An Improved Markov Random Field Model for Supporting Verbose Queries

Matthew Lease
Brown Laboratory for Linguistic Information Processing (BLLIP)
Brown University, Providence, RI USA
Center for Intelligent Information Retrieval (CIIR)
University of Massachusetts, Amherst, MA USA
mlease@cs.brown.edu

ABSTRACT

Recent work in supervised learning of term-based retrieval models has shown significantly improved accuracy can often be achieved via better model estimation [2, 10, 11, 17]. In this paper, we show retrieval accuracy with Metzler and Croft's Markov random field (MRF) approach [20] can be similarly improved via supervised learning. While the original MRF method estimates a parameter for each of its three feature classes from data, parameters within each class are set via a uniform weighting scheme adopted from the standard unigram. We conjecture greater MRF retrieval accuracy should be possible by better estimating within-class parameters, particularly for verbose queries employing natural language terms. Retrieval experiments with these queries on three TREC document collections show our improved MRF consistently out-performs both the original MRF and supervised unigram baselines. Additional experiments using blind-feedback [15] and evaluation with optimal weighting demonstrate both the immediate value and further potential of our method.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query Formulation

General Terms

Algorithms, Experimentation, Theory

1. INTRODUCTION

A document ranking method can be characterized by the model it defines and how its parameters are estimated. With classic term-based approaches, ranking is performed using a linear model computed over a feature space of lexical terms (often coupled with a document-specific prior) [24, 27, 29]. This simple feature set is remarkably expressive: a vast number of rankings are possible given different settings of the individual term weights. In contrast to this modeling expressiveness however, strategies for estimating term weights have traditionally been somewhat limited, and lack of statistical learning means estimation accuracy cannot automatically improve as more observational evidence becomes available. Consequently, recent work has begun exploring supervised approaches for estimating term-based models and shown significant improvement can often be achieved [2, 10, 11, 17].

Of course, language conveys far more information than simple term-based models are able to capture, and an important goal for long-term research is to develop richer models of language. A recent contribution in this direction was the development of a Markov random field (MRF) approach in which a standard unigram model is supplemented by two additional classes of lexical features: contiguous phrases and proximity [20]. While this approach was certainly not the first to use phrases or proximity (cf. [4, 6, 7, 22] inter alia), it incorporates them via a simple, principled framework that is efficient to compute and has been shown to consistently out-perform the standard unigram model across a range of TREC document collections [2, 20]. An important detail of the approach, however, is that although the weights for each feature class are learned from data, feature weights within each class are in fact estimated by the same uniform assumption as the standard unigram. This means that MRF estimation is similarly limited in modeling the varying importance of query terms. Recognizing this limitation, however, also reveals a potential opportunity to improve MRF accuracy by employing a similar supervised approach for parameter estimation as has already been successfully applied to unigram modeling [2, 10, 11, 17].

In this paper, we show this strategy is indeed viable: retrieval accuracy of the MRF model can be significantly increased by applying supervised learning. Our main results show that in comparison to using either the original MRF approach [20] or a supervised unigram model [17], integrating supervised unigram model estimation into the MRF yields significantly improved retrieval accuracy for verbose queries across three TREC document collections (§3.2). Our particular interest in supporting verbose queries is to improve document retrieval underlying question answering and other focused retrieval tasks. Additional experiments performed show the strength of our improved MRF under blind-feedback as well (§3.3). Finally, we evaluate model

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.

performance under optimal weighting of phrase and proximity features to demonstrate how their more accurate estimation also significantly improves retrieval (§3.4). This last experiment shows 3% absolute improvement over the baseline model can be achieved by assigning all phrasal and proximity weight to a single key dependency. In total, our results provide strong evidence that more accurate estimation of feature weights within each lexical class can significantly impact MRF model effectiveness. Results also motivate additional work exploring supervised estimation of feature weights for phrasal and proximity features alongside those of individual terms.

2. METHOD

This section describes our overall approach and its motivation. We begin by reviewing language model-based retrieval (§2.1). We discuss how canonical unigram estimation makes an implicit maximum-likelihood (ML) assumption that all query terms are equally important to the underlying information need, as well as why this is problematic for verbose queries. We then review Regression Rank [17], which applies supervised learning in place of ML to estimate more accurate, context-sensitive term weights (§2.2). Following this, we review the MRF retrieval model (§2.3). We show how parameter estimation for each of lexical feature class also implicitly adopts ML and so is similarly problematic for verbose queries. Finally, we describe how Regression Rank can be used to overcome this limitation.

2.1 Unigram Modeling

Of the three classic term-based approaches to retrieval [24, 27, 29], we adopt language modeling. Each observed document D is assumed to be generated by an underlying language model parameterized by Θ^D. Given an input query Q of length $|Q|$, we infer D's relevance to Q as the probability of observing Q as a random sample drawn from Θ^D. If we further assume bag-of-words modeling, Θ^D specifies a unigram distribution $\{\theta^D_{w_1} \dots \theta^D_{w_N}\}$ over the document collection vocabulary $V = \{w_1 \dots w_N\}$. Finally, letting f^Q_w denote the frequency of word w in Q, the likelihood of Q given D may be succinctly expressed as $log\, p(Q|D) = \sum_{w \in Q} f^Q_w \; log\, \theta^D_w = f^Q \cdot log\, \Theta^D$. This formulation is somewhat cumbersome, however, since the relative importance of query terms can only be expressed by their relative frequency. Fortunately, we can arrive at an equivalent and more convenient formulation by explicitly modeling the user's information need [12]. Specifically, we assume the observed Q is merely representative of a latent query model parameterized by $\Theta^Q = \{\theta^Q_{w_1} \dots \theta^Q_{w_V}\}$, consistent with intuition that the underlying information need might be verbalized in other ways than Q. Query likelihood may then be re-expressed in terms of Θ^Q's ML estimate $\widehat{\Theta^Q} = \frac{1}{|Q|} f^Q$:

$$f^Q \cdot log\, \Theta^D = |Q|\, \widehat{\Theta^Q} \cdot log\, \Theta^D \quad \overset{rank}{=} -\mathcal{D}(\widehat{\Theta^Q} || \Theta^D)$$

where $\overset{rank}{=}$ denotes rank-equivalence. This derivation shows that inferring document relevance on the basis of Q's likelihood given Θ^D has an alternative explanation of ranking based on minimal KL-divergence $\mathcal{D}(\Theta^Q || \Theta^D)$ between Θ^Q and Θ^D (assuming Θ^Q is estimated by ML). The significance of this in our context is showing query likelihood's implicit ML assumption that all query tokens are equally important to the underlying information need. While this assumption appears fairly benign for keyword queries, it is problematic for verbose queries because natural language terms greatly vary in their degree of correlation with the core information need. Fortunately, we see by this same token how retrieval accuracy might be improved by better estimation.

While estimation of both Θ^Q and Θ^D impacts retrieval accuracy, our focus in this paper is showing how better estimating Θ^Q in the MRF model (§2.3) can improve its retrieval accuracy on verbose queries. Consequently, we adopt standard Dirichlet-smoothed estimation of Θ^D, inferring $\hat{\theta^D_w}$ as a mixture of document D and document collection C ML estimates [32]: $\hat{\theta^D_w} = \lambda \frac{f^D_w}{|D|} + (1 - \lambda) \frac{f^C_w}{|C|}$, $\lambda = \frac{|D|}{|D|+\mu}$, where μ specifies a fixed hyper-parameter strength of the prior in smoothing. This reduces parameterization of unigram language modeling entirely to the query model Θ^Q.

2.2 Regression Rank

This section reviews Regression Rank [17], which applies supervised learning in place of ML to better estimate Θ^Q and thereby improve unigram retrieval accuracy. Given a set of training queries with relevant documents, an effective Θ^Q is estimated for each training query (§2.2.1). Secondary features correlated with Θ^Q are introduced to enable generalization (§2.2.2). Finally, a regression function is learned to predict Θ^Q for new queries using secondary features (§2.2.3).

2.2.1 Estimating the Query Model

A key idea of Regression Rank is that one can generalize knowledge of successful query models from past queries to predict effective query models for novel queries. In order to do this, we must have query models to generalize from. This requires a method for estimating a query model Θ^Q for each training query given examples of its relevant (and possibly non-relevant) documents. Essentially we want to perform massive explicit feedback [16] using training queries.

Following previous work [17], we apply grid search [21] to estimate an effective query model Θ^Q for each training query. Estimating the query model based on metric performance rather than likelihood avoids the issue of metric divergence [21] and makes it easy to re-tune the system later according to a different metric if so desired. A noteworthy detail concerns how the query model is estimated once search is complete. The easiest solution would be to simply pick the query model scoring highest according to a chosen metric for evaluating retrieval accuracy (e.g. mean-average precision). However, it turns out this is not the most effective strategy given the goal of enabling subsequent regression across queries (§2.2.3). The problem with simply picking the maximum is that the subsequent regression will be based on a single sample that may be drawn from a sharply-peaked local maximum on the metric surface. This would mean that were we to attempt to recover this parameterization via regression, small regression errors could yield a significant drop in metric performance. Instead, we estimate Θ^Q as the expected query model $\widehat{\Theta^Q} = \sum_s [\ \text{Metric}(\Theta_s)\Theta_s]$, a sum in which each sample query model Θ_s is weighted by the retrieval accuracy it achieved under the chosen accuracy metric (the distribution is left unnormalized due to ranking invariance). The intuition is this expectation should yield parameter values which perform well on average, likely corresponding to a smoother portion of the metric surface.

Finally, to provide a more stable basis for regression, we perform a non-linear normalization after which the expected query models fully span the interval $[0, 1]$. Previous work [17] reported this yielded consistent improvement.

2.2.2 Secondary Features

Given examples of past queries and corresponding inferred query models Θ^Q, Regression Rank uses secondary features correlated with Θ^Q and generalizing across queries to predict an appropriate Θ^Q for each novel query. This section summarizes the set of features used [17]. While existing features have proven effective, their simplicity suggests further improvement should be achievable via use of richer features.

Classic term frequency (tf) and document frequency (*df*) statistics feature prominently in the model. Two Key Concepts [2] features are also adopted: Google 1-gram tf [3] and residual inverse-*df* (*ridf*) statistics. These tf, *df*, and *ridf* statistics were collected from Gigaword [8] in addition to the target retrieval collections to provide robust estimates for general English. While we remove stop words prior to stemming to avoid accidental stemming collisions with the stop list, a stopword feature also provides a soft-test of whether stemmed terms appear in the stop list (§3). Position features correlate term importance with proximity to the start or end of the query string. Lexical features seek correlation of term importance with surrounding terms or punctuation; while many lexical features are instantiated during feature collection, few survive feature pruning (see below). A part-of-speech (POS) feature is also used given POS tags from a treebank parser [18] after detecting sentence boundaries [26].

Feature pruning discards any feature not observed at least a parameter η times. We set $\eta = 12$ following [17]. This significantly reduced the number of lexical features and generally helped filter out chance correlations from sparse features. Non-sparse features like tf which occur for every term were unaffected by pruning. Following previous work [9], feature values were normalized to the interval $[0, 1]$.

2.2.3 Inferring the Query Model via Regression

Given examples of target term weights paired with corresponding secondary features, the last stage of Regression Rank is to predict the query model given the features. This is accomplished via standard regularized linear regression.

Given N query terms in the training data, let $Y = \{y_{1:N}\}$ denote the target term weights and $\mathbf{X} = \{X_{1:N}\}$ the feature vectors. Next, let d denote the number (i.e. dimensionality) of features and $X_i = \{x_i^0, x_i^1 \dots x_i^d\}$ the ith feature vector (with $x_j^0 = 1$ by definition for all j). Also, let $W = \{w_0 w_1 \dots w_d\}$ denote the weight vector with w_0 as the bias term. Assuming $\mathbf{X}$ and Y are drawn from the joint distribution $p(X, y)$, our goal is to minimize expected loss given our prediction $f(X, W)$: $\mathbb{E}_{(X,y)\backsim p}[L(f(X,W), y)]$. Lacking oracle knowledge of $p(X, y)$, we approximate this with the empirical loss $\sum_i^N L(f(X_i, W), Y_i) = \sum_i^N (y_i - \sum_{j=1}^d w_j x_i^d)^2 = (Y - \mathbf{X}W)^T(Y - \mathbf{X}W)$ and minimize to find an optimal weight vector W^*. Conveniently, this *sum of least squares* optimization problem has a closed form solution: $W^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y$. However, since this ML solution often overfits, we alternatively revise the empirical loss formulation as $\sum_i^N L(f(X_i, W), Y_i) = (Y - \mathbf{X}W)^T(Y - \mathbf{X}W) + \beta W^T W$ where β defines a regularization parameter. This L2 (i.e. ridge) regression also has a closed-form solution: $W^* = (\beta I + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y$, where I denotes the identity matrix. Following previous work [17], we set $\beta = 1$.

2.3 The Markov Random Field Model

This section reviews the Markov random field (MRF) model of retrieval [20]. Our presentation shows how parameter estimation for each lexical feature class embodies the same implicit ML assumption underlying the standard unigram model. Finally, we describe how Regression Rank can be applied to more accurately estimate MRF term weights.

2.3.1 The Model

The MRF approach models the joint distribution $P_\Lambda(Q, D)$ over queries Q and documents D. It is constructed from a graph G consisting of a document node and nodes for each query term. Nodes in the graph represent random variables and edges define the independence semantics between the 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. 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 $C(G)$ is the set of cliques in G, each $\psi(\cdot; \Lambda)$ is a non-negative potential function over clique configurations parameterized by Λ, and $Z_\Lambda = \sum_{Q,D} \prod_{c \in C(G)} \psi(c; \Lambda)$ computes the partition function. For document ranking, we can skip the expensive computation of Z_Λ and simply score each document D by its unnormalized joint probability with Q under the MRF. If we define our potential functions as $\psi(c; \Lambda) = exp[\lambda_c f(c)]$, where $f(c)$ is some real-valued feature function over clique values and λ_c is that feature function's assigned weight, we can compute the posterior $P_\Lambda(D|Q)$ as

$$P_\Lambda(D|Q) = \frac{P_\Lambda(Q, D)}{P_\Lambda(Q)} \overset{rank}{=} \sum_{c \in C(G)} log\, \psi(c; \Lambda) = \sum_{c \in C(G)} \lambda_c f(c)$$

The graph G can be constructed in various ways depending on various possible assumptions regarding independence between terms. In the case of *full independence*, query term nodes share an edge with the document only. With *sequential dependence*, adjacent terms in the query share an additional edge in G. Finally, assuming *full dependence* constructs an edge between each pair of query term nodes. The choice of graph structure determines the set of cliques present in G and thereby the set of features used in ranking.

2.3.2 The Features

All of the potential functions used in the MRF can be expressed in the following generic form:

$$log\, \psi_i(c; \Lambda) = \lambda_i f_i(c) = \lambda_i log \left[(1 - \alpha_i^D) \frac{S_i(c)}{|D|} + \alpha_i^D \frac{S_i(c)}{|C|} \right]$$

where $S_i(c)$ denotes a given statistic computed for the given clique c, $|D|$ and $|C|$ indicate respective token counts of the document and entire collection (statistics other than term frequency are only approximately normalized), and $\alpha_i^D = \frac{\mu_i}{\mu_i + |D|}$, where μ_i denotes a smoothing hyper-parameter specific to the potential function $\psi_i(c; \Lambda)$ [32]. Note that use of term frequency as the statistic S_i computes the standard Dirichlet-smoothed unigram (§2.1).

Potential functions are primarily distinguished by the particular statistic S_i they employ. As mentioned earlier (§1), the MRF model exploits three classes of lexical features: individual terms, contiguous phrases, and proximity. Each of these corresponds to a distinct statistic S_i: term frequency, phrase frequency (i.e. "ordered" Indri #1 operator), and frequency of a set of terms within some parameter N-sized window (i.e. "unordered" Indri #uwN operator). The latter two multi-term statistics' corresponding potential functions are applicable when some form of dependency is assumed between query terms in the graph structure. In particular, the phrasal potential function is only applied to cliques connecting contiguous query terms, whereas the proximity potential function is applied to all multi-term cliques, contiguous and non-contiguous alike. This means each pair of contiguous query terms generates a clique c whose potential function is defined by the product $\psi_o(c)\psi_u(c)$ of ordered and unordered potential functions.

Using these three classes of potential functions, the MRF can be expressed as a three component mixture model computed over term, phrase, and proximity feature classes:

$$\sum_{c \in C(G)} \lambda_c f(c) \; = \; \sum_{c \in T} \lambda_T f_T(c) + \sum_{c \in O} \lambda_O f_O(c) + \sum_{c \in O \cup U} \lambda_U f_U(c)$$

Each class effectively computes its own ranking function which is then mixed with that of the other classes. For example, we saw above that the term ranking function is equivalent to the standard Dirichlet-unigram, meaning it embodies the same implicit ML assumption discussed earlier (§2.1) of estimating all class features as equally important to the underlying information need. Since all three classes can be expressed in the same generic form, phrasal and proximity classes also embody the same assumption.

Another way to see this is that all features within the same feature class are weighted by the same tied parameter λ_i. This reflects a choice of potential functions used rather than a general limitation of MRF modeling. We can generalize the model by instead assuming a unique potential function $\psi_i^c(c)$ for each clique rather than having a single function $\psi_i(c)$ for each feature class: $\psi_i(c) = \lambda_i \sum_{c \in i} \psi_i^c(c) = \sum_{c \in i} \lambda_i^c f_i^c$. The class-wide weighting parameter λ_i is preserved here simply for convenience. This generalized model is equivalent to the original under the condition that all clique-specific potential functions $\psi_i^c(c)$ within the same feature class adopt the same statistic S_i and use the same tied parameter $\lambda_i^c = \frac{1}{|c \in i|}$. We argue for breaking this parameter tying and applying supervised learning to estimate a unique weight for each clique to better model context-sensitivity.

2.3.3 Estimation with Regression Rank

We have just discussed how the MRF term component computes the standard Dirichlet-smoothed unigram. Consequently, Θ^Q is implicitly estimated by ML in the MRF as well to yield a uniform distribution over Q's terms. For example, we saw above that each clique is implicitly assigned uniform weight $\frac{1}{|c \in i|}$. This is problematic for verbose queries in which many terms appearing in the query are not strongly related to the core information need and should be assigned lower weight to improve retrieval effectiveness [2, 10]. A similarly striking effect for dependencies is observed in §3.4.

Fortunately, we saw in §2.2 that Θ^Q could be more accurately estimated by applying supervised learning. Instead of applying the MRF's default ML estimation of Θ^Q, we instead use Regression Rank. We adopt the generalized MRF having a distinct $\psi_T^c(c)$ for each clique; the same term frequency statistic is used across terms but the parameter λ_i^c is not tied. We then use our supervised estimate of Θ^Q to set λ_i^c values. This yields a more effective term component in the MRF with the potential of improving the overall MRF ensemble's retrieval accuracy. We evaluate this in §3.

While we do not apply supervised estimation of phrasal f_O and proximal f_U feature weights in this paper, results in §3.4 motivate future work in this direction. This might be achieved, for example, by applying Regression Rank to predict MRF rather than unigram parameters and extending its secondary feature set accordingly. In §4, we further discuss how the MRF model can be generalized beyond ways in which it has been historically used, as well as how better estimation of its parameters can enable us to take greater advantage of its full modeling power.

Collection	Content	# Docs	Topics
Robust04	Newswire	528,155	301-450, 601-700
W10g	Web	1,692,096	451-550
GOV2	Web	25,205,179	701-850

Table 1: Documents and topics used in experiments.

3. EVALUATION

This section presents empirical results measuring the impact of better MRF model estimation on document retrieval accuracy. Retrieval experiments are conducted using three TREC collections of varying size and content (Table 1). In order to improve document retrieval for verbose queries like those found in question answering and other focused retrieval tasks, evaluation primarily addresses use of TREC description queries. We use the *sequential dependence* MRF in our work since the *full dependence* MRF's combinatorial feature growth renders it intractable for use with verbose queries. An interesting topic for future work will be performing feature selection over all dependencies, sequential and non-sequential alike (§4).

Documents are ranked using Indri [31], with rankings scored using trec_eval 8.1[1]. Mean-average precision (MAP) serves as the primary metric, and results are as marked significant† ($p < .05$), highly significant‡($p < .01$), or neither according a non-parametric randomization test computed by Indri's ireval [28]. Experimental conditions reproduce those of previous work [2, 17] for fair comparison. Queries were stopped at query time using the same 418 word INQUERY stop list [1] and then Porter stemmed [25]. The same Dirichlet hyper-parameter $\mu_T = 1500$ was used for term features as well as Indri default values for μ_O and μ_U phrasal and proximity hyper-parameters. A window size of 8 tokens was used with the proximity feature.

3.1 Estimating MRF Component Weights

Recall that the MRF model uses three classes of lexical potential functions: individual terms $\psi_T(c)$, contiguous phrases $\psi_O(c)$, and proximity $\psi_U(c)$ (§2.3). These potential functions are parameterized by λ_T, λ_O, and λ_U weights specifying the relative importance of each lexical class in the overall MRF ensemble. In the original work [20], grid search was used estimate class weights using title queries over several document collections. Results showed an 85-10-5 mixing

[1] http://trec.nist.gov/trec_eval

Query	Model	Robust04	W10g	GOV2
Title	Base Unigram	25.32	19.49	29.61
Desc.	Base Unigram	24.51	18.61	25.22
	MRF [20]	25.64	19.14	27.40
	KC Unigram [2]	25.91	20.40	27.44
	RR Unigram [17]	27.33	22.01	27.35
	MRF+RR	$28.48^{\ddagger}_{\ddagger}$	$23.05^{\ddagger}_{\dagger}$	$29.51^{\ddagger}_{\ddagger}$

Table 2: Main results compare MAP retrieval accuracy of baseline MRF [2] and Regression Rank [17] models vs. their combination. Score^m_r superscripts and subscripts indicate statistical significance of the combined model vs. the MRF (m) and Regression Rank (r) baselines. Key Concepts [2] and cannonical unigram accuracy are also reported.

Query	Model	Robust04	W10g	GOV2
Title	Base Unigram	48.11	31.20	56.24
Desc.	Base Unigram	47.63	39.20	52.21
	MRF [20]	49.32	38.80	56.38
	KC Unigram [2]	47.55	41.40	57.05
	RR Unigram [17]	52.05	40.60	54.50
	MRF+RR	$54.30^{\ddagger}_{\ddagger}$	$42.00^{\dagger}_{\dagger}$	57.18

Table 3: Precision at top 5 ranks corresponding to same retrieval experiments shown in Table 2.

ratio (i.e. $\lambda_T = 0.85$, $\lambda_O = 0.10$, and $\lambda_U = 0.05$) generally performed well across collections.

We begin our evaluation by testing the optimality of these recommended λ_T, λ_O, and λ_U settings for verbose queries since earlier work applied the MRF's 85-10-5 mixing ratio to them without testing it [2, 17]. In comparison to title queries, verbose queries also exhibit more frequent syntactic relations between adjacent terms, and semantically-related terms often occur farther apart. Furthermore, the greater effectiveness of the supervised unigram in comparison to the maximum-likelihood (ML) unigram model used in the original MRF experiments suggested the unigram component here might merit additional weight in the mixture.

Consequently, we performed our own grid search over possible mixture ratios using development topics (§3.3). Despite any premonitions to the contrary, the 85-10-5 mixing ratio achieves MAP performance remarkably close to optimal: 24.79 vs. 24.93 for Robust04, 23.18 vs. 23.35 for W10g, and 26.68 vs. 27.01 for GOV2 (significance not reported). We therefore adopt the 85-10-5 ratio in our subsequent experiments for convenient comparison to previous work.

3.2 Estimating Term Feature Weights

This section presents our main results (Table 2) evaluating retrieval accuracy of the original MRF [20], Regression Rank unigram [17], and our combined model. Following previous work, Regression Rank was trained on each collection using 5-fold cross-validation. However, since the model was developed using only Robust04 (topics 301-450), further improvement of its performance and that of our combined model may also be possible for W10g and GOV2 collections via collection-specific model tuning.

Baseline performance of a standard unigram estimated by ML for both title and description queries shows that title queries consistently perform better than their description counterparts under ML estimation. While description queries are more informative to a human reader, additional terms introduced relative to title queries tend to individually correlate more weakly with the query's underlying core information need. Consequently, these terms should generally be assigned lower weight in estimation. ML's assumption that all observed query terms are equally important fails to do this, and retrieval accuracy suffers. The supervised estimation of Key Concepts [2] and Regression Rank [17] models addresses this limitation and is able to improve unigram retrieval accuracy as a result.

Our combined MRF model further exploits this better unigram estimation by leveraging it in conjunction with phrasal and proximity features. Across the three collections (Robust04, W10g, and GOV2), the combined model achieves absolute MAP improvements of 2.84% ‡ ($p < .0000$), 3.91% ‡ ($p = .0003$), and 2.11% ‡ ($p = .0003$) respectively vs. the original MRF. The number of queries improved, hurt or unchanged for each collection respectively are 166/83/0, 67/31/2, and 96/52/1. In comparison to the Regression Rank supervised unigram [17], absolute MAP improvements of 1.15% ‡ ($p < .0000$), 1.04% † ($p = .0282$), and 2.16% ‡ ($p < .0000$) are achieved. In this case, number of queries improved, hurt or unchanged are 151/96/2, 50/48/2, and 82/66/1.

Precision at early ranks also shows signs of improvement. For the top-5 retrieved documents, the combined model achieves absolute improvements of 4.98% ‡ ($p = .0001$), 3.20% † ($p = .0329$), and 0.80% respectively vs. the original MRF for Robust04, W10g, and GOV2, respectively. The number of queries improved, hurt or unchanged for each collection are 73/37/139, 32/17/51, and 36/38/85. In comparison to the Regression Rank supervised unigram [17], absolute precision improvements of 2.25% ‡ ($p = .0042$), 1.40%, and 2.68% are achieved. Here, the number of queries improved, hurt or unchanged are 52/29/168, 22/16/62, and 35/24/90.

3.3 Pseudo-relevance Feedback

This section reports retrieval accuracy of the original MRF model [20], Regression Rank [17], and our combined model under pseudo-relevance feedback (PRF). PRF was performed using Indri [31], which implements a variation on Lavrenko's relevance models [15]. Only unigram feature weights are re-estimated via PRF since previous work saw little benefit from PRF for re-estimating dependency feature weights [19]. Ten feedback documents were used, with estimated feedback document models truncated to the most probable 50 terms per document. The feedback model mixture weight was tuned on development topics: 301-450 for Robust04, 451-500 for W10g, and 701-750 for GOV2. This resulted in feedback model weights of 0.6, 0.1, and 0.3 for the three collections. Primary evaluation was performed on the remaining topics. Results appear in Table 4. Accuracy on all topics is also shown and allows comparison to earlier non-PRF results (Table 2).

For test set topics across the three collections, MAP accuracy of the combined model was improved by 2.10% ‡ ($p = .0001$), 1.42% † ($p = .0338$), and 2.55% ‡ ($p = .0001$) absolute vs. Regression Rank. The number of queries improved, hurt, or unchanged for each collection were 64/33/2, 24/26/0, and 58/41/1. In comparison to the baseline MRF model, MAP increased by 0.21%, 3.20% † ($p = .0252$), and 0.54%,

Model	Robust04		W10g		GOV2	
	Test	All	Test	All	Test	All
MRF [20]	38.92	30.09	19.99	20.02	32.37	30.26
RR [17]	37.03	30.52	21.77	22.48	30.36	28.96
MRF+RR	$39.13_{\ddagger}$	$31.82^{\ddagger}_{\ddagger}$	$23.19^{\dagger}_{\dagger}$	$23.05^{\ddagger}$	$32.91_{\ddagger}$	$31.20_{\ddagger}$

Table 4: MAP accuracy achieved by MRF [20], Regression Rank [17], and combined models for test and all topics using pseudo-relevance feedback. Statistical significance is reported as in Table 2.

with the number of queries improved, hurt, or unchanged being 44/55/1, 30/20/0, and 49/50/1. As for comparative precision at early ranks, we briefly summarize results. For the top-5 retrieved documents, differences are not significant with respect to the base MRF, but the combined model does achieve significantly better precision than Regression Rank across all collections (highly significant for Robust04).

Over all topics, the combined model is also seen to consistently perform best. While highly significant MAP improvement is achieved over both MRF ($\Delta = 1.73\%, p = .0012$) and Regression Rank ($\Delta = 1.30, p < .0000$) for Robust04, we see an alternation of highly significant improvement over MRF for W10g ($\Delta = 3.03, p = .0013$) and over Regression Rank for GOV2 ($\Delta = 2.24, p = .0001$) due to Regression Rank performing better for W10g while the base MRF model performs better for GOV2. Lacking a means of predicting which base model will perform better for which collection under PRF, the combined model is attractive in providing insulation from this alternation, performing at least as well as the stronger base model in either case. When both base models do perform well (e.g. Robust04), the combined model is seen to out-perform both of them.

3.4 Phrasal and Proximity Feature Weights

Thus far, results have addressed the impact of better estimating MRF term weights. We now report the impact of better estimating MRF phrasal and proximity parameters. Previous work has also explored use of co-occurrence and syntactic relationships in estimating these parameters for sentence retrieval [5].

Previous work generating all possible term subsets of verbose queries found retrieval accuracy could often be far improved by reducing queries to six or fewer terms [10, 11]. This inspired us to try a similar experiment for phrasal and proximity features (i.e. sequential dependencies). We evaluated dependency reductions of the base MRF model in which the default set of all sequential dependencies was similarly reduced to a subset of at most six dependencies. This is equivalent to performing a grid search [21] exploring possible binary assignments to these parameters. Other standard settings of the base MRF were kept fixed: 85-15-5 component weights along with the ML unigram weighting scheme.

Results in Table 5 show retrieval accuracy on Robust04 using a set of development topics (301-450). Statistical significance is not reported but can be safely assumed for the magnitude of improvements we discuss. The most striking observation is that inclusion of only the single most-helpful dependency improves MAP accuracy almost 3% absolute vs. the baseline model's default inclusion of all dependencies (i.e. ML estimation of dependency parameters). Furthermore, we see that adding a second best dependency provides no additional benefit, and that use of any greater fixed-sized subset of dependencies only serves to hurt performance vs. use of the single best dependency. Previous work modeling individual terms has similarly found that emphasizing one or two key terms in verbose queries also has the most significant impact on unigram retrieval accuracy [2]. It would be interesting to measure the degree to which key terms predicted in that work overlap with key dependencies found here. Results also show that if it were possible to simply identify the group of six most helpful dependencies without regard to their respective ordering, improvement of 1% could still be achieved vs. the baseline. Finally, we see upper-bound improvement of about 4% could be achieved by picking the optimal number of best dependencies to use for each query.

Dependencies	MAP	P@5
all (baseline)	21.10	43.84
1-best	24.02	50.27
2-best	24.05	51.37
3-best	23.67	51.10
4-best	23.11	49.18
5-best	22.73	48.49
6-best	22.27	47.12
oracle	25.49	55.07

Table 5: MAP retrieval accuracy of MRF model [20] under varying parameterization of phrasal and proximity features. The Robust04 collection was used with 146 description queries of length 20 or less (topics 301-450). Parameterizations were restricted to binary assignments of pair-wise sequential dependencies. Statistical significance is not shown.

Several details of this experiment merit cause for further optimism regarding the retrieval benefit of better estimating phrasal and proximity parameters. The grid search we performed considered only sequential dependencies; feature selection or weighting over the full cross-product of query dependencies (i.e. the full-dependency model) can only improve upon these results. Similarly, our grid search was restricted to binary assignments of parameters; more flexible weighting might also yield greater improvement. We also assumed fixed MRF component weights and ML estimation of phrasal and proximity parameters; additional relaxation of these assumptions may increase accuracy further.

3.5 Modeling Phrases vs. Proximity

This section describes a final simple experiment studying the effect of modeling ordered phrases vs. proximity. While previous work has shown these two distinct types of features provide complementary benefit to retrieval accuracy, we show here that at least in the case of modeling pair-wise sequential dependencies, nearly identical performance can be achieved across collections by modeling proximity only. Specifically, we replace the ordered `#1` Indri operator with the unordered `#uw2` proximal operator and leave other model settings unchanged. Results are shown in Table 6.

While proximity is still being matched at two different window sizes, results suggest the ordering-restriction is unnecessary under settings in which the MRF model is typically used in practice. Earlier work on biterm modeling

Feature Used	Robust04	W10g	GOV2
ordered #1	25.64	19.14	27.40
unordered #uw2	25.61	18.95	27.20

Table 6: MAP retrieval accuracy of the sequential-dependency MRF [20] on verbose queries using all topics. The standard MRF feature testing ordering of query term dependencies (#1) is seen to have negligible impact vs. order-ambivalent matching (#uw2). Usual 85-15-5 component weights, unigram weighting, proximal #uw8 features, and ML estimation of phrasal and proximal parameters is used.

similarly showed small differences in accuracy when employing ordering-restricted and ordering-ambivalent models [30]. This raises several interesting questions. Do phrasal vs. proximity features really provide distinct value, or are we merely observing a graduated effect of proximity at different window sizes? Important named-entities and collocations being matched may simply occur rarely enough in reversed order that the unordered feature approximates the ordered feature with reasonable accuracy. Would modeling a broader range of window sizes simultaneously be useful with smaller window size suggesting stronger dependencies? Will the utility of distinctly modeling phrases vs. proximity become more clearly marked as we more fully estimate the MRF model, using longer and non-sequential dependencies and abandoning ML estimation of feature weights? We plan to investigate these and related issues in future work.

4. DISCUSSION

We began this paper by emphasizing the distinction between model and estimation in evaluating a document ranking method's effectiveness. Lexical retrieval models are actually remarkably expressive but have typically not been estimated to their full potential. While recent work in *learning to rank* [9] has demonstrated a variety of new and effective retrieval models, the more sophisticated estimation techniques and additional features that typically go into these new models can also alternatively be employed to better estimate existing lexical models and function as a layer atop classic search engines [2, 10, 11, 17].

Consider the model and estimation method underlying classic language modeling [24] and probabilistic approaches [29]. Both can be viewed as constrained log-linear models adopting a specific feature set and restrictions on parameters. Unigram modeling can be viewed as a log-linear model in which the set of permissible parameterizations Λ is restricted to the probability distribution Θ^Q and the feature set F consists solely of the (log) document model Θ^D:

$$log\ p(Q|D) \propto \Theta^Q \cdot \Theta^D = \Lambda \cdot F$$

Building on the derivation in [13], we can similarly express the probabilistic approach as:

$$log\ \frac{p(D|Q,r)}{p(D|Q,\bar{r})} = |D|\ \Theta^D \cdot\ log\ \frac{p(w|Q,r)}{p(w|Q,\bar{r})} = F \cdot \Lambda$$

another constrained log-linear model where r and $\bar{r}$ denote relevant and non-relevant term distributions. Historically it has been a point of contention which of these two models should be preferred [14, 23]. However, if we accept Lavrenko's argument for dropping $|D|$ feature scaling on the grounds that concatenating a document with itself ought not to double its relevance score [14], both models utilize nearly identical features, differing by only a *log* factor, and are in fact rank-equivalent under equal parameterization. In short, we see the two approaches are constrained not by their models but by their fixed estimation strategies. Less constrained estimation would unlock greater modeling power.

We view the MRF approach (§2.3) as defining another such linear model which is more expressive than the ways it which has typically been used. We have discussed at length how the MRF has historically assumed one weight parameter per feature class: λ_T, λ_U, and λ_O. While parameter tying within each feature class certainly simplifies estimation, modeling power is reduced, and we have seen how breaking this parameter tying indeed has a positive effect on retrieval accuracy. The MRF variants for *full independence*, *sequential dependence*, and *full dependence* similarly provide a means of enforcing constraints on model sparsity to simplify estimation, but they represent only three fixed options out of an infinite space of possible continuous parameterizations. While it is impractical to model an exponential number of features at retrieval time, off-line methods for feature selection and estimation can be explored and subsequently applied to dynamically select and weight the most important features at run-time. Adopting the general linear model perspective of the model has the further benefit of enabling us to exploit the large body of existing techniques for maximizing such models, including recent work specifically targeting maximization of ranking metrics [9].

5. CONCLUSION

This paper addressed generalization and better estimation of Metzler and Croft's Markov random field (MRF) [20] approach to document retrieval. While the original MRF method estimated a parameter for each feature class from data, we showed how parameters within each class were implicitly estimated using the same maximum-likelihood assumption employed with the standard unigram. Because this scheme does not model context-sensitivity, its use particularly limits retrieval accuracy with verbose queries in which many terms appearing in the query are not strongly related to the core information need and so ought to be assigned lower weight. By employing supervised estimation instead, however, we showed this deficit could be remedied. Retrieval experiments conducted with verbose queries on three TREC document collections showed our improved MRF consistently out-performs both the original MRF and the supervised unigram model. Additional experiments using blind-feedback and evaluation with optimal weighting demonstrate both the immediate value and further potential of performing more accurate MRF model estimation. Future work will explore broader supervised estimation of the MRF model, addressing phrasal and proximity parameters in conjunction with term parameters.

Acknowledgments

I would like to thank James Allan and Michael Bendersky for valuable discussion, and the anonymous reviewers and my other colleagues for their useful feedback. Support for this work was provided in part by NSF PIRE Grant No OISE-0530118 and the Center for Intelligent Information Retrieval

(CIIR) at the University of Massachusetts Amherst. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

6. REFERENCES

[1] J. Allan, M. Connell, W. B. Croft, F. Feng, D. Fisher, and X. Li. INQUERY and TREC-9. In *Proc. of TREC-9*, pages 551–562, 2000.

[2] M. Bendersky and W. B. Croft. Discovering key concepts in verbose queries. In *Proc. of SIGIR*, pages 491–498. ACM New York, NY, USA, 2008.

[3] T. Brants and A. Franz. Web 1T 5-gram v1, LDC Catalog No. LDC2006T13, 2006.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[5] K. Cai, C. Chen, K. Liu, J. Bu, and P. Huang. MRF based approach for sentence retrieval. In *Proc. of SIGIR*, pages 795–796, 2007.

[6] C. Clarke, G. Cormack, and E. Tudhope. Relevance ranking for one to three term queries. *Information Processing and Management*, 36(2):291–311, 2000.

[7] J. Gao, J.-Y. Nie, G. Wu, and G. Cao. Dependence language model for information retrieval. In *Proc. of SIGIR*, pages 170–177, 2004.

[8] D. Graff, J. Kong, K. Chen, and K. Maeda. English Gigaword. *Linguistic Data Consortium catalog number LDC2005T12*, 2005.

[9] T. Joachims, H. Li, T.-Y. Liu, and C. Zhai. Learning to rank for information retrieval (lr4ir 2007). *SIGIR Forum*, 41(2):58–62, 2007.

[10] G. Kumaran and J. Allan. A Case for Shorter Queries, and Helping Users Create Them. In *Proceedings of NAACL HLT*, pages 220–227, 2007.

[11] G. Kumaran and J. Allan. Effective and efficient user interaction for long queries. In *Proc. of SIGIR*, pages 11–18, 2008.

[12] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proc. of SIGIR*, pages 111–119, 2001.

[13] J. Lafferty and C. Zhai. Probabilistic Relevance Models Based on Document and Query Generation. *Language Modeling for Information Retrieval*, pages 1–10, 2003.

[14] V. Lavrenko. *A Generative Theory of Relevance*. PhD thesis, University of Massachusetts Amherst, 2004.

[15] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th ACM SIGIR conference*, pages 120–127, 2001.

[16] M. Lease. Brown at TREC'08 Relevance Feedback Track. In *Proc. of the 17th Text Retrieval Conference (TREC)*, 2008.

[17] M. Lease, J. Allan, and W. B. Croft. Regression Rank: Learning to Meet the Opportunity of Descriptive Queries. In *Proc. of the 31st European Conference on Information Retrieval (ECIR)*, 2009. To appear.

[18] D. McClosky, E. Charniak, and M. Johnson. Effective self-training for parsing. In *Proc. of HLT-NAACL 2006*, pages 152–159, 2006.

[19] D. Metzler and W. Croft. Latent concept expansion using markov random fields. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 311–318. ACM Press New York, NY, USA, 2007.

[20] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *Proc. of SIGIR*, pages 472–479, 2005.

[21] D. Metzler and W. B. Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, 2007.

[22] G. Mishne and M. de Rijke. Boosting web retrieval through query operations. In *Proc. of ECIR*, 2005.

[23] R. Nallapati. *The Smoothed Dirichlet Distribution: Understanding Cross-Entropy Ranking in Information Retrieval*. PhD thesis, University of Massachusetts Amherst, 2006.

[24] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proc. of SIGIR*, pages 275–281, 1998.

[25] M. Porter. The Porter Stemming Algorithm. *Accessible at http://www. tartarus. org/martin/PorterStemmer.*

[26] J. C. Reynar and A. Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the fifth conference on Applied natural language processing*, pages 16–19, 1997.

[27] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proc. of SIGIR*, pages 21–29, 1996.

[28] M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proc. of CIKM*, pages 623–632, 2007.

[29] K. Sparck Jones, S. Walker, and S. Robertson. A probabilistic model of information retrieval: development and comparative experiments (parts i and ii). *Information Processing and Management*, 36:779–840, 2000.

[30] M. Srikanth and R. Srihari. Biterm language models for document retrieval. In *Proc. of SIGIR*, pages 425–426, 2002.

[31] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligence Analysis*, 2004.

[32] 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, 2004.