Predicting Diverse Subsets Using Structural SVMs

Yisong Yue YYUE@CS.CORNELL.EDU
Thorsten Joachims TJ@CS.CORNELL.EDU
Department of Computer Science, Cornell University, Ithaca, NY 14853 USA

Abstract

In many retrieval tasks, one important goal involves retrieving a diverse set of results (e.g., documents covering a wide range of topics for a search query). First of all, this reduces redundancy, effectively showing more information with the presented results. Secondly, queries are often ambiguous at some level. For example, the query "Jaguar" can refer to many different topics (such as the car or feline). A set of documents with high topic diversity ensures that fewer users abandon the query because no results are relevant to them. Unlike existing approaches to learning retrieval functions, we present a method that explicitly trains to diversify results. In particular, we formulate the learning problem of predicting diverse subsets and derive a training method based on structural SVMs.

1. Introduction

State of the art information retrieval systems commonly use machine learning techniques to learn ranking functions (Burges et al., 2006; Chapelle et al., 2007). Existing machine learning approaches typically optimize for ranking performance measures such as mean average precision or normalized discounted cumulative gain. Unfortunately, these approaches do not consider diversity, and also (often implicitly) assume that a document's relevance can be evaluated independently from other documents.

Indeed, several recent studies in information retrieval have emphasized the need to optimize for diversity (Zhai et al., 2003; Carbonell & Goldstein, 1998; Chen & Karger, 2006; Zhang et al., 2005; Swaminathan et al., 2008). In particular, they stressed the need to model inter-document dependencies. However, none of these approaches addressed the learning problem, and thus either use a limited feature space or require extensive tuning for different retrieval settings. In contrast, we present a method which can automatically learn a good retrieval function using a rich feature space.

In this paper we formulate the task of diversified retrieval as the problem of predicting diverse subsets. Specifically, we formulate a discriminant based on maximizing word coverage, and perform training using the structural SVM framework (Tsochantaridis et al., 2005). For our experiments, diversity is measured using subtopic coverage on manually labeled data. However, our approach can incorporate other forms of training data such as clickthrough results. To the best of our knowledge, our method is the first approach that can directly train for subtopic diversity. We have also made available a publicly downloadable implementation of our algorithm[1].

For the rest of this paper, we first provide a brief survey of recent related work. We then present our model and describe the prediction and training algorithms. We finish by presenting experiments on labeled query data from the TREC 6-8 Interactive Track as well as a synthetic dataset. Our method compares favorably to conventional methods which do not perform learning.

2. Related Work

Our prediction method is most closely related to the Essential Pages method (Swaminathan et al., 2008), since both methods select documents to maximize weighted word coverage. Documents are iteratively selected to maximize the marginal gain, which is also similar to approaches considered by (Zhai et al., 2003; Carbonell & Goldstein, 1998; Chen & Karger, 2006; Zhang et al., 2005). However, none of these previous approaches addressed the learning problem.

Learning to rank is a well-studied problem in machine learning. Existing approaches typically consider the one-dimensional ranking problem, e.g., (Burges et al.,

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

[1] `http://projects.yisongyue.com/svmdiv/`

2006; Yue et al., 2007; Chapelle et al., 2007; Zheng et al., 2007; Li et al., 2007). These approaches maximize commonly used measures such as mean average precision and normalized discounted cumulative gain, and generalize well to new queries. However, diversity is not considered. These approaches also evaluate each document independently of other documents.

From an online learning approach, Kleinberg et al. (2008) used a multi-armed bandit method to minimize abandonment (maximizing clickthrough) for a single query. While abandonment is provably minimized, their approach cannot generalize to new queries.

The diversity problem can also be treated as learning preferences for sets, which is the approach taken by the DD-PREF modeling language (desJardins et al., 2006; Wagstaff et al., 2007). In their case, diversity is measured on a per feature basis. Since subtopics cannot be treated as features (it is only given in the training data), their method cannot be directly applied to maximizing subtopic diversity. Our model does not need to derive diversity directly from individual features, but does require richer forms of training data (i.e., subtopics explicitly labeled).

Another approach uses a global class hierarchy over queries and/or documents, which can be leveraged to classify new documents and queries (Cai & Hofmann, 2004; Broder et al., 2007). While previous studies on hierarchical classification did not focus on diversity, one might consider diversity by mapping subtopics onto the class hierarchy. However, it is difficult for such hierarchies to achieve the granularity required to measure diversity for individual queries (see beginning of Section 6 for a description of subtopics used in our experiments). Using a large global hierarchy also introduces other complications such as how to generate a comprehensive set of topics and how to assign documents to topics. It seems more efficient to collect labeled training data containing query-specific subtopics (e.g., TREC Interactive Track).

3. The Learning Problem

For each query, we assume that we are given a set of candidate documents $\mathbf{x} = \{x_1, \ldots, x_n\}$. In order to measure diversity, we assume that each query spans a set of topics (which may be distinct to that query). We define $\mathbb{T} = \{T_1, \ldots, T_n\}$, where topic set T_i contains the subtopics covered by document $x_i \in \mathbf{x}$. Topic sets may overlap. Our goal is to select a subset $\mathbf{y}$ of K documents from $\mathbf{x}$ which maximizes topic coverage.

If the topic sets $\mathbb{T}$ were known, a good solution could be computed via straightforward greedy subset selection, which has a $(1 - 1/e)$-approximation bound (Khuller et al., 1997). Finding the globally optimal subset takes n choose K time, which we consider intractable for even reasonably small values of K. However, the topic sets of a candidate set are not known, nor is the set of all possible topics known. We merely assume to have a set of training examples of the form $(\mathbf{x}^{(i)}, \mathbb{T}^{(i)})$, and must find a good function for predicting $\mathbf{y}$ in the absence of $\mathbb{T}$. This in essence is the learning problem.

Let $\mathcal{X}$ denote the space of possible candidate sets $\mathbf{x}$, $\mathcal{T}$ the space of topic sets $\mathbb{T}$, and $\mathcal{Y}$ the space of predicted subsets $\mathbf{y}$. Following the standard machine learning setup, we formulate our task as learning a hypothesis function $h : \mathcal{X} \to \mathcal{Y}$ to predict a $\mathbf{y}$ when given $\mathbf{x}$. We quantify the quality of a prediction by considering a loss function $\Delta : \mathcal{T} \times \mathcal{Y} \to \Re$ which measures the penalty of choosing $\mathbf{y}$ when the topics to be covered are those in $\mathbb{T}$.

We restrict ourselves to the supervised learning scenario, where training examples $(\mathbf{x}, \mathbb{T})$ consist of both the candidate set of documents and the subtopics. Given a set of training examples, $S = \{(\mathbf{x}^{(i)}, \mathbb{T}^{(i)}) \in \mathcal{X} \times \mathcal{T} : i = 1, \ldots, N\}$, the strategy is to find a function h which minimizes the empirical risk,

$$R_S^{\Delta}(h) = \frac{1}{N} \sum_{i=1}^{N} \Delta(\mathbb{T}^{(i)}, h(\mathbf{x}^{(i)})).$$

We encourage diversity by defining our loss function $\Delta(\mathbb{T}, \mathbf{y})$ to be the weighted percentage of distinct subtopics in $\mathbb{T}$ not covered by $\mathbf{y}$, although other formulations are possible, which we discuss in Section 8.

We focus on hypothesis functions which are parameterized by a weight vector $\mathbf{w}$, and thus wish to find $\mathbf{w}$ to minimize the empirical risk, $R_S^{\Delta}(\mathbf{w}) \equiv R_S^{\Delta}(h(\cdot; \mathbf{w}))$. We use a discriminant $\mathcal{F} : \mathcal{X} \times \mathcal{Y} \to \Re$ to compute how well predicting $\mathbf{y}$ fits for $\mathbf{x}$. The hypothesis then predicts the $\mathbf{y}$ which maximizes $\mathcal{F}$:

$$h(\mathbf{x}; \mathbf{w}) = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} \mathcal{F}(\mathbf{x}, \mathbf{y}; \mathbf{w}). \tag{1}$$

We assume our discriminant to be linear in a joint feature space $\Psi : \mathcal{X} \times \mathcal{Y} \to \Re^m$, which we can write as

$$\mathcal{F}(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}). \tag{2}$$

The feature representation Ψ must enable meaningful discrimination between high quality and low quality predictions. As such, different feature representations may be appropriate for different retrieval settings. We discuss some possible extensions in Section 8.

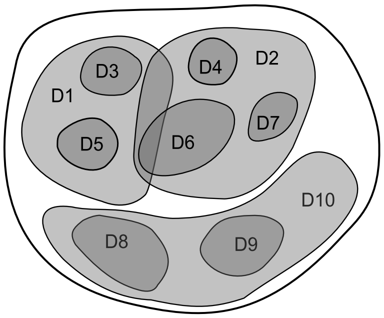

Figure 1. Visualization of Documents Covering Subtopics

4. Maximizing Word Coverage

Figure 1 depicts an abstract visualization of our prediction problem. The sets represent candidate documents $\mathbf{x}$ of a query, and the area covered by each set is the "information" (represented as subtopics $\mathbb{T}$) covered by that document. If $\mathbb{T}$ were known, we could use a greedy method to find a solution with high subtopic diversity. For $K = 3$, the optimal solution in Figure 1 is $\mathbf{y} = \{D1, D2, D10\}$. In general however, the subtopics are unknown. We instead assume that the candidate set contains discriminating features which separates subtopics from each other, and these are primarily based on word frequencies.

As a proxy for explicitly covering subtopics, we formulate our discriminant Ψ based on weighted word coverage. Intuitively, covering more (distinct) words should result in covering more subtopics. The relative importance of covering any word can be modeled using features describing various aspects of word frequencies within documents in $\mathbf{x}$. We make no claims regarding any generative models relating topics to words, but rather simply assume that word frequency features are highly discriminative of subtopics within $\mathbf{x}$.

We now present a simple example of Ψ from (2). Let $V(\mathbf{y})$ denote the union of words contained in the documents of the predicted subset $\mathbf{y}$, and let $\phi(v, \mathbf{x})$ denote the feature vector describing the frequency of word v amongst documents in $\mathbf{x}$. We then write Ψ as

$$\Psi(\mathbf{x}, \mathbf{y}) = \sum_{v \in V(\mathbf{y})} \phi(v, \mathbf{x}). \tag{3}$$

Given a model vector $\mathbf{w}$, the benefit of covering word v in candidate set x is $\mathbf{w}^T \phi(v, \mathbf{x})$. This benefit is realized when a document in $\mathbf{y}$ contains v, i.e., $v \in V(\mathbf{y})$. We use the same model weights for all words. A prediction is made by choosing $\mathbf{y}$ to maximize (2).

This word appears ...
... in a document in $\mathbf{y}$.
... at least 5 times in a document in $\mathbf{y}$.
... with frequency at least 5% in a document in $\mathbf{y}$.
... in the title of a document in $\mathbf{y}$.
... within the top 5 TFIDF of a document in $\mathbf{y}$.

Table 1. Examples of Importance Criteria

The word v has ...
... a $\lvert D_1(v) \rvert / n$ ratio of at least 40%
... a $\lvert D_2(v) \rvert / n$ ratio of at least 50%
... a $\lvert D_\ell(v) \rvert / n$ ratio of at least 25%

Table 2. Examples of Document Frequency Features

This formulation yields two properties which enable optimizing for diversity. First, covering a word twice provides no additional benefit. Second, the feature vector $\phi(v, \mathbf{x})$ is computed using other documents in the candidate set. Thus, diversity is measured locally rather than relative to the whole corpus. Both properties are absent from conventional ranking methods which evaluate each document individually.

In practical applications, a more sophisticated Ψ may be more appropriate. We develop our discriminant by addressing two criteria: how well a document covers a word, and how important it is to cover a word in $\mathbf{x}$.

4.1. How well a document covers a word

In our simple example (3), a single word set $V(\mathbf{y})$ is used, and all words that appear at least once in $\mathbf{y}$ are included. However, documents do not cover all words equally well, which is something not captured in (3). For example, a document which contains 5 instances of the word "lion" might cover the word better than another document which only contains 2 instances.

Instead of using only one $V(\mathbf{y})$, we can use L such word sets $V_1(\mathbf{y}), \ldots, V_L(\mathbf{y})$. Each word set $V_\ell(\mathbf{y})$ contains only words satisfying certain importance criteria. These importance criteria can be based on properties such as appearance in the title, the term frequency in the document, and having a high TFIDF value in the document (Salton & Buckley, 1988). Table 1 contains examples of importance criteria that we considered. For example, if importance criterion ℓ requires appearing at least 5 times in a document, then $V_\ell(\mathbf{y})$ will be the set of words which appear at least 5 times in some document in $\mathbf{y}$. The most basic criterion simply requires appearance in a document, and using only this criterion will result in (3).

We use a separate feature vector $\phi_\ell(v, \mathbf{x})$ for each importance level. We will describe ϕ_ℓ in greater detail in Section 4.2. We define Ψ from (2) to be the vector

Algorithm 1 Greedy subset selection by maximizing weighted word coverage

```
Input: $\mathbf{w}$, $\mathbf{x}$
Initialize solution $\hat{\mathbf{y}} \leftarrow \emptyset$
for $k = 1, \ldots, K$ do
    $\hat{x} \leftarrow \text{argmax}_{x:x \notin \hat{\mathbf{y}}} \mathbf{w}^T \Psi(\mathbf{x}, \hat{\mathbf{y}} \cup \{d\})$
    $\hat{\mathbf{y}} \leftarrow \hat{\mathbf{y}} \cup \{\hat{x}\}$
end for
return $\hat{\mathbf{y}}$
```

composition of all the ϕ_ℓ vectors,

$$\Psi(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \sum_{v \in V_1(\mathbf{y})} \phi_1(v, \mathbf{x}) \\ \vdots \\ \sum_{v \in V_L(\mathbf{y})} \phi_L(v, \mathbf{x}) \\ \sum_{i=1}^{n} y_i \psi(x_i, \mathbf{x}) \end{bmatrix}. \tag{4}$$

We can also include a feature vector $\psi(x, \mathbf{x})$ to encode any salient document properties which are not captured at the word level (e.g., "this document received a high score with an existing ranking function").

4.2. The importance of covering a word

In this section, we describe our formulation for the feature vectors $\phi_1(v, \mathbf{x}), \ldots, \phi_L(v, \mathbf{x})$. These features encode the benefit of covering a word, and are based primarily on document frequency in $\mathbf{x}$.

Using the importance criteria defined in Section 4.1, let $D_\ell(v)$ denote the set of documents in $\mathbf{x}$ which cover word v at importance level ℓ. For example, if the importance criterion is "appears at least 5 times in the document", then $D_\ell(v)$ is the set of documents that have at least 5 copies of v. This is, in a sense, a complementary definition to $V_\ell(\mathbf{y})$.

We use thresholds on the ratio $|D_\ell(v)|/n$ to define feature values of $\phi_\ell(v, \mathbf{x})$ that describe word v at different importance levels. Table 2 describes examples of features that we considered.

4.3. Making Predictions

Putting the formulation together, $\mathbf{w}_\ell^T \phi_\ell(v, \mathbf{x})$ denotes the benefit of covering word v at importance level ℓ, where $\mathbf{w}_\ell$ is the sub-vector of $\mathbf{w}$ which corresponds to ϕ_ℓ in (4). A word is only covered at importance level ℓ if it appears in $V_\ell(\mathbf{y})$. The goal then is to select K documents which maximize the aggregate benefit.

Selecting the K documents which maximizes (2) takes n choose K time, which quickly becomes intractable for even small values of K. Algorithm 1 describes a greedy algorithm which iteratively selects the document with highest marginal gain. Our prediction problem is a special case of the Budgeted Max Coverage problem (Khuller et al., 1997), and the greedy algorithm is known to have a $(1 - 1/e)$-approximation bound. During prediction, the weight vector $\mathbf{w}$ is assumed to be already learned.

5. Training with Structural SVMs

SVMs have been shown to be a robust and effective approach to complex learning problems in information retrieval (Yue et al., 2007; Chapelle et al., 2007). For a given training set $S = \{(\mathbb{T}^{(i)}, \mathbf{x}^{(i)})\}_{i=1}^{N}$, we use the structural SVM formulation, presented in Optimization Problem 1, to learn a weight vector $\mathbf{w}$.

Optimization Problem 1. (Structural SVM)

$$\min_{\mathbf{w}, \xi \geq 0} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{N}\sum_{i=1}^{N} \xi_i \tag{5}$$

s.t. $\forall i, \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}^{(i)}$:

$$\mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \geq \mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{y}) + \Delta(\mathbb{T}^{(i)}, \mathbf{y}) - \xi_i \tag{6}$$

The objective function (5) is a tradeoff between model complexity, $\|\mathbf{w}\|^2$, and a hinge loss relaxation of the training loss for each training example, $\sum \xi_i$, and the tradeoff is controlled by the parameter C. The $\mathbf{y}^{(i)}$ in the constraints (6) is the prediction which minimizes $\Delta(\mathbb{T}^{(i)}, \mathbf{y}^{(i)})$, and can be chosen via greedy selection.

The formulation of Ψ in (4) is very similar to learning a straightforward linear model. The key difference is that each training example is now a set of documents $\mathbf{x}$ as opposed to a single document. For each training example, each "suboptimal" labeling is associated with a constraint (6). There are now an immense number of constraints to define for SVM training.

Despite the large number of constraints, we can use Algorithm 2 to solve OP 1 efficiently. Algorithm 2 is a cutting plane algorithm, iteratively adding constraints until we have solved the original problem within a desired tolerance ϵ (Tsochantaridis et al., 2005). The algorithm starts with no constraints, and iteratively finds for each example $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ the $\hat{\mathbf{y}}$ which encodes the most violated constraint. If the corresponding constraint is violated by more than ϵ we add $\hat{\mathbf{y}}$ into the working set $\mathcal{W}_i$ of active constraints for example i, and re-solve (5) using the updated $\mathcal{W}$. Algorithm 2's outer loop is guaranteed to halt within a polynomial number of iterations for any desired precision ϵ.

Theorem 1. *Let* $\bar{R} = \max_i \max_{\mathbf{y}} \|\Psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \Psi(\mathbf{x}^{(i)}, \mathbf{y})\|$, $\bar{\Delta} = \max_i \max_{\mathbf{y}} \Delta(\mathbb{T}^{(i)}, \mathbf{y})$, *and for any*

Algorithm 2 Cutting plane algorithm for solving OP 1 within tolerance ϵ.

```
Input: $(\mathbf{x}^{(1)}, \mathbb{T}^{(1)}), \ldots, (\mathbf{x}^{(N)}, \mathbb{T}^{(N)})$, $C$, $\epsilon$
$\mathcal{W}_i \leftarrow \emptyset$ for all $i = 1, \ldots, n$
repeat
  for $i = 1, \ldots, n$ do
    $H(\mathbf{y}; \mathbf{w}) \equiv \Delta(\mathbb{T}^{(i)}, \mathbf{y}) + \mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{y}) - \mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{y}_i)$
    compute $\hat{\mathbf{y}} = \text{argmax}_{\mathbf{y} \in \mathcal{Y}} H(\mathbf{y}; \mathbf{w})$
    compute $\xi_i = \max\{0, \max_{\mathbf{y} \in \mathcal{W}_i} H(\mathbf{y}; \mathbf{w})\}$
    if $H(\hat{\mathbf{y}}; \mathbf{w}) > \xi_i + \epsilon$ then
      $\mathcal{W}_i \leftarrow \mathcal{W}_i \cup \{\hat{\mathbf{y}}\}$
      $\mathbf{w} \leftarrow$ optimize (5) over $\mathcal{W} = \bigcup_i \mathcal{W}_i$
    end if
  end for
until no $\mathcal{W}_i$ has changed during iteration
```

$\epsilon > 0$, Algorithm 2 terminates after adding at most

$$\max \left\{ \frac{2n\bar{\Delta}}{\epsilon}, \frac{8C\bar{\Delta}\bar{R}^2}{\epsilon^2} \right\}$$

constraints to the working set $\mathcal{W}$. See (Tsochantaridis et al., 2005) for proof.

However, each iteration of the inner loop of Algorithm 2 must compute $\text{argmax}_{\mathbf{y} \in \mathcal{Y}} H(\mathbf{y}; \mathbf{w})$, or equivalently,

$$\underset{\mathbf{y} \in \mathcal{Y}}{\text{argmax}}\, \Delta(\mathbb{T}^{(i)}, \mathbf{y}) + \mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{y}), \tag{7}$$

since $\mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ is constant with respect to $\mathbf{y}$. Though closely related to prediction, this has an additional complication with the $\Delta(\mathbb{T}^{(i)}, \mathbf{y})$ term. As such, a constraint generation oracle is required.

5.1. Finding Most Violated Constraint

The constraint generation oracle must efficiently solve (7). Unfortunately, solving (7) exactly is intractable since exactly solving the prediction task,

$$\underset{\mathbf{y} \in \mathcal{Y}}{\text{argmax}}\, \mathbf{w}^T \Psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}),$$

is intractable. An approximate method must be used. The greedy inference method in Algorithm 1 can be easily modified for this purpose. Since constraint generation is also a special case of the Budgeted Max Coverage Problem, the $(1-1/e)$-approximation bound still holds. Despite using an approximate constraint generation oracle, SVM training is still known to terminate in a polynomial number of iterations (Finley & Joachims, 2008). Furthermore in practice, training typically converges much faster than the worst case considered by the theoretical bounds. Intuitively, a small set of the constraints can approximate to ϵ precision the feasible space defined by the intractably many constraints. When constraint generation is approximate however, the ϵ precision guarantee no longer holds. Nonetheless, using approximate constraint generation can still offer good performance, which we will evaluate empirically.

6. Experiment Setup

We tested the effectiveness of our method using the TREC 6-8 Interactive Track Queries[2]. Relevant documents are labeled using subtopics. For example, query 392 asked human judges to identify different applications of robotics in the world today, and they identified 36 subtopics among the results such as nanorobots and using robots for space missions.

The 17 queries we used are 307, 322, 326, 347, 352, 353, 357, 362, 366, 387, 392, 408, 414, 428, 431, 438, and 446. Three of the original 20 queries were discarded due to having small candidate sets, making them uninteresting for our experiments. Following the setup in (Zhai et al., 2003), candidate sets only include documents which are relevant to at least one subtopic. This decouples the diversity problem, which is the focus of our study, from the relevance problem. In practice, approaches like ours might be used to post-process the results of a commercial search engine. We also performed Porter stemming and stop-word removal.

We used a 12/4/1 split for our training, validation and test sets, respectively. We trained our SVM using C values varying from 1e-5 to 1e3. The best C value is then chosen on the validation set, and evaluated on the test query. We permuted our train/validation/test splits until all 17 queries were chosen once for the test set. Candidate sets contain on average 45 documents, 20 subtopics, and 300 words per document. We set the retrieval size to $K = 5$ since some candidate sets contained as few as 16 documents.

We compared our method against Okapi (Robertson et al., 1994), and Essential Pages (Swaminathan et al., 2008). Okapi is a conventional retrieval function which evaluates the relevance of each document individually and does not optimize for diversity. Like our method, Essential Pages also optimizes for diversity by selecting documents to maximize weighted word coverage (but based on a fixed, rather than a learned, model). In their model, the benefit of document x_i covering a word v is defined to be

$$TF(v, x_i) \log \left(\frac{1}{DF(v, \mathbf{x})} \right),$$

[2] `http://trec.nist.gov/`

Method	Loss
Random	0.469
Okapi	0.472
Unweighted Model	0.471
Essential Pages	0.434
SVM$^{\Delta}_{div}$	0.349
SVM$^{\Delta}_{div2}$	0.382

Table 3. Performance on TREC ($K = 5$)

where $TF(v, x_i)$ is the term frequency of v in x_i and $DF(v, \mathbf{x})$ is the document frequency of v in $\mathbf{x}$.

We define our loss function to be the weighted percentage of subtopics not covered. For a given candidate set, each subtopic's weight is proportional to the number of documents that cover that subtopic. This is attractive since it assigns a high penalty to not covering a popular subtopic. It is also compatible with our discriminant since frequencies of important words will vary based on the distribution of subtopics.

The small quantity of TREC queries makes some evaluations difficult, so we also generated a larger synthetic dataset of 100 candidate sets. Each candidate set has 100 documents covering up to 25 subtopics. Each document samples 300 words independently from a multinomial distribution over 5000 words. Each document's word distribution is a mixture of its subtopics' distributions. We used this dataset to evaluate how performance changes with retrieval size K. We used a 15/10/75 split for training, validation, and test sets.

7. Experiment Results

Let SVM$^{\Delta}_{div}$ denote our method which uses term frequencies and title words to define importance criteria (how well a document covers a word), and let SVM$^{\Delta}_{div2}$ denote our method which in addition also uses TFIDF. SVM$^{\Delta}_{div}$ and SVM$^{\Delta}_{div2}$ use roughly 200 and 300 features, respectively. Table 1 contains examples of importance criteria that could be used.

Table 3 shows the performance results on TREC queries. We also included the performance of randomly selecting 5 documents as well as an unweighted word coverage model (all words give equal benefit when covered). Only Essential Pages, SVM$^{\Delta}_{div}$ and SVM$^{\Delta}_{div2}$ performed better than random.

Method Comparison	Win / Tie / Lose
SVM$^{\Delta}_{div}$vs Essential Pages	14 / 0 / 3 **
SVM$^{\Delta}_{div2}$vs Essential Pages	13 / 0 / 4
SVM$^{\Delta}_{div}$vs SVM$^{\Delta}_{div2}$	9 / 6 / 2

Table 4. Per Query Comparison on TREC ($K = 5$)

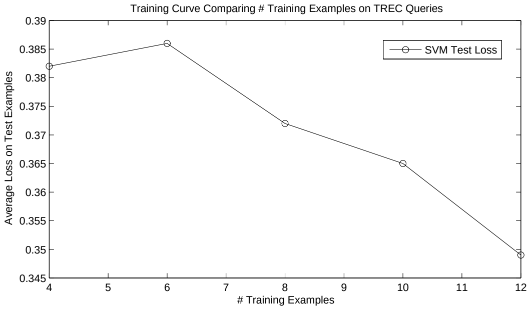

Figure 2. Comparing Training Size on TREC ($K = 5$)

Table 4 shows the per query comparisons between SVM$^{\Delta}_{div}$, SVM$^{\Delta}_{div2}$ and Essential Pages. Two stars indicate 95% significance using the Wilcoxon signed rank test. While the comparison is not completely fair since Essential Pages was designed for a slightly different setting, it demonstrates the benefit of automatically fitting a retrieval function to the specific task at hand.

Despite having a richer feature space, SVM$^{\Delta}_{div2}$ performs worse than SVM$^{\Delta}_{div}$. We conjecture that the top TFIDF words do not discriminate between subtopics. These words are usually very descriptive of the query as a whole, and thus will appear in all subtopics.

Figure 2 shows the average test performance of SVM$^{\Delta}_{div}$ as the number of training examples is varied. We see a substantial improvement in performance as training set size increases. It appears that more training data would further improve performance.

7.1. Approximate Constraint Generation

Using appoximate constraint generation might compromise our model's ability to (over-)fit the data. We addressed this concern by examining the training loss as the C parameter is varied. The training curve of SVM$^{\Delta}_{div}$ is shown in Figure 3. Greedy optimal refers to the loss incurred by a greedy method with knowledge of subtopics. As we increase C (favoring low training loss over low model complexity), our model is able to fit the training data almost perfectly. This indicates that approximate constraint generation is acceptable for our training purposes.

7.2. Varying Predicted Subset Size

We used the synthetic dataset to evaluate the behavior of our method as we vary the retrieval size K. It is difficult to perform this evaluation on the TREC queries – since some candidate sets have very few documents

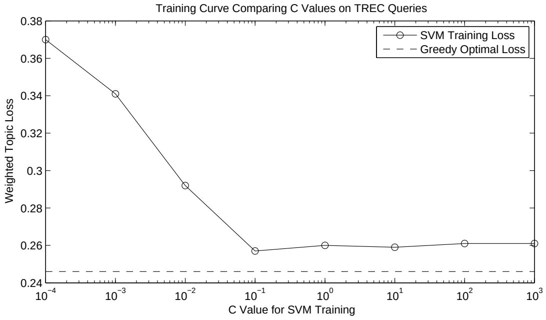

Figure 3. Comparing C Values on TREC ($K = 5$)

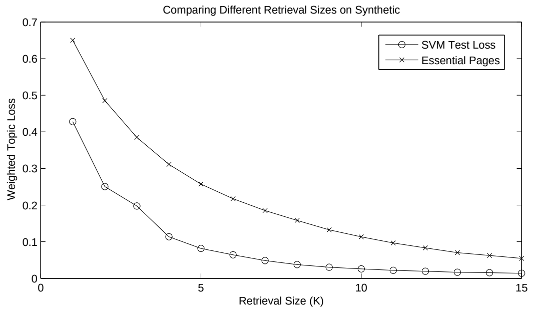

Figure 4. Varying Retrieval Size on Synthetic

or subtopics, using higher K would force us to discard more queries. Figure 4 shows that the test performance of SVM$^{\Delta}_{div}$ consistently outperforms Essential Pages at all levels of K.

7.3. Running Time

Predicting takes linear time. During training, Algorithm 2 loops for 10 to 100 iterations. For ease of development, we used a Python interface[3] to SVM^{struct}. Even with our unoptimized code, most models trained within an hour, with the slowest finishing in only a few hours. We expect our method to easily accomodate much more data since training scales linearly with dataset size (Joachims et al., to appear).

8. Extensions

8.1. Alternative Discriminants

Maximizing word coverage might not be suitable for other types of retrieval tasks. Our method is a general framework which can incorporate other discriminant formulations. One possible alternative is to maximize the pairwise distance of items in the predicted subset. Learning a weight vector for (2) would then amount to finding a distance function for a specific retrieval task. Any discriminant can be used so long as it captures the salient properties of the retrieval task, is linear in a joint feature space (2), and has effective inference and constraint generation methods.

8.2. Alternative Loss Functions

Our method is not restricted to using subtopics to measure diversity. Only our loss function $\Delta(\mathbb{T}, \mathbf{y})$ makes use of subtopics during SVM training. We can also incorporate loss functions which can penalize other types of diversity criteria and also use other forms of training data, such as clickthrough logs. The only requirement is that it must be computationally compatible with the constraint generation oracle (7).

8.3. Additional Word Features

Our choice of features is based almost exclusively on word frequencies. The sole exception is using title words as an importance criterion. The goal of these features is to describe how well a document covers a word and the importance of covering a word in a candidate set. Other types of word features might prove useful, such as anchor text, URL, and any meta information contained in the documents.

9. Conclusion

In this paper we have presented a general machine learning approach to predicting diverse subsets. Our method compares favorably to methods which do not perform learning, demonstrating the usefulness of training feature rich models for specific retrieval tasks. To the best of our knowledge, our method is the first approach which directly trains for subtopic diversity. Our method is also efficient since it makes predictions in linear time and has training time that scales linearly in the number of queries.

In this paper we separated the diversity problem from the relevance problem. An interesting direction for future work would be to jointly model both relevance and diversity. This is a more challenging problem since it requires balancing a tradeoff for presenting both novel and relevant information.

The non-synthetic TREC dataset is also admittedly small. Generating larger (and publicly available) labeled datasets which encode diversity information is another important direction for future work.

Acknowledgements

The work was funded under NSF Award IIS-0713483, NSF CAREER Award 0237381, and a gift from Ya-

[3] `http://www.cs.cornell.edu/~tomf/svmpython2/`

hoo! Research. The first author is also partly funded by a Microsoft Research Fellowship and a Yahoo! Key Technical Challenge Grant. The authors also thank Darko Kirovski for initial discussions regarding his work on Essential Pages.

References

Broder, A., Fontoura, M., Gabrilovich, E., Joshi, A., Josifovski, V., & Zhang, T. (2007). Robust classification of rare queries using web knowledge. *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR).*

Burges, C. J. C., Ragno, R., & Le, Q. (2006). Learning to rank with non-smooth cost functions. *Proceedings of the International Conference on Advances in Neural Information Processing Systems (NIPS).*

Cai, L., & Hofmann, T. (2004). Hierarchical document categorization with support vector machines. *In Proceedings of the ACM Conference on Information and Knowledge Management (CIKM).*

Carbonell, J., & Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and reproducing summaries. *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR).*

Chapelle, O., Le, Q., & Smola, A. (2007). Large margin optimization of ranking measures. *NIPS workshop on Machine Learning for Web Search.*

Chen, H., & Karger, D. (2006). Less is more: Probabilistic models for retrieving fewer relevant documents. *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR).*

desJardins, M., Eaton, E., & Wagstaff, K. (2006). Learning user preferences for sets of objects. *Proceedings of the International Conference on Machine Learning (ICML)* (pp. 273–280). ACM.

Finley, T., & Joachims, T. (2008). Training structural svms when exact inference is intractable. *Proceedings of the International Conference on Machine Learning (ICML).*

Joachims, T., Finley, T., & Yu, C. (to appear). Cutting-plane training of structural svms. *Machine Learning.*

Khuller, S., Moss, A., & Naor, J. (1997). The budgeted maximum coverage problem. *Information Processing Letters*, *70(1)*, 39–45.

Kleinberg, R., Radlinski, F., & Joachims, T. (2008). Learning diverse rankings with multi-armed bandits. *Proceedings of the International Conference on Machine Learning (ICML).*

Li, P., Burges, C., & Wu, Q. (2007). Learning to rank using classification and gradient boosting. *Proceedings of the International Conference on Advances in Neural Information Processing Systems (NIPS).*

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., & Gatford, M. (1994). Okapi at TREC-3. *Proceedings of TREC-3.*

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, *24(5)*, 513–523.

Swaminathan, A., Mathew, C., & Kirovski, D. (2008). *Essential pages* (Technical Report MSR-TR-2008-015). Microsoft Research.

Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, *6(Sep)*, 1453–1484.

Wagstaff, K., desJardins, M., Eaton, E., & Montminy, J. (2007). Learning and visualizing user preferences over sets. *American Association for Artificial Intelligence (AAAI).*

Yue, Y., Finley, T., Radlinski, F., & Joachims, T. (2007). A support vector method for optimizing average precision. *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR).*

Zhai, C., Cohen, W. W., & Lafferty, J. (2003). Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR).*

Zhang, B., Li, H., Liu, Y., Ji, L., Xi, W., Fan, W., Chen, Z., & Ma, W. (2005). Improving web search results using affinity graph. *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR).*

Zheng, Z., Zha, H., Zhang., T., Chapelle, O., Chen, K., & Sun, G. (2007). A general boosting method and its application to learning ranking functions for web search. *Proceedings of the International Conference on Advances in Neural Information Processing Systems (NIPS).*