Modeling Query Term Dependencies in Information Retrieval with Markov Random Fields

Donald Metzler METZLER@CS.UMASS.EDU
W. Bruce Croft CROFT@CS.UMASS.EDU
Department of Computer Science, University of Massachusetts, Amherst, MA 01003

Abstract

This paper develops a general, formal framework for modeling term dependencies via Markov random fields. The model allows for arbitrary text features to be incorporated as evidence. In particular, we make use of features based on occurrences of single terms, ordered phrases, and unordered phrases. We explore full independence, sequential dependence, and full dependence variants of the model. A novel approach is developed to train the model by directly maximizing mean average precision. Our results show that significant improvements are possible by modeling dependencies, especially on larger web collections.

1. Introduction

There is a rich history of statistical models for information retrieval, including the binary independence model (BIM), language modeling, and the inference network model, amongst others. It is well known that dependencies exist between terms in a collection of text. For example, within a *SIGIR* proceedings, occurrences of certain pairs of terms are correlated, such as *information* and *retrieval*. The fact that either one occurs provides strong evidence that the other is also likely to occur. Unfortunately, estimating statistical models for general term dependencies is infeasible, due to data sparsity. For this reason, most retrieval models assume some form of independence exists between terms. Some researchers even suggest modeling term dependencies is unnecessary as long as a good term weighting function is used (Salton & Buckley, 1988).

Most work on modeling term dependencies in the past has focused on phrases/proximity (Croft et al., 1991; Fagan, 1987) or term co-occurrences (van Rijsbergen, 1977). Most of these models only consider dependencies between pairs of terms. In (Fagan, 1987), Fagan examines how to identify and use non-syntactic (statistical) phrases. He identifies phrases using factors such as the number of times the phrase occurs in the collection and the proximity of the phrase terms. His results suggest no single method of phrase identification consistently yields improvements in retrieval effectiveness across a range of collections. For several collections, significant improvements in effectiveness are achieved when phrases are defined as any two terms within a query or document with unlimited proximity. That is, any two terms that co-occurred within a query or document were considered a phrase. However, for other collections, this definition proved to yield marginal or negative improvements. The results presented by Croft et. al. in (Croft et al., 1991) on the CACM collection suggest similar results, where phrases formed with a probabilistic AND operator slightly outperformed proximity phrases. Term co-occurrence information also plays an important role in the tree dependence model, which attempts to incorporate dependencies between terms in the BIM (van Rijsbergen, 1977). The model treats each term as a node in a graph and constructs a maximum spanning tree over the nodes, where the weight between a pair of terms (nodes) is the expected mutual information measure (EMIM) between them.

Other models have been proposed to capture dependencies between more than two terms, such as the Bahadur Lazarsfeld expansion (BLE) (Robert M. Losee, 1994), which is an exact method of modeling dependencies between all terms, and the Generalized Dependence Model that generalizes both the tree dependence model and the BLE expansion (Yu et al., 1983). Despite the more complex nature of these models, they have been shown to yield little or no improvements in effectiveness.

Several recent studies have examined term dependence models for the language modeling framework (Gao

This is a shortened, slightly modified version of the paper "A Markov Random Field Model for Term Dependencies," which appeared in the *Proceedings of SIGIR 2005*.

et al., 2004; Nallapati & Allan, 2002). These models are inspired by the tree dependence model and again only consider dependencies among pairs of terms. The model presented by Gao et. al in (Gao et al., 2004) showed consistent improvements over a baseline query likelihood system on a number of TREC collections. Unfortunately, the model requires computing a link structure for each query, which is not straightforward.

We formulate the following hypotheses: 1) dependence models will be more effective for larger collections than smaller collections, and 2) incorporating several types of evidence (features) into a dependence model will further improve effectiveness. Our first hypothesis is based on the fact that larger collections are noisier despite the fact they contain more information. As a result, independent query terms will match many irrelevant documents. If matching is instead done against more specific patterns of dependent query terms, then many of the noisy, irrelevant documents will be filtered out. In addition, we feel that considering various combinations of term features can yield further improvements over the previously researched methods, as it allows us to abstract the notions of dependence and co-occurrence.

The rest of this paper is presented as follows. In Section 2 we describe the details of our model and how training is done. Section 3 describes the results of ad hoc retrieval experiments done using our model on two newswire and two web collections, which show that dependence models can significantly improve effectiveness, especially on the larger web collections. Finally, in Section 4 we summarize the results and propose future research directions.

2. Model

In this section we describe a Markov random field approach to modeling term dependencies. Markov random fields (MRF), also called undirected graphical models, are commonly used in the statistical machine learning domain to succinctly model joint distributions. In this paper we use MRFs to model the joint distribution $P_\Lambda(Q, D)$ over queries Q and documents D, parameterized by Λ. We model the joint using this formalism because we feel it is illustrative and provides an intuitive, general understanding of different forms of term dependence.

Much like the language modeling framework, our model does not explicitly include a relevance variable. Instead, we assume there is some underlying joint distribution over queries and documents ($P_\Lambda(Q, D)$). Given a set of query and document pairs, Λ must be estimated according to some criteria. In the case of modeling relevance or usefulness, which we wish to do here, imagine there exists a list of query and document pairs. Suppose elements of this list are gathered from many (infinite) users of some information retrieval system who theoretically examine every pair of queries and documents. If a user finds document D relevant to query Q, then the pair (Q, D) is added to the list. This list can be thought of as a sample from some relevance distribution from which Λ can be estimated. We feel that ranking documents by $P_\Lambda(D|Q)$ upholds the original spirit of the Probability Ranking Principle (Robertson, 1977) under the assumption that the issuer of query Q is likely to agree with the relevance assessments of a majority of users. One could also estimate such a model for non-relevance or user-specific relevance, amongst others.

2.1. Overview

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. The joint distribution is uniquely defined by the graph G, the potential functions ψ, and the parameter Λ.

For ranking purposes we compute the conditional:

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

which can be computed efficiently for reasonable graphs.

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 this graph, 3) rank documents in descending order of $P_\Lambda(D|Q)$.

2.2. Variants

We now describe and analyze three variants of the MRF model, each with different underlying dependence assumptions. 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. That is, the likelihood of query term q_i occurring is not affected by the occurrence of any other query term, or more succinctly, $P(q_i|D, q_{j\neq i}) = P(q_i|D)$.

As its name implies, the sequential dependence variant assumes a dependence between neighboring query terms. Formally, this assumption states that $P(q_i|D, q_j) = P(q_i|D)$ only for nodes q_j that are not adjacent to q_i. Models of this form are capable of emulating bigram and biterm language models (Song & Croft, 1999; Srikanth & Srihari, 2002).

The last variant we consider is the full dependence variant. In this variant we assume all query terms are in some way dependent on each other. Graphically, a query of length n translates into the complete graph K_{n+1}, which includes edges from each query node to the document node D, as well. 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.

2.3. Potential Functions

The potential functions ψ play a very important role in how accurate our approximation of the true joint distribution is. These functions can be thought of as compatibility functions. Therefore, a good potential function assigns high values to the clique settings that are the most "compatible" with each other under the given distribution. As an example, consider a document D on the topic of *information retrieval*. Using the sequential dependence variant, we would expect $\psi(information, retrieval, D) > \psi(information, assurance, D)$, as the terms *information* and *retrieval* are much more "compatible" with the topicality of document D than the terms *information* and *assurance*.

It is also important that the potential functions can be computed efficiently. With this in mind, we opt to use potential functions that can be calculated using Indri[1], our new scalable search engine that combines language modeling and the inference network framework (Metzler & Croft, 2004).

Based on these criteria and previous research on phrases and term dependence (Croft et al., 1991; Fagan, 1987) we focus on three types of potential functions. These potential functions attempt to abstract the idea of term co-occurrence. In the remainder of this section we specify the potential functions used.

Since potentials are defined over cliques in the graph, we now proceed to enumerate all of the possible ways graph cliques are formed in our model and how potential function(s) are defined for each. The simplest type of clique that can appear in our graph is a 2-clique consisting of an edge between a query term q_i and the document D. A potential function over such a clique should measure how well, or how likely query term q_i describes the document. In keeping with simple to compute measures, we define this potential as:

$$\begin{aligned} \log \psi_T(c) &= \lambda_T \log P(q_i|D) \\ &= \lambda_T \log \left[(1-\alpha_D) \frac{tf_{q_i,D}}{|D|} + \alpha_D \frac{cf_{q_i}}{|C|} \right] \end{aligned}$$

where $P(q_i|D)$ is simply a smoothed language modeling estimate. Here, $tf_{w,D}$ is the number of times term w occurs in document D, $|D|$ is the total number of terms in document D, cf_w is the number of times term w occurs in the entire collection, and $|C|$ is the length of the collection. Finally, α_D acts as a smoothing parameter (Zhai & Lafferty, 2001). This potential makes the assumption that the more likely a term is given a document's language model, the more "compatible" the two random variables q_i and D are.

Next, we consider cliques that contain two or more query terms. For such cliques there are two possible cases, either all of the query terms within the clique appear contiguously in the query or they do not. The fact that query terms appear contiguously within a query provides different (stronger) evidence about the information need than a set of non-contiguous query terms. For example, in the query *train station security measures* (TREC topic 711), if any of the subphrases, *train station*, *train station security*, *station*

[1] Available at http://www.lemurproject.org

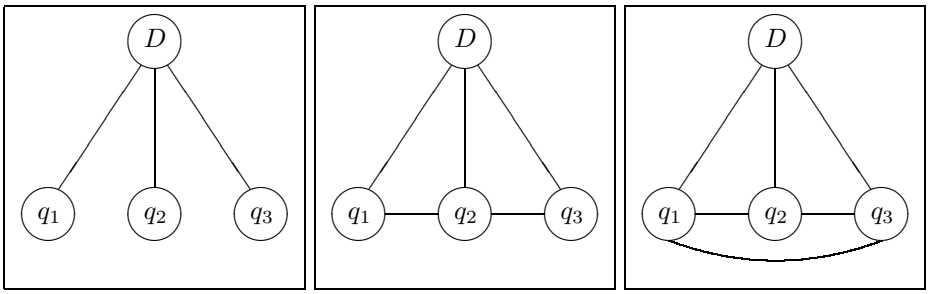

Figure 1. Example Markov random field model for three query terms under various independence assumptions. (left) full independence, (middle) sequential dependence, (right) full dependence.

security measures, or *security measures* appear in a document then there is strong evidence in favor of relevance. Therefore, for every clique that contains a contiguous set of two or more terms $q_i, \ldots, q_{i+k}$ and the document node D we apply the following "ordered" potential function:

$$\begin{aligned}\log \psi_O(c) &= \lambda_O \log P(\#1(q_i, \ldots, q_{i+k})|D) \\ &= \lambda_O \log \left[(1-\alpha_D) \frac{tf_{\#1(q_i \ldots q_{i+k}),D}}{|D|} + \alpha_D \frac{cf_{\#1(q_i \ldots q_{i+k})}}{|C|} \right]\end{aligned}$$

where $tf_{\#1(q_i \ldots q_{i+k}),D}$ denotes the number of times the exact phrase $q_i \ldots q_{i+k}$ occurs in document D, with an analogous definition for $cf_{\#1(q_i \ldots q_{i+k})}$. For more information on estimating so-called *language feature models* see (Metzler et al., 2004).

Although the occurrence of contiguous sets of query terms provide strong evidence of relevance, it is also the case that the occurrence of non-contiguous sets of query terms can provide valuable evidence. However, since the query terms are not contiguous we do not expect them to appear in order within relevant documents. Rather, we only expect the terms to appear ordered or unordered within a given proximity of each other. In the previous example, documents containing the terms *train* and *security* within some short proximity of one another also provide additional evidence towards relevance. This issue has been explored in the past by a number of researchers (Croft et al., 1991; Fagan, 1987). For our purposes, we construct an "unordered" potential function over cliques that consist of sets of two or more query terms $q_i, \ldots, q_j$ and the document node D. Such potential functions have the same form as the "ordered" potential functions, except #1 is replaced with #uwN, where $tf_{\#\text{uwN}(q_i \ldots q_j),D}$ is the number of times the terms $q_i, \ldots q_j$ appear ordered or unordered within a window N terms. In our experiments we will explore various settings of N to study the impact it has on retrieval effectiveness. It should also be noted that not only do we add a potential function of this form for non-contiguous sets of two or more query terms, but also for contiguous sets of two or more query terms. Therefore, for cliques consisting of contiguous sets of two or more query terms and the document node D we define the potential function to be the product $\psi_O(c)\psi_U(c)$, which itself is a valid potential function.

Using these potential functions we derive the following specific ranking function:

$$\begin{aligned}P_\Lambda(D|Q) &\overset{rank}{=} \sum_{c \in T} \log \psi_T(c) + \sum_{c \in O} \log \psi_O(c) \\ &+ \sum_{c \in O \cup U} \log \psi_O(c)\end{aligned}$$

where T is defined to be the set of 2-cliques involving a query term and a document D, O is the set of cliques containing the document node and two or more query terms that appear contiguously within the query, and U is the set of cliques containing the document node and two or more query terms appearing non-contiguously within the query. For any clique c that does not contain the document node we assume that $\psi(c) = 1$ for all settings of the clique, which has no impact on ranking. One may wish to define a potential over the singleton document node, which could act as a form of document prior.

2.4. Training

Given our parameterized joint distribution and a set of potential functions, the final step is to set the parameter values $(\lambda_T, \lambda_O, \lambda_U)$. These parameters are typically set using maximum likelihood or maximum *a posteriori* estimation. Margin-based methods exist for training MRFs, as well (Taskar et al., 2003).

However, two issues cause us to consider alternative training methodologies. First, the event space $Q \times D$ is large or even infinite depending on how it is defined.

	FI		SD		FD	
	AvgP	P@10	AvgP	P@10	AvgP	P@10
AP	0.1775	0.2912	0.1867* (+5.2%)	0.2980 (+2.3%)	0.1866* (+5.1%)	0.3068* (+5.4%)
WSJ	0.2592	0.4327	0.2776† (+7.1%)	0.4427 (+2.3%)	0.2738* (+5.6%)	0.4413 (+2.0%)
WT10g	0.2032	0.2866	0.2167* (+6.6%)	0.2948 (+2.9%)	0.2231** (+9.8%)	0.3031 (+5.8%)
GOV2	0.2502	0.4837	0.2832* (+13.2%)	0.5714* (+18.1%)	0.2844* (+13.7%)	0.5837* (+20.7%)
$(\tilde{\lambda}_T, \tilde{\lambda}_O, \tilde{\lambda}_U)$	(1.00, 0.00, 0.00)		(0.85, 0.10, 0.05)		(0.80, 0.10, 0.10)	

Table 1. Mean average precision and precision at 10 using optimal parameter settings for each model. 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. Suggested parameter values for each variant are also given.

Generally, the only training data available is a set of TREC relevance judgments for a set of queries. The documents found to be relevant for a query can then be assumed to be samples from this underlying relevance distribution. However, this sample is extremely small compared to the event space. For this reason, it is highly unlikely that a maximum likelihood estimate from such a sample would yield an accurate estimate to the true distribution. Next, it has been observed elsewhere that maximizing the likelihood will not necessarily also maximize the underlying retrieval metric (Morgan et al., 2004).

Therefore, we choose to train the model by directly maximizing mean average precision. Since our model only has three parameters, it is possible to do a parameter sweep to find the optimal parameters. However, such a parameter sweep can be computationally intensive. A few observations allow us to devise a relatively efficient training procedure. First, all features are assumed to provide positive evidence, and thus we can only consider positive parameter values. Second, the ranking function is invariant to parameter scale. That is, for some fixed K, rank order will be preserved if we modify the parameters such that $\hat{\lambda}_T = K\lambda_T$, $\hat{\lambda}_O = K\lambda_O$, and $\hat{\lambda}_U = K\lambda_U$, since the constant can always be factored out. Therefore, a simple coordinate-level hill climbing search over the simplex formed by the parameters is used to optimize mean average precision by starting at the full independence parameter setting ($\lambda_T = 1, \lambda_O = \lambda_U = 0$). More details can be found in (Metzler, 2005). Recently, other approaches to maximizing information retrieval metrics have been proposed, but are not explored here (Joachims, 2005; Burges et al., 2005).

3. Experimental Results

In this section we describe experiments using the three model variants. Our aim is to analyze and compare the retrieval effectiveness of each variant across collections of varying size and type. We make use of the Associated Press and Wall Street Journal subcollections of TREC, which are small homogeneous collections, and two web collections, WT10g and GOV2, which are considerably larger and less homogeneous.

All experiments make use of the Indri search engine (Metzler et al., 2004). Documents are stemmed using the Porter stemmer, but not stopped at index time. Instead, stopping is done at query time using a standard list of 421 stopwords. Only the title portion of the TREC topics are considered. The newswire queries are typically longer than the short, keyword-based web queries.

Due to space limitations, we only briefly summarize the results. Table 1 gives mean average precision, precision at 10, and suggested model parameters for each variant. The results given use the optimal parameter values to allow a fair comparison. Both the sequential and full dependence variants significantly improve mean average precision over the full independence variant for all four collections. Therefore, modeling dependencies between terms can be done consistently and can result in significant improvements. We also note the considerable improvements on the WT10g and GOV2 collections. These improvements support our hypothesis that dependence models may yield larger improvements for large collections. As further evidence of the power of these models on large collections, we note that a slightly modified version of the full dependence variant of this model was the best automatic, title-only run at both the 2004 and 2005 TREC Terabyte Tracks (Metzler et al., 2004; Metzler et al., 2005). Although not explored here, the P@10 results could likely be significantly improved by directly maximizing over the P@10 metric.

4. Conclusions

In this paper we develop a general term dependence model that can make use of arbitrary text features. Three variants of the model are described, where each captures different dependencies between query terms.

The full independence variant assumes that query terms are independent. The sequential dependence variant assumes certain dependencies exist between adjacent query terms, which is akin to bigram and biterm language models (Song & Croft, 1999; Srikanth & Srihari, 2002). Finally, the full dependence model makes no independence assumptions and attempts to capture dependencies that exist between every subset of query terms.

Our results show that modeling dependencies can significantly improve retrieval effectiveness across a range of collections. In particular, the sequential dependence variant using term and ordered features is more effective on smaller, homogeneous collections with longer queries, whereas the full dependence variant is best for larger, less homogeneous collections with shorter queries. In all cases, however, the sequential dependence variant closely approximates the full dependence variant. This provides the ability to tradeoff effectiveness for efficiency.

Directions of possible future work include exploring a wider range of potential functions, applying the model to other retrieval tasks, exploring different training methods including the use of clickthrough data, and constructing the graph G in other ways. For example, one could compute the EMIM between all pairs of query terms and only choose to model dependencies between terms with a high EMIM value. Or, similarly, one could apply the link approach taken in (Gao et al., 2004) to determine the important dependencies to model for a given query.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #CNS-0454018 , in part by Advanced Research and Development Activity and NSF grant #CCF-0205575, and in part by NSF grant #IIS-0527159. Any opinions, findings and conclusions or recommendations expressed in this material are those of 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., & Hullender, G. (2005). Learning to rank using gradient descent. *ICML '05: Proceedings of the 22nd international conference on Machine learning* (pp. 89–96).

Croft, W. B., Turtle, H. R., & Lewis, D. D. (1991). The use of phrases and structured queries in information retrieval. *Proc. 14th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (pp. 32–45).

Fagan, J. L. (1987). Automatic phrase indexing for document retrieval: An examination of syntactic and non-syntactic methods. *Proc. tenth Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (pp. 91–101).

Gao, J., Nie, J.-Y., Wu, G., & Cao, G. (2004). Dependence language model for information retrieval. *Proc. 27th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (pp. 170–177).

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

Metzler, D. (2005). *Direct maximization of rank-based metrics* (Technical Report). University of Massachusetts, Amherst.

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

Metzler, D., Strohman, T., Turtle, H., & Croft, W. B. (2004). Indri at terabyte track 2004. *Text REtrieval Conference (TREC 2004)*.

Metzler, D., Strohman, T., Zhou, Y., & Croft, W. B. (2005). Indri at terabyte track 2005. *Text REtrieval Conference (TREC 2005)*.

Morgan, W., Greiff, W., & Henderson, J. (2004). *Direct maximization of average precision by hill-climbing with a comparison to a maximum entropy approach* (Technical Report). MITRE.

Nallapati, R., & Allan, J. (2002). Capturing term dependencies using a language model based on sentence trees. *Proc. of the 2002 ACM CIKM International Conference on Information and Knowledge Management* (pp. 383–390).

Robert M. Losee, J. (1994). Term dependence: Truncating the Bahadur Lazarsfeld expansion. *Information Processing and Management*, *30*, 293–303.

Robertson, S. (1977). The probability ranking principle in IR. *Journal of Documentation*, *33*, 294–303.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, *24*, 513–523.

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

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

Taskar, B., Guestrin, C., & Koller, D. (2003). Max-margin markov networks. *Proc. of Advances in Neural Information Processing Systems (NIPS 2003)*.

van Rijsbergen, C. J. (1977). A theoretical basis for the use of cooccurrence data in information retrieval. *Journal of Documentation*, *33*, 106–119.

Yu, C. T., Buckley, C., Lam, K., & Salton, G. (1983). *A generalized term dependence model in information retrieval* (Technical Report). Cornell University.

Zhai, C., & Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. *Proc. 24th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (pp. 334–342).