential number of terms in the summation.

For ranking purposes we compute the conditional:

$$\begin{aligned} P_\Lambda(D|Q) &= \frac{P_\Lambda(Q, D)}{P_\Lambda(Q)} \\ &\stackrel{rank}{=} \log P_\Lambda(Q, D) - \log P_\Lambda(Q) \\ &\stackrel{rank}{=} \sum_{c \in C(G)} \log \psi(c; \Lambda) \end{aligned}$$

which can be computed efficiently for reasonable graphs.

Therefore, to utilize the model, the following steps must be taken for each query Q: 1) construct a graph representing the query term dependencies to model, 2) define a set of potential functions over the cliques of the graph, 3) rank documents in descending order of $P_\Lambda(D|Q)$.

Variants

Although any set of dependencies between query terms can be constructed, we consider three special cases that are of greatest interest. The three variants are *full independence* (FI), *sequential dependence* (SD), and *full dependence* (FD). Figure 1 shows graphical model representations of each.

The full independence variant makes the assumption that query terms q_i are independent given some document D, which is in line with the bag of words model. The sequential dependence variant assumes dependence between neighboring query terms. Models of this form, such as bigram or biterm models, are popular in information retrieval (Song & Croft 1999; Srikanth & Srihari 2002). Finally, the last variant we consider is the full dependence variant in which we assume all query terms are in some way dependent on each other. This model is an attempt to capture longer range dependencies than the sequential dependence variant. If such a model can accurately be estimated, it should be expected to perform at least as well as a model that ignores term dependence.

Potential Functions

The three model variants just described provide a way of modeling different dependencies between query terms. However, we still have not addressed the issue of how to encode implicit user preferences into the model. As we now show, this can be done using the potential functions.

There are two types of potential functions that directly affect how documents are ranked for a query. The first type consists of potentials over cliques that contain the document node and one or more query nodes. The potentials over these types of cliques can capture dependencies, semantic relationships, and other constraints between terms. For example, consider document D which is on the subject of artificial intelligence. A potential function should be constructed in such a way that $\psi(neural, network, D) > \psi(neural, surgeon, D)$, for example, by taking into account a variety of evidence and knowledge.

The other type is the potential that covers only the document node itself. In this case, implicit user preferences, such as preferring more recent documents to older documents, can be encoded. In this case, if document D_1 is an older document and document D_2 is newer, we could construct a potential function such that $\psi(D_1) < \psi(D_2)$.

By constructing meaningful potential functions, many implicit user preferences can be encoded into the model. Such potential functions can be based on semantic, syntactic, contextual, stylistic, or other types of evidence. This results in a more accurate representation of the user's information need.

	FI	SD	FD
AP	0.1775	0.1867* (+5.2%)	0.1866* (+5.1%)
WSJ	0.2592	0.2776† (+7.1%)	0.2738* (+5.6%)
WT10g	0.2032	0.2167* (+6.6%)	0.2231** (+9.8%)
GOV2	0.2502	0.2832* (+13.2%)	0.2844* (+13.7%)

Table 1: Mean average precision over a range of collections using each model variant. Values in parenthesis denote percentage improvement over full independence (FI) model. The symbols indicate statistical significance ($p < 0.05$ with a one-tailed paired t-test), where * indicates a significant improvement over the FI variant, ** over both the FI and SD variants, and † over the FI and FD variants.

Training

Training our model consists of learning the best setting for Λ given some training data. Since we are dealing with information retrieval, training data comes in the form of relevance judgments, which say whether or not some document is relevant to some query. Rather than learning a maximum likelihood or maximum *a posteriori* estimate, we choose to find the parameter setting that directly maximizes the information retrieval metric under consideration. We have derived a novel approach for carrying out such a maximization by hill climbing on the non-differentiable evaluation metric surface. Due to space limitations, we refer the reader to (Metzler 2005) for more details. We also note that a number of recent approaches have also been proposed to solve the problem of directly maximizing information retrieval-like metrics (Burges *et al.* 2005; Joachims 2005).

Summary of Results

In (Metzler & Croft 2005), we derive potential functions that build a simple model of term proximity. That is, we modeled the fact that users implicitly prefer query terms to appear within close proximity to each other within documents and that subphrases appearing within the query (such as *white house* and *rose garden* in our example) should also appear as phrases within relevant documents.

Table 1 summarizes our results on four data sets. The AP and WSJ data sets consist of news articles, whereas the WT10g and GOV2 data sets are very large (10GB and 426GB, respectively) collections of web documents. We note that the FI variant corresponds to a bag of words model. As we see, by modeling the proximities between terms with the SD and FD variants we are able to significantly improve effectiveness on every collection. For added evidence of the model's potential, we note that a slightly modified version of the model had either the best or second best results at the 2004 and 2005 TREC Terabyte Tracks (Metzler *et al.* 2004; 2005b), and the 2005 TREC Robust Track (Metzler *et al.* 2005a), which are international evaluations of information retrieval systems held yearly by NIST.

Based on these results, we feel that further implicit user assumptions and knowledge can be encoded into the model to yield even better performance.

Discussion

In this section we discuss several high level issues concerning knowledge representation in information retrieval.

Implicit vs. Explicit Representations

As described previously, a large amount of information is lost when an information need is transcribed into a query. This brings up the question of how much information a user should explicitly input to a search system and what information the system itself should implicitly extract from the query. There exists an interesting tradeoff between the burden put on the user and that put on the system. The more information a user is willing to input, the less intelligent the system has to be. However, since users are typically only willing to enter very short queries, the burden is typically left to the retrieval system.

Several retrieval systems exist that provide the user with a robust query language which allows the user to express their information need in greater detail. Two examples of such systems are Indri and its predecessor InQuery, which are based on the inference network retrieval model (Metzler & Croft 2004; Turtle & Croft 1991). The following is an example Indri query corresponding to the information need of locating the birthplace of George Washington:

```
#weight[sentence]( 2.0 #uw8( george washington )
                   1.0 born
                   1.0 #any:location )
```

which says "I want to find sentences that contain the terms George and Washington, in any order, within 8 words of each other (weighted 2), the term born (weighted 1), and any text indicative of a location (weighted 1)". As we see, this provides the user with a powerful tool for finding information by explicitly stating their preferences. However, this requires users to learn a query language, which many novice users may be unwilling to do. It is unclear if users of a commercial search engine would use such a powerful query language even if it significantly improved their user experience.

Therefore, as long as a gap exists between the information need and the query, there will be a need to build models that implicitly capture the preferences users are unwilling or unable to explicitly represent. We feel that the inference network model and our model provide an interesting set of tools for developing a better understanding of these issues.

Beyond Bags of Words

We have argued against the use of bag of words models and shown that modeling term proximity preferences can significantly improve retrieval effectiveness. It is worth looking into why such a model performs so much better than the bag of words approach.

As we showed with the *white house rose garden* example, permuting the terms leads to a semantically different query. Bag of words models are inherently deficient, in that they are incapable of capturing implicit concepts represented within the query. Both our work and the work of others (Mishne & de Rijke 2005) has shown that modeling these concepts,

teaching	#uw8(teaching children)
disabled	#uw8(disabled children)
children	#uw8(teaching disabled)
#1(disabled children)	#uw12(teaching disabled children)
#1(teaching disabled)	#1(teaching disabled children)

Table 2: Implicit concepts extracted for the query *teaching disabled children*, where #1 indicates the terms should appear as an exact phrase, and #uwN indicates the terms should appear, in any order, within a window of N terms.

via the use of proximity preferences, yields improvements in retrieval effectiveness. For example, for the query *teaching disabled children*, our model extracts the implicit concepts shown in Table 2. More details of how these concepts are extracted can be found in our previous work (Metzler & Croft 2005).

As we see, these implicit concepts represent the underlying information need better than a bag of words representation. We see that concepts such as #1(disabled children) and #uw8(teaching children) appear in the list, both of which are items the user probably had in mind while formulating the query, but was unable or unwilling to explicitly represent.

Even popular n-gram models are too rigid and fail to properly model all of the implicit concepts that our generalized model does. This is mostly due to the fact that n-gram models try to explain the *sequential* generation of text, whereas the idea of a concept is much more loosely defined in the context of the query. For this reason, non-sequential terms within a query, when combined together, may form useful concepts.

Therefore, we again see the importance of modeling implicit preferences. Queries, and texts in general, are filled with ambiguity, where there may be a large difference between what the author had in mind and what was actually written. We have shown that it is fruitful to consider modeling these implicit preferences in the context of information retrieval, but note that these observations may also be applicable to related areas, such as natural language processing and text classification.

Conclusions

In this paper we summarized a recently proposed information retrieval model based on Markov random fields. The model provides a robust framework for incorporating implicit preferences and knowledge that users have in mind when formulating a query, but are unable to explicitly express. We showed that when using the model to build a simple representation of term proximity we were able to achieve significantly better performance over standard bag of words models. We feel that building more knowledge into the system will only help further improve retrieval performance.

In this vein, there are several interesting open questions with regard to this model. First, can such a model be used to encode common sense knowledge into the search process? Next, what are the limits as to what can be represented in such a framework? Finally, how can the user (and their past experiences) be modeled in such a framework in order to infuse the model with personalized preferences?

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #CNS-0454018, and in part by NSF grant #IIS-0527159. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

References

Burges, C.; Shaked, T.; Renshaw, E.; Lazier, A.; Deeds, M.; Hamilton, N.; and Hullender, G. 2005. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, 89–96.

Joachims, T. 2005. A support vector method for multivariate performance measures. In *Proceedings of the International Conference on Machine Learning*, 377–384.

Metzler, D., and Croft, W. B. 2004. Combining the language model and inference network approaches to retrieval. *Information Processing and Management* 40(5):735–750.

Metzler, D., and Croft, W. B. 2005. A markov random field model for term dependencies. In *Proc. 28th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 472–479.

Metzler, D.; Strohman, T.; Turtle, H.; and Croft, W. B. 2004. Indri at trec 2004: Terabyte track. In *Proceedings TREC 2004*.

Metzler, D.; Diaz, F.; Strohman, T.; and Croft, W. B. 2005a. Umass robust 2005: Using mixture of relevance models for query expansion. In *Proceedings TREC 2005*.

Metzler, D.; Strohman, T.; Zhou, Y.; and Croft, W. B. 2005b. Indri at trec 2005: Terabyte track. In *Proceedings TREC 2005*.

Metzler, D. 2005. Direct maximization of rank-based metrics. Technical report, University of Massachusetts, Amherst.

Mishne, G., and de Rijke, M. 2005. Boosting web retrieval through query operations. In *Proc. 27th European Conf. on Information Retrieval*, 502–516.

Song, F., and Croft, W. B. 1999. A general language model for information retrieval. In *Proc. eighth international conference on Information and knowledge management (CIKM 99)*, 316–321.

Srikanth, M., and Srihari, R. 2002. Biterm language models for document retrieval. In *Proc. 25th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 425–426.

Turtle, H., and Croft, W. B. 1991. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems* 9(3):187–222.