Selectively Diversifying Web Search Results

Rodrygo L. T. Santos
rodrygo@dcs.gla.ac.uk

Craig Macdonald
craigm@dcs.gla.ac.uk

Iadh Ounis
ounis@dcs.gla.ac.uk

School of Computing Science
University of Glasgow
G12 8QQ Glasgow, UK

ABSTRACT

Search result diversification is a natural approach for tackling ambiguous queries. Nevertheless, not all queries are equally ambiguous, and hence different queries could benefit from different diversification strategies. A more lenient or more aggressive diversification strategy is typically encoded by existing approaches as a trade-off between promoting relevance or diversity in the search results. In this paper, we propose to learn such a trade-off on a per-query basis. In particular, we examine how the need for diversification can be learnt for each query—given a diversification approach and an unseen query, we predict an effective trade-off between relevance and diversity based on similar previously seen queries. Thorough experiments using the TREC ClueWeb09 collection show that our selective approach can significantly outperform a uniform diversification for both classical and state-of-the-art diversification approaches.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models*

General Terms

Algorithms, Experimentation, Performance

Keywords

Web search, relevance, diversity, selective retrieval, machine learning, feature selection

1. INTRODUCTION

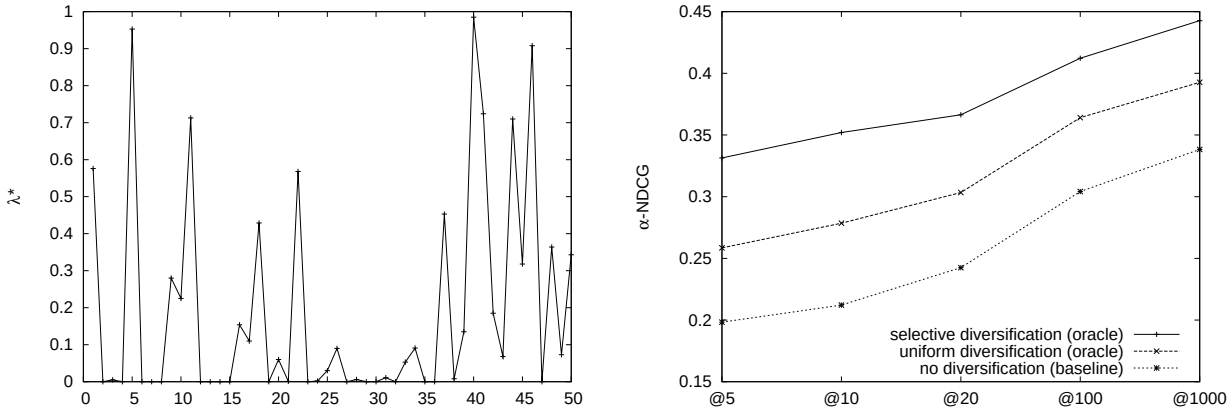
Queries submitted to a Web search engine are often ambiguous [33]. For instance, a user issuing the query '*bond*' could mean the financial instrument for debt security, the classical crossover string quartet 'Bond', or Ian Fleming's secret agent character 'James Bond'. In the absence of any knowledge of the user's context or preferences, a sensible approach for a search engine is to diversify the results retrieved for this query. By doing so, the chance that any user issuing this query will find *at least one relevant result* to their particular information need is maximised [10].

Intuitively, maximising the satisfaction of the population of users issuing the same, ambiguous query involves trading off relevance for diversity in the search results. On the one hand, a standard relevance-oriented ranking could focus on the most likely interpretation of the query (e.g., the most popular). On the other hand, a diversity-oriented ranking could also cater for other plausible query interpretations. These two strategies could be then integrated as a bi-criteria ranking objective for an improved performance [16].

Most of the existing diversification approaches build upon this idea, with an interpolation parameter λ controlling the trade-off between relevance and diversity [5, 27, 34, 38]. Typically, this trade-off is *uniformly* optimised so as to maximise the average diversification performance on a set of training queries. However, different queries might benefit from different diversification strategies, as not all queries are equally ambiguous. For instance, while the query '*bond*' might benefit from a more aggressive diversification strategy, a more lenient strategy might suffice for a less ambiguous query such as '*james bond*'. In the extreme case, a clear query like '*quantum of solace website*' might attain an optimal performance even without any diversification. To quantify this observation, Figure 1(a) shows the optimal trade-off λ^* for one of the diversification approaches investigated in this work for each of the TREC 2009 Web track topics [9]. From the figure, it is clear that different queries benefit from different trade-offs, and that any uniform choice of λ for all queries would be suboptimal. Indeed, Figure 1(b) shows that *selectively* optimising this trade-off on a per-query basis substantially outperforms a uniform optimisation regime. That said, the key challenge becomes how to automatically estimate such a trade-off for an unseen query.

In this paper, we hypothesise that existing diversification approaches can be improved by selecting an appropriate diversification strategy on a per-query basis. In particular, we propose to selectively diversify the results retrieved for a given query, by predicting an effective trade-off between relevance and diversity for this query. As a result, we estimate not only whether a particular query would benefit from diversifying its results, but also by how much. To enable our approach, we leverage a large pool of features from different branches of query analysis in the literature and cast this problem as a nearest neighbour regression task. Given a diversification approach and an unseen query, we estimate an effective diversification trade-off based on the optimal trade-

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.



(a) Optimal trade-off per query. (b) Optimal diversification performance.

Figure 1: Optimal trade-off (a) and diversification performance (b) for the TREC 2009 Web track topics. The baseline, non-diversified ranking is provided by the Divergence From Randomness DPH model [3]. Diversified (both uniform and selective) re-rankings are produced using the xQuAD framework [27].

off observed for similar, previously seen queries using the same diversification approach. To the best of our knowledge, our approach constitutes the first attempt to tackle search result diversification as a query-dependent ranking problem. We thoroughly evaluate our approach in the context of the diversity task of the TREC 2009 Web track [9], using both a classical [5] and a state-of-the-art [27] diversification approaches as baselines. Our results show that both baselines can be markedly improved by learning their diversification trade-off on a per-query basis, even when compared to an upper-bound uniform setting for this trade-off. Moreover, we analyse the influence of the deployed features on the accuracy of the predicted trade-offs and the robustness of our approach to perturbations in the trade-off prediction.

The remainder of this paper is organised as follows. Section 2 introduces the related work upon which we build. Section 3 further details our main contributions. Section 4 introduces our selective search result diversification approach. Section 5 describes the features deployed to learn the diversification trade-off. Sections 6 and 7 detail the experimental setup and the evaluation of our proposed approach, respectively. Finally, Section 8 presents our conclusions.

2. BACKGROUND AND RELATED WORK

In this section, we introduce the search result diversification problem and the most successful approaches for this problem in the literature. In common, these approaches are typically applied with a uniform setting across all queries. As a step towards introducing our selective diversification approach, we review related works on selective information retrieval, particularly on query-dependent ranking.

2.1 Search Result Diversification

Search result diversification can be stated as an optimisation problem, in which the goal is to find a ranking of documents that together provide a complete coverage of the aspects underlying a query, as early as possible. In its general form, this is an NP-hard problem [1]. In practice, most previous works on search result diversification are based on a greedy approximation to this problem. Given a ranking R for an ambiguous query, a re-ranking S is produced by iteratively selecting a 'local-best' document from $R \setminus S$. The choice of this document is typically governed by an objective function, which linearly combines *relevance* and *diversity* estimates, in order to obtain the final ranking score [16].

Relevance estimates are normally produced by standard retrieval approaches, such as BM25 [25] or the DPH Divergence From Randomness (DFR) model [3]. The key difference among existing diversification approaches lies on how they estimate the diversity of a document. In particular, the diversification approaches in the literature can be categorised as either *implicit* or *explicit* [29], depending on how they account for the different aspects underlying a query.

Implicit search result diversification approaches assume that similar documents will cover similar aspects of a query, and should hence be demoted in the final ranking, so as to reduce its overall redundancy. Among the approaches in this family reported in the literature, the maximal marginal relevance (MMR) method of Carbonell and Goldstein [5] is a typical example. The general idea of the MMR method is to trade off document similarity with respect to the query and document dissimilarity with respect to the already selected documents. Other implementations of this idea include the approach of Zhai et al. [38] to model relevance and redundancy within a risk minimisation framework. In particular, they promote documents with highly divergent language models from those of the already selected documents. Chen and Karger [8] proposed a probabilistic approach to the related problem of finding at least one relevant result for a given query. Their approach chooses documents under the assumption that those already chosen are not relevant to the query. More recently, Wang and Zhu [34] proposed to diversify a document ranking as a means to reduce the risk of overestimating relevance. In their approach, documents with low relevance score correlations with respect to documents selected in previous iterations are promoted.

In common, the aforementioned approaches only consider the aspects underlying a query in an implicit manner. An alternative approach consists of explicitly modelling these aspects. For instance, Agrawal et al. [1] employed a classification taxonomy over queries and documents to represent

query aspects. Their approach iteratively promotes documents that share a high number of classes with the query, while demoting those with classes already well represented in the ranking. In a similar vein, Carterette and Chandar [7] proposed a probabilistic approach to maximise the coverage of the retrieved documents with respect to the aspects of a query. In particular, they model these aspects as topics identified from the top ranked documents. Recently, Santos et al. [27] introduced the xQuAD probabilistic framework for search result diversification, which explicitly represents query aspects as 'sub-queries'. They define diversity based on the estimated relevance of documents to multiple sub-queries and the relative importance of each sub-query.

In Section 7, both MMR [5] and xQuAD [27] are used as baselines for our selective approach, as representatives of implicit and explicit diversification approaches, respectively. Moreover, xQuAD was the best approach in the diversity task of the TREC 2009 Web track (Cat. B), and hence represents the state-of-the-art in search result diversification [9].

2.2 Selective Information Retrieval

Selective approaches are relatively common in other information retrieval tasks. A typical example is the identification of queries more likely to benefit from query expansion [37]. Other selective approaches deal with applying different retrieval techniques for different queries. These approaches can be generally categorised based on whether they rely on a hard classification of queries into predefined types.

Query type detection approaches aim to classify a query into one of a set of target types (e.g., informational, navigational, or transactional [4]), and then apply a retrieval model specifically trained for the predicted type. For instance, Kang and Kim [19] showed that different query types can benefit from the application of different retrieval approaches. However, a major drawback of these approaches is the limited accuracy of existing type detection mechanisms. Moreover, queries of different types can often benefit from the application of the same retrieval approach [12].

A different approach for selective information retrieval relies on a softer classification. In particular, instead of classifying a query into a predefined target type, an alternative is to identify similar queries from a training set, and then to apply a retrieval model suitable for this set. This was the approach taken by Geng et al. [15] to improve Web search effectiveness. In their approach, a k-nearest neighbour classifier [2] was used to identify training queries similar to an unseen query. A retrieval model was then learnt based on the identified queries and applied to the unseen query. A more general approach was proposed by Peng et al. [24]. In their work, a ranking function was chosen from a large pool of candidate functions, based on their performance on neighbouring training queries to an unseen query.

Inspired by these works, in this paper, we seek to learn the diversification trade-off for an unseen query based on the optimal trade-offs of similar training queries. However, differently from these works, which relied on a single feature to identify neighbouring queries, we leverage a large pool of query features, inspired by different query analysis tasks. To our knowledge, our work constitutes the first attempt to tackle search result diversification as a query-dependent ranking problem. By doing so, we recognise that different queries may have a different level of ambiguity, and hence demand different diversification strategies.

3. CONTRIBUTIONS OF THIS PAPER

The major contributions of this paper are:

- A novel selective approach for search result diversification, which effectively predicts the trade-off between relevance and diversity on a per-query basis;
- A thorough evaluation of the proposed approach in terms of diversification effectiveness and sensitivity to prediction perturbations;
- An examination of the usefulness of several features, inspired by different query analysis tasks, at predicting an effective trade-off between relevance and diversity.

4. SELECTIVE DIVERSIFICATION

As discussed in the previous sections, not all user queries are equally ambiguous, which suggests that a one-size-fits-all diversification strategy might be suboptimal. In order to learn a trade-off between relevance and diversity for queries with different levels of ambiguity, we propose a supervised selective diversification approach. Given a diversification approach δ and an unseen query q, our goal is to learn an effective setting for the diversification trade-off λ, which maximises the performance of δ according to an evaluation metric ϵ.[1] In particular, we predict this trade-off based on the optimal trade-off observed for a set of training queries, optimised for the same diversification approach δ, in order to maximise the metric ϵ. Our approach is general and can be applied to improve most existing diversification approaches (e.g., [5, 27, 34, 38]), regardless of their particular implementation choices. In fact, we consider a diversification approach δ as a baseline system, which ranks documents according to an abstract model:

$$\text{score}_\delta(q, D) = (1 - \lambda)\,\text{rel}_\delta(q, D) + \lambda\,\text{div}_\delta(q, D) \qquad (1)$$

where a collection of documents D is scored with respect to a query q based on a linear combination of relevance ($\text{rel}_\delta(q, D)$) and diversity ($\text{div}_\delta(q, D)$) estimates, with the interpolation parameter λ trading off between the two.

In Section 5, we describe the features employed to enable our selective diversification approach. In the remainder of this section, we detail the three major steps in this approach: (1) labelling training queries with their optimal trade-off, (2) learning a regression model based on the labelled queries and a set of query features, and (3) applying the learnt model to predict an effective trade-off for unseen queries.

4.1 Labelling Training Queries

The first step in our proposed selective diversification approach is to label training data. More precisely, our goal is to build a set of training queries $Q = \{q_i\}$ with corresponding labels $\Lambda_{\delta,\epsilon} = \{\lambda^*_{\delta,\epsilon,i}\}$. A label $\lambda^*_{\delta,\epsilon,i}$ for a training query q_i corresponds to the optimal trade-off between relevance and diversity obtained for this query using the diversification approach δ, according to the evaluation metric ϵ.

In principle, to obtain such an optimal trade-off, we could use any optimisation method (e.g., simulated annealing [20]). In this work, we perform a full scan over the range of possible λ values (i.e., $0 \leq \lambda \leq 1$), with steps of 0.001, and

[1]For instance, δ could be one of the approaches described in Section 2.1, and ϵ could be any metric for diversity evaluation, such as those described in Section 6.2.

select the best value (according to the evaluation metric ϵ) as the label $\lambda^*_{\delta,\epsilon,i}$ for the query q_i. Note that this process is performed offline, with no knowledge of unseen queries.

4.2 Learning a Regression Model

In order to learn a regression model to predict the diversification trade-off λ for an unseen query, we could use different numeric prediction approaches, such as linear regression or model trees [36]. In this work, we employ a k-nearest neighbour (k-NN) [2] algorithm. As an instance-based learning approach, k-NN does not have an explicit training phase. Instead, it stores the training data in memory and performs an online regression for each unseen query.

The main advantage of such a lazy learning approach is that a different and potentially more targeted learning function is estimated based on the training neighbourhood of an unseen query, rather than on the entire training data. This reduces the complexity of the learning process by exploiting the locality of the data [15, 36]. Additionally, k-NN does not make strong assumptions about the underlying data distribution, as other regression approaches do [2].

4.3 Predicting the Diversification Trade-Off

During the online query processing, our goal is to predict an effective trade-off λ for an unseen query q based on a learnt regression model. Since k-NN is a lazy approach, no explicit model is actually learnt a priori, and hence most of the work is conducted online. In particular, the predicted trade-off λ for the test query q is set as the mean of the λ^*_i values of the k nearest neighbouring training queries to q:

$$\lambda = \frac{1}{k} \sum_{i|q_i \in N_k(q)} \lambda^*_i \quad (2)$$

where $N_k(q)$ is a set comprising the k closest training queries to q in the space of the considered features, according to a distance function, typically the Euclidian distance.

Despite its simplicity and effectiveness, two main concerns arise when employing an instance-based learning approach such as k-NN. Firstly, the cost of prediction can be significant, particularly when a large number of training instances is available. In our experiments, we deploy a linear nearest neighbour search algorithm, which requires $O(dn)$ time, considering a set of n training queries in a d-dimensional space [36]. In a real deployment of our approach, the searching time can be dramatically reduced with the use of algorithms based on more sophisticated indexing structures to store the training queries, such as ball trees [22]. The second concern related to instance-based learning is the dimensionality of the feature space. In particular, k-NN considers all instance features when searching for the nearest neighbours. When a true neighbour shares only a few common features with an unseen query, they may be considered far from each other in light of the entire feature space, potentially compromising the accuracy of the prediction [36]. To tackle this issue, we perform a feature selection ahead of the prediction step, as described in Section 6.4.

5. QUERY FEATURES

A pool of meaningful features is crucial for the effectiveness of any learning process. As the goal of our particular task is to learn a trade-off between relevance and diversity for a given query, a natural first direction is to look for features that capture the ambiguity of this query. Intuitively, we would expect unambiguous queries to benefit more from a relevance-oriented ranking strategy, while ambiguous queries should benefit more from a diversity-oriented strategy. However, the diversification trade-off depends not only on the ambiguity of a query, but also on how a diversification approach δ tackles such ambiguity, through its particular estimations of relevance ($\text{rel}_\delta(q, D)$) and diversity ($\text{div}_\delta(q, D)$). For this reason, we also consider non-ambiguity-related features for predicting the diversification trade-off.

In this work, we leverage a total of 953 query features, organised in 33 different classes, in order to predict the diversification trade-off for an unseen query. In particular, these features are extracted from different sources, including the query itself, the top documents retrieved for this query in a target collection, and a query log. Moreover, these features are inspired by five different query analysis tasks, including query concept identification (QCI), query type detection (QTD), query performance prediction (QPP), query log mining (QLM), and query topic classification (QTC). Table 1 summarises all features used in this work. In the remainder of this section, we describe each of them.

5.1 Query Concept Identification (QCI)

A first sign of ambiguity is present at the word level [26]. For instance, a query might contain multiple concepts or named entities, possibly representing a complex information need with multiple intents. Alternatively, a single query concept can have multiple meanings according to a particular source, such as a dictionary or an encyclopedia. To capture these intuitions, we propose two query features based on the identification of query concepts. The first of these counts the number of named entities in the query. In particular, we employ an efficient named-entity recognition approach [28], backed up by a dictionary of entity names built from DBPedia 3.3,[2] with additional person names from the 1990 US Census.[3] In total, we identify entities of four types: people, organisations, products, and locations.

Our second query feature focuses on occurrences of particularly ambiguous query terms, namely, acronyms. As short abbreviations, acronyms typically have multiple interpretations. For instance, 'ACM' has 174 different definitions according to `all-acronyms.com`, including 'Association for Computing Machinery', 'Air Cycle Machine', and 'Air Chief-Marshall'. Instead of deploying sophisticated natural language processing techniques for acronym identification, we simply compute the number of interpretations returned by `all-acronyms.com` for single-term queries. In particular, we assume that acronyms occurring in multi-term queries are disambiguated by the additional terms.

Besides features computed directly from an unseen query, we consider ambiguity-related features based on Wikipedia[4] disambiguation pages. In particular, a Wikipedia disambiguation page represents an ambiguous concept and its associated senses or interpretations [26]. Based on a ranking of Wikipedia articles for the query, we compute two features: the total number of disambiguation pages retrieved, and the number of disambiguating senses associated with each retrieved disambiguation page.

[2] `http://dbpedia.org`
[3] `http://www.census.gov`
[4] `http://en.wikipedia.org`

	source	task	class	description	total
1	query	QCI	AcronymSenses	Number of acronym senses	1
2	query	QPP	AvICTF [18]	Pre-retrieval performance predictor	1
3	query	QPP	AvIDF [13]	Pre-retrieval performance predictor	1
4	query	QPP	AvPMI [6]	Pre-retrieval performance predictor	1
5	query	QPP	EnIDF [6]	Pre-retrieval performance predictor	1
6	query	QCI	EntityCount	Number of named entities in the query	4
7	query	QPP	Gamma1 [18]	Pre-retrieval performance predictor	1
8	query	QPP	Gamma2 [18]	Pre-retrieval performance predictor	1
9	query	QPP	TermCount	Number of unique terms	1
10	query	QPP	TokenCount	Number of tokens	1
11	documents	QPP	ClarityScore [13]	Post-retrieval performance predictor	50
12	documents	QTD	DomainDistribution	Number of documents per domain	15
13	documents	QTC	DocEntityCount	Number of retrieved entities	135
14	documents	QTC	DocEntityEntropy	Entity entropy of centroid document	150
15	documents	QTC	DocEntityPairwiseCosine	Entity distance over pairs of top documents	330
16	documents	QTD	HomePage	Whether results include a homepage	2
17	documents	QTD	HostDistribution	Number of documents per host	15
18	documents	QTD	MaximumScoreIncrement	Maximum difference between any two scores	2
19	documents	QPP	QueryDifficulty [6]	Post-retrieval performance predictor	20
20	documents	QPP	QueryFeedback [6]	Post-retrieval performance predictor	20
21	documents	QTD	URLTypeDistribution	Number of URL components per document	20
22	documents	QCI	WPDisambCount	Number of disambiguation pages retrieved	20
23	documents	QCI	WPDisambSenses [26]	Number of disambiguation senses per document	56
24	documents	QTC	WPCategoryCount	Number of retrieved categories	18
25	documents	QTC	WPCategoryEntropy	Category entropy of centroid document	18
26	documents	QTC	WPPairwiseCosine	Categorical distance over pairs of top documents	54
27	query log	QLM	ClickCount	Number of clicks	3
28	query log	QLM	ClickEntropy [11]	Click entropy at the URL level	1
29	query log	QLM	HostEntropy [35]	Click entropy at the host level	1
30	query log	QLM	QueryFrequency	Number of occurrences	1
31	query log	QLM	ReformulationCount [11]	Number of reformulations in a session	3
32	query log	QLM	ResultCount	Number of displayed results in a session	3
33	query log	QLM	SessionDuration	Session duration in seconds	3
TOTAL					953

Table 1: All query features used in this work.

5.2 Query Type Detection (QTD)

Navigational queries are usually less ambiguous than informational ones, which suggests that useful query type detection features might also be useful for predicting query ambiguity [19]. With this in mind, we leverage several query type detection features proposed in the literature in our learning task. These include the distribution of host names, domain names, and other URL fragments among the top retrieved results for a query, as well as the presence of a homepage among these results. Additionally, we consider the maximum difference in relevance scores between any two retrieved results as a strong indicator of the query type [32]. This feature captures the intuition that relevance scores tend to drop quickly for navigational queries, as typically only a few (often one) documents are relevant for such queries.

5.3 Query Performance Prediction (QPP)

As previously discussed in this section, our approach considers a baseline diversification approach δ as a black-box system with relevance and diversity estimates. Since an optimal diversification trade-off clearly depends on the performance of these estimates, a promising direction is to leverage query performance prediction features [6]. For instance, query ambiguity often correlates negatively with query performance [13]. In this work, we employ a range of both pre-retrieval and post-retrieval predictors. Pre-retrieval predictors depend solely on the query, and estimate its performance based on statistics derived from the target collection, such as the document frequency of individual query terms or the pointwise mutual information of pairs of query terms. Post-retrieval predictors, in turn, are based on the top retrieved results for a given query [17]. For instance, they can estimate the query performance based on how cohesive these results are, according to their language models [13] or relevance models built from them [39].

5.4 Query Log Mining (QLM)

Another promising direction for inferring the ambiguity of a query is to observe the past usage of this query in a query log [30]. Inspired by previous research on query log mining for ambiguity detection, we deploy a number of additional features. For instance, queries often clicked for a single document are intuitively less ambiguous than queries with clicks spread over different documents. We capture this intuition by computing the entropy of user clicks [11, 35], at both the result URL and host levels. Additionally, for each query session, we compute the total number of results displayed to the user, the total duration of the session in seconds, and the total number of query reformulations performed during the session [11]. Here, the intuition is that ambiguous queries will demand longer user interactions comparatively to clear, unambiguous queries. Finally, we also consider other basic features such as the raw frequency of the query in a query log and the total number of clicks it received.

Query log features are computed from the 15-million query MSN Search Spring 2006 Query Log, released in the context of the 2009 Workshop on Web Search Click Data.[5]

[5] `http://research.microsoft.com/en-us/um/people/nickcr/wscd09/`

5.5 Query Topic Classification (QTC)

To further refine the prediction of the diversification trade-off for an unseen query, we consider more specialised features, which capture the distribution of topics among the documents retrieved for this query. These include the raw number of topics represented in the top retrieved results for the query, the pairwise 'topic' distance between any two retrieved documents for the query, and the 'topic' entropy of the centroid of all retrieved documents [31].

In this work, we propose two alternatives to represent documents as vectors over topics. The first of these is inspired by traditional text classification. However, to avoid having to classify a large number of search results into a set of predefined categories, we leverage the category hierarchy of Wikipedia. In particular, given a ranking of Wikipedia articles retrieved for the query, we represent each article as a vector over 12 top-level categories. These are Wikipedia equivalents to the top-level categories in the Open Directory Project (ODP),[6] except for the 'Adult' and 'Shopping' categories, not present in Wikipedia. In order to estimate the classification of each retrieved Wikipedia article into one of these top-level categories, we compute the shortest-path distance between any of the original categories associated with the article and each of the target top-level categories in the Wikipedia category hierarchy. Using these top-level categories rather than the original (more specific) ones reduces the dimensionality of the resulting vectors, and their sparsity when estimating vector distances.

Besides using Wikipedia categories, our second alternative investigates the usefulness of named entities as topic representations. In particular, our intuition is that documents sharing the same entities tend to be more similar, in which case the query for which they are retrieved tends to be less ambiguous. As in Section 5.1, to identify entity occurrences in the top retrieved documents, we rely on a dictionary-based approach [28], based on named entities from DBPedia. By representing each retrieved document as a vector over the entities it contains, we derive analogous features to those generated using top-level Wikipedia categories, but based on an alternative representation of topics.

5.6 Post-Processing

As a further step for data preparation, we produce different variants of most of the features described in this section. In particular, document features are computed based on 5 different retrieval approaches: BM25 [25], the DPH Divergence From Randomness (DFR) model [3], and the Bing, Google, and Yahoo! Web search engines.[7] Additionally, document features are computed at 10 different rank cutoffs: 1, 2, 3, 5, 10, 20, 50, 100, 500, and 1000. Moreover, distributional features (e.g., the number of documents per domain or the pairwise distance between any two retrieved documents) are summarised using up to four different statistics: mean, standard deviation, median, and maximum. Finally, features with no discriminative power across training instances are discarded, and the scores of the remaining features are normalised to lie in the interval [0,1].

[6] http://www.dmoz.org

[7] For Bing, Google, and Yahoo!, documents not present in our target test collection are discarded, and relevance scores are assumed to be a linear function of the logarithm of the rank position of the remaining documents.

6. EXPERIMENTAL SETUP

In this section, we describe the setup for the experiments conducted in Section 7. In particular, these experiments aim to answer the following research questions:

1. Can we effectively diversify the results for a given query by learning its diversification trade-off?
2. What features are effective predictors for the diversification trade-off?
3. How robust is our selective approach to perturbations in the trade-off prediction accuracy?

In the remainder of this section, we detail the test collection, topics, and evaluation metrics used in our experiments. Additionally, we describe the baseline diversification approaches and the different learning regimes considered in the evaluation of our selective diversification approach.

6.1 Collection and Topics

Our analysis is conducted within the standard experimentation paradigm provided by the diversity task of the TREC 2009 Web track [9]. In particular, this task comprises a test collection of 50 topics, which is currently the only publicly available test bed for diversity evaluation on a Web setting. For each topic, from 3 to 8 sub-topics were identified by TREC assessors as representing different aspects of the initial topic, with relevance assessments conducted at the sub-topic level. As the underlying document collection, we consider the category-B ClueWeb09 dataset,[8] as used in the TREC 2009 Web track. In particular, this collection comprises 50 million English documents. We index it using the Terrier platform [23],[9] with Porter's stemmer and standard English stopwords removal.

6.2 Evaluation Metrics

Our experiments use the official evaluation metrics in the diversity task of the TREC 2009 Web track [9]. In order to label training queries, we use α-NDCG@10 [10] as the target metric ϵ within the optimisation procedure described in Section 4.1. The α-normalised discounted cumulative gain (α-NDCG) metric balances relevance and diversity through the tuning parameter α. The larger the value of α, the more diversity is rewarded. Following the standard evaluation practice in TREC 2009, we compute α-NDCG with $\alpha = 0.5$, so as to reward relevance and diversity equally.

Besides α-NDCG, the diversification performance of our approach is also reported in terms of IA-P [1], after a 5-fold cross validation on the TREC 2009 Web track topics. Intent-aware precision (IA-P) averages the traditional notion of precision across different sub-topics, potentially weighted by the relative importance of these sub-topics. Once again, following the standard TREC evaluation, we compute IA-P by considering all sub-topics as equally important.

6.3 Retrieval Baselines

We use MMR [5] and xQuAD [27] as baselines, representing the two broad families of diversification approaches introduced in Section 2. For MMR, we use the cosine distance as the similarity metric. For xQuAD, to isolate the impact

[8] http://boston.lti.cs.cmu.edu/Data/clueweb09/

[9] http://www.terrier.org

of sub-query generation, we use the official TREC 2009 Web track sub-topics as sub-queries, weighted uniformly.[10]

In order to test the consistency of our evaluation results, these diversification baselines are deployed on top of two different ad-hoc retrieval approaches: BM25 [25] and DPH [3]. For efficiency reasons, MMR is applied on the top 100 documents returned by each of these approaches.

6.4 Training Regimes

In our evaluation, five different training regimes are considered in order to set the parameter λ to control the diversification trade-off for both MMR and xQuAD:

1. Uni(base): a baseline uniform diversification regime, with a single λ value learnt for all queries in each fold through a 5-fold cross validation.

2. Uni(oracle): an upper-bound uniform diversification regime, with a single λ value selected to maximise the average performance across all queries.

3. Sel(rand): a baseline selective diversification regime, with a different λ value randomly sampled from the interval [0,1] on a per-query basis.

4. Sel(oracle): an upper-bound selective diversification regime, with a different λ value selected on a per-query basis, so as to maximise the performance of each query individually.

5. Sel(k-nn): a selective diversification regime based on our proposed approach, with a different λ value learnt for each query through a 5-fold cross validation.

To set the k parameter for k-NN, a leave-one-out cross-validation is performed, by minimising mean absolute error (MAE) [36]. Additionally, given the large number of features described in Section 5, we investigate the impact of different feature selection mechanisms for the Sel(k-nn) regime. In particular, besides a simple variant with no feature selection applied (Sel(k-nn,nofs)), we deploy two standard feature selection techniques:

- Sel(k-nn,pca) performs a principal component analysis (PCA) in order to reduce the dimensionality of the feature space [36]. Note that this is an unsupervised process, and hence no training queries are required.

- Sel(k-nn,bfs) performs a best-first search (BFS) over the space of all candidate feature combinations [21]. In particular, to try to avoid converging on a local maximum, we allow negative improvements in the search for the next feature to be added to the current best combination. As a result, our stopping criterion becomes the maximum number of features to be selected: 100.

In the next section, we evaluate our selective approach—using these alternative feature selection techniques—in comparison to the uniform training regimes at predicting the diversification trade-off for MMR and xQuAD.

[10]For effective alternatives on generating sub-queries and estimating their relative importance, please refer to [27, 29].

7. EXPERIMENTAL EVALUATION

In this section, we thoroughly evaluate our proposed approach for selective search result diversification. In particular, in Section 7.1, we assess the effectiveness of our approach for improving the diversification performance of a classical and a state-of-the-art diversification baselines: MMR and xQuAD. In Section 7.2, we analyse the suitability of different groups of features for predicting an effective diversification trade-off. Finally, in Section 7.3, we further investigate the robustness of our approach based on the impact of random perturbations on the prediction of this trade-off.

7.1 Diversification Effectiveness

In this experiment, we aim to answer our first research question, namely, whether learning a different diversification trade-off for different queries results in improved performance. To investigate this, we evaluate the performance of two diversification approaches as baselines for our selective diversification approach. In particular, Tables 2 and 3 show the diversification performance of MMR and xQuAD, respectively, under the training regimes described in the previous section. These include Uni(base) as a baseline uniform diversification regime, and Sel(rand) as a sanity check for the performance of our proposed selective diversification approach using k-NN. Additionally, Uni(oracle) and Sel(oracle) provide upper-bound performances for both a uniform and a selective diversification regime, respectively. Diversification performance is given by α-NDCG and IA-P at different rank cutoffs. The best among the oracle and non-oracle regimes are highlighted in bold. Significance with respect to the Uni(base) regime is given by the Wilcoxon signed-rank matched-pairs test. In particular, $\triangle$ and $\triangledown$ denote a significant increase or decrease with respect to Uni(base) with $p < 0.05$, while ▲ and ▼ denote significant increases or decreases with $p < 0.01$.

From Table 2, based on the performance of our approach using MMR as a baseline, we first observe that Sel(k-nn) improves over Uni(base) in all cases for BM25+MMR, and in most cases for DPH+MMR, often significantly. This attests the effectiveness of our selective approach when compared to learning a single diversification trade-off for all queries. Moreover, Sel(k-nn) significantly improves over Sel(rand) on all settings. This attests the non-triviality of our results. Indeed, randomly assigning the diversification trade-off for MMR performs poorly, which shows the sensitivity of this approach to the accuracy of the learning process. Comparing the different variants of our approach, we note that feature selection plays an important role in the identification of an effective diversification trade-off, particularly when such a large set of features as the one described in Section 5 is employed. In particular, Sel(k-nn,pca), our unsupervised approach based on principal component analysis, brings improvements over no feature selection for most settings. Further improvements are observed when a greedy best-first search feature selection approach is used. Indeed, the performance of Sel(k-nn,bfs) is comparable to the upper-bound performance attained by a uniform diversification, as given by the Uni(oracle) regime.

From Table 3, additional observations can be made about the performance of our approach using xQuAD as a baseline. Firstly, Sel(k-nn) continues to improve over Uni(base). This further confirms our hypothesis that a selective diversification mechanism is effective compared to a uniform di-

	α-NDCG			IA-P		
	@5	@10	@100	@5	@10	@100
BM25+MMR						
UNI(BASE)	0.0918	0.1199	0.2132	0.0448	0.0485	0.0373
SEL(RAND)	0.0559	0.0864	0.1770	0.0260	0.0337	$0.0266^{\triangledown}$
SEL(k-NN,NOFS)	0.1354	0.1610	0.2691	0.0641	$0.0624^{\triangle}$	$0.0541^{\triangle}$
SEL(k-NN,PCA)	$0.1391^{\triangle}$	0.1635	0.2669	$0.0662^{\triangle}$	$0.0641^{\blacktriangle}$	$\mathbf{0.0526}^{\triangle}$
SEL(k-NN,BFS)	$\mathbf{0.1604}^{\blacktriangle}$	$\mathbf{0.1981}^{\blacktriangle}$	$\mathbf{0.2919}^{\blacktriangle}$	$\mathbf{0.0729}^{\blacktriangle}$	$\mathbf{0.0746}^{\blacktriangle}$	$0.0516^{\blacktriangle}$
UNI(ORACLE)	$0.1619^{\blacktriangle}$	$0.1897^{\blacktriangle}$	$0.2943^{\blacktriangle}$	$0.0760^{\blacktriangle}$	$0.0708^{\blacktriangle}$	$0.0575^{\blacktriangle}$
SEL(ORACLE)	$\mathbf{0.1739}^{\blacktriangle}$	$\mathbf{0.2184}^{\blacktriangle}$	$\mathbf{0.2997}^{\blacktriangle}$	$\mathbf{0.0787}^{\blacktriangle}$	$\mathbf{0.0813}^{\blacktriangle}$	$\mathbf{0.0518}^{\blacktriangle}$
DPH+MMR						
UNI(BASE)	0.1778	0.1927	0.2902	0.0957	0.0902	0.0586
SEL(RAND)	$0.0571^{\blacktriangledown}$	$0.0685^{\blacktriangledown}$	$0.1625^{\blacktriangledown}$	$0.0306^{\blacktriangledown}$	$0.0260^{\blacktriangledown}$	$0.0231^{\blacktriangledown}$
SEL(k-NN,NOFS)	0.1875	0.1974	0.2875	0.1017	0.0960	0.0572
SEL(k-NN,PCA)	0.1854	0.2008	0.2930	0.1007	0.0993	0.0560
SEL(k-NN,BFS)	**0.1983**	$\mathbf{0.2162}^{\triangle}$	**0.3027**	**0.1090**	$\mathbf{0.1082}^{\triangle}$	**0.0588**
UNI(ORACLE)	0.1983	0.2121	$0.3042^{\triangle}$	0.1090	0.1064	$\mathbf{0.0621}^{\blacktriangle}$
SEL(ORACLE)	$\mathbf{0.2047}^{\triangle}$	$\mathbf{0.2232}^{\blacktriangle}$	**0.3058**	$\mathbf{0.1115}^{\triangle}$	$\mathbf{0.1091}^{\triangle}$	0.0577

Table 2: Diversification performance of MMR under different training regimes.

	α-NDCG			IA-P		
	@5	@10	@100	@5	@10	@100
BM25+xQuAD						
UNI(BASE)	0.2209	0.2483	0.3416	0.1038	0.0961	0.0599
SEL(RAND)	0.2146	0.2383	$0.3304^{\triangledown}$	0.0983	0.0891	$0.0563^{\blacktriangledown}$
SEL(k-NN,NOFS)	0.2145	$0.2350^{\triangledown}$	0.3327	0.0977	0.0862	$0.0578^{\triangledown}$
SEL(k-NN,PCA)	0.2194	0.2405	0.3418	0.0959	0.0887	0.0592
SEL(k-NN,BFS)	**0.2579**	**0.2834**	**0.3716**	**0.1184**	**0.1128**	**0.0602**
UNI(ORACLE)	0.2314	0.2604	0.3457	0.1116	0.1059	0.0611
SEL(ORACLE)	**0.3040**	**0.3162**	**0.3977**	**0.1335**	**0.1103**	**0.0595**
DPH+xQuAD						
UNI(BASE)	0.2569	0.2739	0.3637	0.1374	0.1154	0.0609
SEL(RAND)	0.2396	0.2577	$0.3458^{\blacktriangledown}$	0.1158	$0.0995^{\triangledown}$	$0.0572^{\blacktriangledown}$
SEL(k-NN,NOFS)	0.2543	0.2670	0.3546	**0.1403**	**0.1220**	**0.0634**
SEL(k-NN,PCA)	0.2458	0.2758	**0.3639**	0.1234	0.1130	0.0612
SEL(k-NN,BFS)	**0.2653**	**0.2833**	0.3636	0.1349	0.1172	0.0602
UNI(ORACLE)	0.2585	0.2785	0.3641	0.1400	0.1177	0.0611
SEL(ORACLE)	$\mathbf{0.3314}^{\blacktriangle}$	$\mathbf{0.3520}^{\blacktriangle}$	$\mathbf{0.4122}^{\blacktriangle}$	$\mathbf{0.1633}^{\triangle}$	$\mathbf{0.1488}^{\blacktriangle}$	**0.0634**

Table 3: Diversification performance of xQuAD under different training regimes.

versification, even for a state-of-the-art diversification baseline such as xQuAD. Feature selection plays an even more important role for xQuAD compared to MMR. In particular, while using all available features (the SEL(k-NN,NOFS) variant) hardly improves over a random assignment of the diversification trade-off for BM25+xQuAD, performing feature selection can bring substantial improvements. Indeed, SEL(k-NN,BFS) can outperform even an oracle uniform assignment of the diversification trade-off for BM25+xQuAD. For DPH+xQuAD, both SEL(k-NN,PCA) and SEL(k-NN,BFS) improve over the oracle uniform diversification. Comparing the SEL(RAND) regime in Tables 2 and 3 reveals an interesting behaviour. While the performance of MMR is highly sensitive to a random assignment of the diversification trade-off, xQuAD still performs relatively well in this scenario. Nonetheless, such a resilient behaviour from xQuAD does not mean it would benefit less from our selective approach. Indeed, the upper-bound performance of SEL(ORACLE) gives encouraging room for further improvements.

Recalling our first research question, the results in Tables 2 and 3 attest the effectiveness of our selective diversification approach, with significant improvements over a uniform diversification across different experimental settings.

7.2 Feature Group Performance

The results in Section 7.1 are particularly promising given the simple techniques we deployed to select a subset of effective features from the large pool used in this work. Although automatically finding an optimal subset of these features is beyond the scope of this paper, in this section, we investigate the predictive power of different groups of features. In particular, we aim to answer our second research question, concerning the usefulness of different features for learning the diversification trade-off for an unseen query.

Inspired by our proposed classification of the features described in Section 5, we analyse the performance of our selective diversification approach using features from five different groups: query concept identification (QCI), query performance prediction (QPP), query topic classification (QTC), query type detection (QTD), and query log mining (QLM). In particular, Table 4 shows the performance of our selective diversification approach for both MMR and xQuAD, with features grouped according to the aforementioned classification. In each row, only features of the corresponding category are selected (e.g., SEL(k-NN,QCI) comprises only query concept identification features). SEL(k-NN,NOFS) provides a baseline performance, with no feature selection.

	MMR		xQuAD	
	BM25	DPH	BM25	DPH
SEL(k-NN,NOFS)	0.1610	0.1974	0.2350	0.2670
SEL(k-NN,QCI)	0.1656	0.1591	0.2393	0.2611
SEL(k-NN,QLM)	**0.1743**	0.1824	0.2399	0.2463
SEL(k-NN,QPP)	0.1684	0.1703	0.2447	0.2574
SEL(k-NN,QTC)	0.1721	**0.2030**	0.2384	**0.2844**
SEL(k-NN,QTD)	0.1640	0.1819	**0.2511**	0.2490
Pearson's ρ		0.53		-0.52

Table 4: Per-feature group diversification performance in terms of α-NDCG@10.

From Table 4, compared to using all features (i.e., SEL(k-NN,NOFS)), we first observe that selecting features according to any of the proposed categories improves the performance of our selective approach for both MMR and xQuAD on top of BM25. However, when DPH is used as the underlying weighting model, only the query type classification features (QTC) provide an effective feature selection. Indeed, and recalling our second research question, our QTC features constitute the most robust group of all features considered in this work, with consistent improvements across all settings. This observation also agrees with the output of our greedy best-first search feature selection approach. In particular, DocEntityCount, DocEntityEntropy, DocEntityPairwiseCosine, and WPCategoryCount are among the most useful features for predicting the diversification trade-off. Other features selected through best-first search include two query concept identification (QCI) features (WPDisambCount and WPDisambSenses), one query type detection (QTD) feature (HostDistribution), and one query performance prediction (QPP) feature (QueryFeedback). In common, all of these features are computed from the documents retrieved for a query. This suggests that features derived from the query itself or a query log are not as effective predictors for the diversification trade-off. In the latter case, this might be due to the sparsity of clicks and reformulations for the considered test queries in the MSN query log.

Finally, another interesting observation relates to how different diversification approaches leverage different features. While MMR shows similar performances across different feature groups using either BM25 or DPH ($\rho = 0.53$), the most useful features for xQuAD do not agree across these two weighting models ($\rho = -0.52$). Although anecdotal, these observations illustrate the challenge of selecting a suitable subset of features for learning an effective diversification trade-off for different diversification approaches.

7.3 Prediction Robustness

In the previous sections, we have shown that our approach can be effective even when deploying relatively simple feature selection techniques to reduce the dimensionality of our feature space. In this section, we investigate the reasons for such a robust behaviour. More precisely, we aim to answer our third research question, regarding the sensitivity of our approach to perturbations in the underlying regression accuracy. To do so, we propose a simple perturbation criterion, which introduces randomness in the regression process. In particular, we predict a diversification trade-off λ for a training query q according to a linear combination:

$$\lambda = (1 - \phi)\lambda^* + \phi\lambda_{\text{rand}} \quad (3)$$

where λ^* is the optimal trade-off for the training query q, obtained as described in Section 4.1, and λ_{rand} is a random number in the interval [0,1]. The interpolation parameter ϕ represents the perturbation level. When $\phi = 0$, we have a perfect prediction accuracy, equivalent to our upper-bound regime SEL(ORACLE). On the other extreme, when $\phi = 1$, we have a completely random prediction accuracy, equivalent to our baseline regime SEL(RAND). Figure 2 shows the diversification performance of SEL(ORACLE) in terms of α-NDCG@10 for different levels of prediction perturbation, using DPH+xQuAD. The UNI(ORACLE) regime is also included as an upper-bound uniform diversification approach.

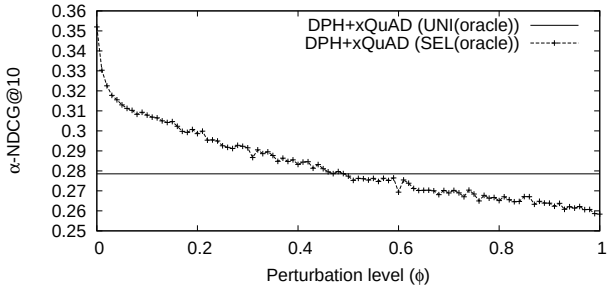


Figure 2: Diversification performance across different levels of prediction perturbation.

From Figure 2, we make two main observations. Firstly, our selective diversification approach is very robust to perturbations in regression accuracy, outperforming an upper-bound uniform diversification even with up to 50% of accuracy perturbation, which answers our third research question. This is remarkable, and confirms the effectiveness of our approach, despite the inherent difficulty of the prediction task. A second observation relates to how close to the upper-bound performance we can expect to be in a realistic scenario. From Figure 2, we can observe that gradual improvements are attained as the level of perturbation drops. However, after a certain level, further improvements seem unlikely, as they would require a near-perfect regression accuracy. In this example, we could expect the upper-bound performance of DPH+xQuAD in terms of α-NDCG@10 to lie in between 0.31 and 0.32 in a more realistic scenario.

8. CONCLUSIONS

In this paper, we have introduced a novel selective approach for search result diversification. In particular, our approach predicts not only whether a particular query could benefit from diversification, but also to what extent its results should be diversified. Our thorough experiments have shown that our approach is effective and can significantly outperform a uniform diversification strategy.

By improving a classical and a state-of-the-art diversification approaches from rather distinct families, we have shown that our approach is general and agnostic to any particular baseline diversification approach. Moreover, by deploying a large pool of features while relying on relatively simple feature selection techniques, we have shown that our approach is also robust to perturbations in the prediction accuracy.

While feature selection is a challenging research problem in itself [14], given the high dimensionality of the learning task tackled in this work and the limited amount of training queries we had available, our results are even more promising. In fact, we believe we have only scratched the surface of an emerging field. As illustrated by the performance of an oracle selective diversification regime, substantial improve-

ments in diversification performance should be possible by deploying even more effective and sophisticated feature selection and learning techniques.

9. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*, pages 5–14, 2009.

[2] D. W. Aha, D. F. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[3] G. Amati, E. Ambrosi, M. Bianchi, C. Gaibisso, and G. Gambosi. FUB, IASI-CNR and University of Tor Vergata at TREC 2007 Blog track. In *TREC*, 2007.

[4] A. Broder. A taxonomy of Web search. *SIGIR Forum*, 36(2):3–10, 2002.

[5] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336, 1998.

[6] D. Carmel and E. Yom-Tov. Estimating the query difficulty for information retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 2(1):1–89, 2010.

[7] B. Carterette and P. Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *CIKM*, pages 1287–1296, 2009.

[8] H. Chen and D. R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *SIGIR*, pages 429–436, 2006.

[9] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2009 Web track. In *TREC*, 2009.

[10] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR*, pages 659–666, 2008.

[11] P. Clough, M. Sanderson, M. Abouammoh, S. Navarro, and M. Paramita. Multiple approaches to analysing query diversity. In *SIGIR*, pages 734–735, 2009.

[12] N. Craswell and D. Hawking. Overview of the TREC 2004 Web track. In *TREC*, 2004.

[13] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *SIGIR*, pages 299–306, 2002.

[14] E. Gabrilovich, A. Smola, and N. Tishby, editors. *SIGIR Workshop on Feature Generation and Selection for IR*, 2010.

[15] X. Geng, T.-Y. Liu, T. Qin, A. Arnold, H. Li, and H.-Y. Shum. Query dependent ranking using k-nearest neighbor. In *SIGIR*, pages 115–122, 2008.

[16] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW*, pages 381–390, 2009.

[17] B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. In *SPIRE*, pages 43–54, 2004.

[18] B. He and I. Ounis. Query performance prediction. *Inf. Syst.*, 31(7):585–594, 2006.

[19] I.-H. Kang and G. Kim. Query type classification for Web document retrieval. In *SIGIR*, pages 64–71, 2003.

[20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[21] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1-2):273–324, 1997.

[22] S. M. Omohundro. Five balltree construction algorithms. Technical Report TR-89-063, International Computer Science Institute, 1989.

[23] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: a high performance and scalable information retrieval platform. In *OSIR*, 2006.

[24] J. Peng, C. Macdonald, and I. Ounis. Learning to select a ranking function. In *ECIR*, pages 114–126, 2010.

[25] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *TREC*, 1994.

[26] M. Sanderson. Ambiguous queries: Test collections need more sense. In *SIGIR*, pages 499–506, 2008.

[27] R. L. T. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for Web search result diversification. In *WWW*, pages 881–890, 2010.

[28] R. L. T. Santos, C. Macdonald, and I. Ounis. Voting for related entities. In *RIAO*, 2010.

[29] R. L. T. Santos, J. Peng, C. Macdonald, and I. Ounis. Explicit search result diversification through sub-queries. In *ECIR*, pages 87–99, 2010.

[30] F. Silvestri. Mining query logs: turning search usage data into knowledge. *Found. Trends Inf. Retr.*, 4(1–2):1–174, 2010.

[31] R. Song, Z. Luo, J.-Y. Nie, Y. Yu, and H.-W. Hon. Identification of ambiguous queries in Web search. *Inf. Process. Manage.*, 45(2):216–229, 2009.

[32] R. Song, J.-R. Wen, S. Shi, G. Xin, T.-Y. Liu, T. Qin, X. Zheng, J. Zhang, G.-R. Xue, and W.-Y. Ma. Microsoft Research Asia at Web track and Terabyte track of TREC 2004. In *TREC*, 2004.

[33] K. Spärck-Jones, S. E. Robertson, and M. Sanderson. Ambiguous requests: implications for retrieval tests, systems and theories. *SIGIR Forum*, 41(2):8–17, 2007.

[34] J. Wang and J. Zhu. Portfolio theory of information retrieval. In *SIGIR*, pages 115–122, 2009.

[35] Y. Wang and E. Agichtein. Query ambiguity revisited: clickthrough measures for distinguishing informational and ambiguous queries. In *NAACL-HLT 2010*, pages 361–364, 2010.

[36] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools*. Morgan Kaufmann, 2005.

[37] E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow. Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *SIGIR*, pages 512–519, 2005.

[38] C. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR*, pages 10–17, 2003.

[39] Y. Zhou and W. B. Croft. Query performance prediction in Web search environments. In *SIGIR*, pages 543–550, 2007.