Performance Prediction Using Spatial Autocorrelation

Fernando Diaz
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
fdiaz@cs.umass.edu

ABSTRACT

Evaluation of information retrieval systems is one of the core tasks in information retrieval. Problems include the inability to exhaustively label all documents for a topic, non-generalizability from a small number of topics, and incorporating the variability of retrieval systems. Previous work addresses the evaluation of systems, the ranking of queries by difficulty, and the ranking of individual retrievals by performance. Approaches exist for the case of few and even no relevance judgments. Our focus is on zero-judgment performance prediction of individual retrievals.

One common shortcoming of previous techniques is the assumption of uncorrelated document scores and judgments. If documents are embedded in a high-dimensional space (as they often are), we can apply techniques from spatial data analysis to detect correlations between document scores. We find that the low correlation between scores of topically close documents often implies a poor retrieval performance. When compared to a state of the art baseline, we demonstrate that the spatial analysis of retrieval scores provides significantly better prediction performance. These new predictors can also be incorporated with classic predictors to improve performance further. We also describe the first large-scale experiment to evaluate zero-judgment performance prediction for a massive number of retrieval systems over a variety of collections in several languages.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models; H.3.4 [**Systems and Software**]: Performance evaluation (efficiency and effectiveness)

General Terms

Performance, Design, Reliability, Experimentation

Keywords

autocorrelation, regularization, performance prediction

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.

1. INTRODUCTION

In information retrieval, a user poses a query to a system. The system retrieves n documents each receiving a real-valued *score* indicating the predicted degree of relevance. If we randomly select pairs of documents from this set, we expect some pairs to share the same topic and other pairs to not share the same topic. Take two topically-related documents from the set and call them a and b. If the scores of a and b are very different, we may be concerned about the performance of our system. That is, if a and b are both on the topic of the query, we would like them *both* to receive a high score; if a and b are not on the topic of the query, we would like them *both* to receive a low score. We might become more worried as we find more differences between scores of related documents. We would be more comfortable with a retrieval where scores are consistent between related documents.

Our paper studies the quantification of this inconsistency in a retrieval from a spatial perspective. Spatial analysis is appropriate since many retrieval models embed documents in some vector space. If documents are embedded in a space, proximity correlates with topical relationships. Score consistency can be measured by the spatial version of autocorrelation known as the Moran coefficient or I_M [5, 10]. In this paper, we demonstrate a strong correlation between I_M and retrieval performance.

The discussion up to this point is reminiscent of the cluster hypothesis. The cluster hypothesis states: *closely-related documents tend to be relevant to the same request* [12]. As we shall see, a retrieval function's spatial autocorrelation measures the degree to which closely-related documents receive similar scores. Because of this, we interpret autocorrelation as measuring the degree to which a retrieval function satisfies the clustering hypothesis. If this connection is reasonable, in Section 6, we present evidence that failure to satisfy the cluster hypothesis correlates strongly with poor performance.

In this work, we provide the following contributions,

1. A general, robust method for predicting the performance of retrievals with zero relevance judgments (Section 3).

2. A theoretical treatment of the similarities and motivations behind several state-of-the-art performance prediction techniques (Section 4).

3. The first large-scale experiments of zero-judgment, single run performance prediction (Sections 5 and 6).

2. PROBLEM DEFINITION

Given a *query*, an information retrieval *system* produces a ranking of documents in the collection encoded as a set of scores associated with documents. We refer to the set of scores for a particular query-system combination as a *retrieval*. We would like to predict the performance of this retrieval with respect to some evaluation measure (eg, mean average precision). In this paper, we present results for ranking retrievals from arbitrary systems. We would like this ranking to approximate the ranking of retrievals by the evaluation measure. This is different from *ranking queries* by the average performance on each query. It is also different from *ranking systems* by the average performance on a set of queries.

Scores are often only computed for the top n documents from the collection. We place these scores in the length n vector, $\mathbf{y}$, where y_i refers to the score of the ith-ranked document. We adjust scores to have zero mean and unit variance. We use this method because of its simplicity and its success in previous work [15].

3. SPATIAL CORRELATION

In information retrieval, we often assume that the representations of documents exist in some high-dimensional vector space. For example, given a vocabulary, $\mathcal{V}$, this vector space may be an arbitrary $|\mathcal{V}|$-dimensional space with cosine inner-product or a multinomial simplex with a distribution-based distance measure. An embedding space is often selected to respect topical proximity; if two documents are near, they are more likely to share a topic.

Because of the prevalence and success of spatial models of information retrieval, we believe that the application of spatial data analysis techniques are appropriate. Whereas in information retrieval, we are concerned with the score at a point in a space, in spatial data analysis, we are concerned with the value of a function at a point or *location* in a space. We use the term function here to mean a mapping from a location to a real value. For example, we might be interested in the prevalence of a disease in the neighborhood of some city. The function would map the location of a neighborhood to an infection rate.

If we want to quantify the spatial dependencies of a function, we would employ a measure referred to as the *spatial autocorrelation* [5, 10]. High spatial autocorrelation suggests that knowing the value of a function at location a will tell us a great deal about the value at a neighboring location b. There is a high spatial autocorrelation for a function representing the temperature of a location since knowing the temperature at a location a will tell us a lot about the temperature at a neighboring location b. Low spatial autocorrelation suggests that knowing the value of a function at location a tells us little about the value at a neighboring location b. There is low spatial autocorrelation in a function measuring the outcome of a coin toss at a and b.

In this section, we will begin by describing what we mean by spatial proximity for documents and then define a measure of spatial autocorrelation. We conclude by extending this model to include information from multiple retrievals from multiple systems for a single query.

3.1 Spatial Representation of Documents

Our work does not focus on improving a specific similarity measure or defining a novel vector space. Instead, we choose an inner product known to be effective at detecting inter-document topical relationships. Specifically, we adopt tf.idf document vectors,

$$\tilde{d}_i = d_i \log\left(\frac{(n+0.5)-c_i}{0.5+c_i}\right) \quad (1)$$

where $\mathbf{d}$ is a vector of term frequencies, $\mathbf{c}$ is the length-$|\mathcal{V}|$ document frequency vector. We use this weighting scheme due to its success for topical link detection in the context of Topic Detection and Tracking (TDT) evaluations [6]. Assuming vectors are scaled by their L_2 norm, we use the inner product, $\langle \tilde{\mathbf{d}}_i, \tilde{\mathbf{d}}_j \rangle$, to define similarity.

Given documents and some similarity measure, we can construct a matrix which encodes the similarity between pairs of documents. Recall that we are given the top n documents retrieved in $\mathbf{y}$. We can compute an $n \times n$ similarity matrix, $\mathbf{W}$. An element of this matrix, W_{ij} represents the similarity between documents ranked i and j. In practice, we only include the affinities for a document's k-nearest neighbors. In all of our experiments, we have fixed k to 5. We leave exploration of parameter sensitivity to future work. We also row normalize the matrix so that $\sum_{j=1}^{n} W_{ij} = 1$ for all i.

3.2 Spatial Autocorrelation of a Retrieval

Recall that we are interested in measuring the similarity between the scores of spatially-close documents. One such suitable measure is the Moran coefficient of spatial autocorrelation. Assuming the function $\mathbf{y}$ over n locations, this is defined as

$$\begin{aligned}\tilde{I}_M &= \frac{n}{\mathbf{e}^\mathsf{T}\mathbf{W}\mathbf{e}} \frac{\sum_{i,j} W_{ij} y_i y_j}{\sum_i y_i^2} \\ &= \frac{n}{\mathbf{e}^\mathsf{T}\mathbf{W}\mathbf{e}} \frac{\mathbf{y}^\mathsf{T}\mathbf{W}\mathbf{y}}{\mathbf{y}^\mathsf{T}\mathbf{y}} \end{aligned} \quad (2)$$

where $\mathbf{e}^\mathsf{T}\mathbf{W}\mathbf{e} = \sum_{ij} W_{ij}$.

We would like to compare autocorrelation values for different retrievals. Unfortunately, the bound for Equation 2 is not consistent for different $\mathbf{W}$ and $\mathbf{y}$. Therefore, we use the Cauchy-Schwartz inequality to establish a bound,

$$\tilde{I}_M \leq \frac{n}{\mathbf{e}^\mathsf{T}\mathbf{W}\mathbf{e}} \sqrt{\frac{\mathbf{y}^\mathsf{T}\mathbf{W}^\mathsf{T}\mathbf{W}\mathbf{y}}{\mathbf{y}^\mathsf{T}\mathbf{y}}}$$

And we define the normalized spatial autocorrelation as

$$I_M = \frac{\mathbf{y}^\mathsf{T}\mathbf{W}\mathbf{y}}{\sqrt{\mathbf{y}^\mathsf{T}\mathbf{y} \times \mathbf{y}^\mathsf{T}\mathbf{W}^\mathsf{T}\mathbf{W}\mathbf{y}}}$$

Notice that if we let $\tilde{\mathbf{y}} = \mathbf{W}\mathbf{y}$, then we can write this formula as,

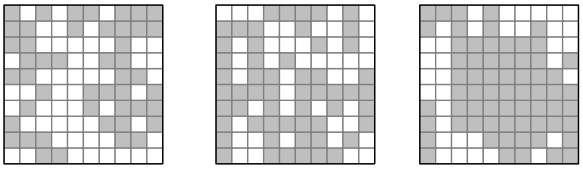
$$I_M = \frac{\mathbf{y}^\mathsf{T}\tilde{\mathbf{y}}}{\|\mathbf{y}\|_2 \|\tilde{\mathbf{y}}\|_2} \quad (3)$$

which can be interpreted as the correlation between the original retrieval scores and a set of retrieval scores "diffused" in the space.

We present some examples of autocorrelations of functions on a grid in Figure 1.

3.3 Correlation with Other Retrievals

Sometimes we are interested in the performance of a single retrieval but have access to scores from multiple systems for



(a) $I_M = 0.006$ (b) $I_M = 0.241$ (c) $I_M = 0.487$

Figure 1: The Moran coefficient, I_M for a several binary functions on a grid. The Moran coefficient is a local measure of function consistency. From the perspective of information retrieval, each of these grid spaces would represent a document and documents would be organized so that they lay next to topically-related documents. Binary retrieval scores would define a pattern on this grid. Notice that, as the Moran coefficient increases, neighboring cells tend to have similar values.

the same query. In this situation, we can use combined information from these scores to construct a surrogate for a high-quality ranking [17]. We can treat the correlation between the retrieval we are interested in and the combined scores as a predictor of performance.

Assume that we are given m score functions, $\mathbf{y}_i$, for the same n documents. We will represent the mean of these vectors as $\mathbf{y}_\mu = \sum_{i=1}^{m} \mathbf{y}_i$. We use the mean vector as an approximation to relevance. Since we use zero mean and unit variance normalization, work in metasearch suggests that this assumption is justified [15]. Because $\mathbf{y}_\mu$ represents a very good retrieval, we hypothesize that a strong similarity between $\mathbf{y}_\mu$ and $\mathbf{y}$ will correlate positively with system performance. We use Pearson's product-moment correlation to measure the similarity between these vectors,

$$\rho(\mathbf{y}, \mathbf{y}_\mu) = \frac{\mathbf{y}^\mathsf{T}\mathbf{y}_\mu}{\|\mathbf{y}\|_2 \|\mathbf{y}_\mu\|_2} \tag{4}$$

We will comment on the similarity between Equation 3 and 4 in Section 7.

Of course, we can combine $\rho(\mathbf{y}, \tilde{\mathbf{y}})$ and $\rho(\mathbf{y}, \mathbf{y}_\mu)$ if we assume that they capture different factors in the prediction. One way to accomplish this is to combine these predictors as independent variables in a linear regression. An alternative means of combination is suggested by the mathematical form of our predictors. Since $\tilde{\mathbf{y}}$ encodes the spatial dependencies in $\mathbf{y}$ and $\mathbf{y}_\mu$ encodes the spatial properties of the multiple runs, we can compute a third correlation between these two vectors,

$$\rho(\tilde{\mathbf{y}}, \mathbf{y}_\mu) = \frac{\tilde{\mathbf{y}}^\mathsf{T}\mathbf{y}_\mu}{\|\tilde{\mathbf{y}}\|_2 \|\mathbf{y}_\mu\|_2} \tag{5}$$

We can interpret Equation 5 as measuring the correlation between a high quality ranking ($\mathbf{y}_\mu$) and a spatially smoothed version of the retrieval ($\tilde{\mathbf{y}}$).

4. RELATIONSHIP WITH OTHER PREDICTORS

One way to predict the effectiveness of a retrieval is to look at the shared vocabulary of the top n retrieved documents. If we computed the most frequent content words in this set, we would hope that they would be consistent with our topic. In fact, we might believe that a bad retrieval would include documents on many disparate topics, resulting in an overlap of terminological noise. The Clarity of a query attempts to quantify exactly this [7]. Specifically, Clarity measures the similarity of the words most frequently used in retrieved documents to those most frequently used in the whole corpus. The conjecture is that a good retrieval will use language distinct from general text; the overlapping language in a bad retrieval will tend to be more similar to general text. Mathematically, we can compute a representation of the language used in the initial retrieval as a weighted combination of document *language models*,

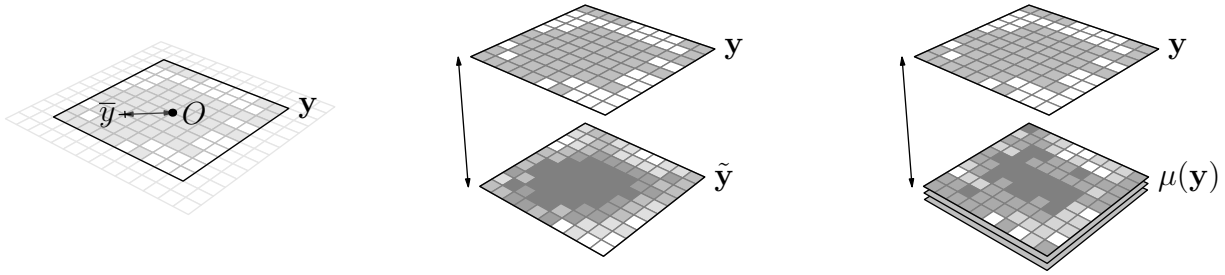
$$P(w|\theta_Q) = \sum_{i=1}^{n} P(w|\theta_i) \frac{P(Q|\theta_i)}{\mathcal{Z}} \tag{6}$$

where θ_i is the language model of the ith-ranked document, $P(Q|\theta_i)$ is the query likelihood score of the ith-ranked document and $\mathcal{Z} = \sum_{i=1}^{n} P(Q|\theta_i)$ is a normalization constant. The similarity between the multinomial $P(w|\theta_Q)$ and a model of "general text" can be computed using the Kullback-Leibler divergence, $D_{KL}^{\mathcal{V}}(\theta_Q \| \theta_C)$. Here, the distribution $P(w|\theta_C)$ is our model of general text which can be computed using term frequencies in the corpus. In Figure 2a, we present Clarity as measuring the distance between the "weighted center of mass" of the retrieval (labeled $\overline{y}$) and the "unweighted center of mass" of the collection (labeled O). Clarity reaches a minimum when a retrieval assigns every document the same score.

Let's again assume we have a set of n documents retrieved for our query. Another way to quantify the dispersion of a set of documents is to look at how clustered they are. We may hypothesize that a good retrieval will return a single, tight cluster. A poorly performing retrieval will return a loosely related set of documents covering many topics. One proposed method of quantifying this dispersion is to measure the distance from a random document a to it's nearest neighbor, b. A retrieval which is tightly clustered will, on average, have a low distance between a and b; a retrieval which is less tightly-closed will, on average have high distances between a and b. This average corresponds to using the Cox-Lewis statistic to measure the randomness of the top n documents retrieved from a system [18]. In Figure 2a, this is roughly equivalent to measuring the area of the set n. Notice that we are throwing away information about the retrieval function $\mathbf{y}$. Therefore the Cox-Lewis statistic is highly dependent on selecting the top n documents.[1]

Remember that we have n documents and a set of scores. Let's assume that we have access to the system which provided the original scores and that we can also request scores for new documents. This suggests a third method for predicting performance. Take some document, a, from the retrieved set and arbitrarily add or remove words at random to create a new document $\tilde{a}$. Now, we can ask our system to score $\tilde{a}$ with respect to our query. If, on average over the n documents, the scores of a and $\tilde{a}$ tend to be very different, we might suspect that the system is failing on this query. So, an alternative approach is to measure the simi-

[1]The authors have suggested coupling the query with the distance measure [18]. The information introduced by the query, though, is retrieval-independent so that, if two retrievals return the same set of documents, the approximate Cox-Lewis statistic will be the same *regardless of the retrieval scores.*



(a) Global Divergence (b) Score Perturbation (c) Multirun Averaging

Figure 2: Representation of several performance predictors on a grid. In Figure 2a, we depict predictors which measure the divergence between the "center of mass" of a retrieval and the center of the embedding space. In Figure 2b, we depict predictors which compare the original retrieval, y, to a perturbed version of the retrieval, $\tilde{\mathbf{y}}$. Our approach uses a particular type of perturbation based on score diffusion. Finally, in Figure 2c, we depict prediction when given retrievals from several other systems on the same query. Here, we can consider the fusion of these retrieval as a surrogate for relevance.

larity between the retrieval and a perturbed version of that retrieval [18, 19]. This can be accomplished by either perturbing the documents or queries. The similarity between the two retrievals can be measured using some correlation measure. This is depicted in Figure 2b. The upper grid represents the original retrieval, $\mathbf{y}$, while the lower grid represents the function after having been perturbed, $\tilde{\mathbf{y}}$. The nature of the perturbation process requires additional scorings or retrievals. Our predictor does not require access to the original scoring function or additional retrievals. So, although our method is similar to other perturbation methods in spirit, it can be applied in situations when the retrieval system is inaccessible or costly to access.

Finally, assume that we have, in addition to the retrieval we want to evaluate, m retrievals from a variety of different systems. In this case, we might take a document a, compare its rank in the retrieval to its *average rank* in the m retrievals. If we believe that the m retrievals provide a satisfactory approximation to relevance, then a very large difference in rank would suggest that our retrieval is misranking a. If this difference is large on average over all n documents, then we might predict that the retrieval is bad. If, on the other hand, the retrieval is very consistent with the m retrievals, then we might predict that the retrieval is good. The similarity between the retrieval and the combined retrieval may be computed using some correlation measure. This is depicted in Figure 2c. In previous work, the Kullback-Leibler divergence between the normalized scores of the retrieval and the normalized scores of the combined retrieval provides the similarity [1].

5. EXPERIMENTS

Our experiments focus on testing the predictive power of each of our predictors: $\rho(\mathbf{y}, \tilde{\mathbf{y}})$, $\rho(\mathbf{y}, \mathbf{y}_\mu)$, and $\rho(\tilde{\mathbf{y}}, \mathbf{y}_\mu)$. As stated in Section 2, we are interested in predicting the performance of the retrieval generated by an arbitrary system. Our methodology is consistent with previous research in that we predict the *relative performance* of a retrieval by comparing a ranking based on our predictor to a ranking based on average precision.

We present results for two sets of experiments. The first set of experiments presents detailed comparisons of our predictors to previously-proposed predictors using identical data sets. Our second set of experiments demonstrates the generalizability of our approach to arbitrary retrieval methods, corpus types, and corpus languages.

5.1 Detailed Experiments

In these experiments, we will predict the performance of language modeling scores using our autocorrelation predictor, $\rho(\mathbf{y}, \tilde{\mathbf{y}})$; we do not consider $\rho(\mathbf{y}, \mathbf{y}_\mu)$ or $\rho(\tilde{\mathbf{y}}, \mathbf{y}_\mu)$ because, in these detailed experiments, we focus on ranking the retrievals from a single system. We use retrievals, values for baseline predictors, and evaluation measures reported in previous work [19].

5.1.1 Topics and Collections

These performance prediction experiments use language model retrievals performed for queries associated with collections in the TREC corpora. Using TREC collections allows us to confidently associate an average precision with a retrieval. In these experiments, we use the following topic collections: TREC 4 ad hoc, TREC 5 ad hoc, Robust 2004, Terabyte 2004, and Terabyte 2005.

5.1.2 Baselines

We provide two baselines. Our first baseline is the classic Clarity predictor presented in Equation 6. Clarity is designed to be used with language modeling systems. Our second baseline is Zhou and Croft's "ranking robustness" predictor. This predictor corrupts the top k documents from retrieval and re-computes the language model scores for these corrupted documents. The value of the predictor is the Spearman rank correlation between the original ranking and the corrupted ranking. In our tables, we will label results for Clarity using $D_{KL}^{\mathcal{V}}$ and the ranking robustness predictor using P.

5.2 Generalizability Experiments

Our predictors do not require a particular baseline retrieval system; the predictors can be computed for an arbitrary retrieval, regardless of how scores were generated. We believe that that is one of the most attractive aspects of our algorithm. Therefore, in a second set of experiments, we demonstrate the ability of our techniques to generalize to a variety of collections, topics, and retrieval systems.

5.2.1 Topics and Collections

We gathered a diverse set of collections from all possible TREC corpora. We cast a wide net in order to locate collections where our predictors might fail. Our hypothesis is that documents with high topical similarity should have correlated scores. Therefore, we avoided collections where scores were unlikely to be correlated (eg, question-answering) or were likely to be negatively correlated (eg, novelty). Nevertheless, our collections include corpora where correlations are weakly justified (eg, non-English corpora) or not justified at all (eg, expert search). We use the ad hoc tracks from TREC3-8, TREC Robust 2003-2005, TREC Terabyte 2004-2005, TREC4-5 Spanish, TREC5-6 Chinese, and TREC Enterprise Expert Search 2005. In all cases, we use only the automatic runs for ad hoc tracks submitted to NIST.

For all English and Spanish corpora, we construct the matrix $\mathbf{W}$ according to the process described in Section 3.1. For Chinese corpora, we use naïve character-based tf.idf vectors. For entities, entries in $\mathbf{W}$ are proportional to the number of documents in which two entities cooccur.

5.2.2 Baselines

In our detailed experiments, we used the Clarity measure as a baseline. Since we are predicting the performance of retrievals which are not based on language modeling, we use a version of Clarity referred to as ranked-list Clarity [7]. Ranked-list clarity converts document ranks to $P(Q|\theta_i)$ values. This conversion begins by replacing all of the scores in $\mathbf{y}$ with the respective ranks. Our estimation of $P(Q|\theta_i)$ from the ranks, then is,

$$P(Q|\theta_i) = \begin{cases} \frac{2(c+1-y_i)}{c(c+1)} & \text{if } y_i \leq c \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where c is a cutoff parameter. As suggested by the authors, we fix the algorithm parameters c and λ_2 so that $c = 60$ and $\lambda_2 = 0.10$. We use Equation 6 to estimate $P(w|\theta_Q)$ and $D_{KL}^{\mathcal{V}}(\theta_Q \| \theta_C)$ to compute the value of the predictor. We will refer to this predictor as $D_{KL}^{\mathcal{V}}$, superscripted by $\mathcal{V}$ to indicate that the Kullback-Leibler divergence is with respect to the term embedding space.

When information from multiple runs on the same query is available, we use Aslam and Pavlu's document-space multinomial divergence as a baseline [1]. This rank-based method first normalizes the scores in a retrieval as an n-dimensional multinomial. As with ranked-list Clarity, we begin by replacing all of the scores in $\mathbf{y}$ with their respective ranks. Then, we adjust the elements of $\mathbf{y}$ in the following way,

$$\hat{y}_i = \frac{1}{2n}\left(1 + \sum_{k=y_i}^{n} \frac{1}{k}\right) \quad (8)$$

In our multirun experiments, we only use the top 75 documents from each retrieval ($n = 75$); this is within the range of parameter values suggested by the authors. However, we admit not tuning this parameter for either our system or the baseline. The predictor is the divergence between the candidate distribution, $\mathbf{y}$, and the mean distribution, $\mathbf{y}_\mu$. With the uniform linear combination of these m retrievals represented as $\mathbf{y}_\mu$, we can compute the divergence as $D_{KL}^{n}(\hat{\mathbf{y}} \| \hat{\mathbf{y}}_\mu)$ where we use the superscript n to indicate that the summation is over the set of n documents. This baseline was developed in the context of predicting query difficulty but we adopt it as a reasonable baseline for predicting retrieval performance.

5.2.3 Parameter Settings

When given multiple retrievals, we use documents in the union of the top $k = 75$ documents from each of the m retrievals for that query. If the size of this union is $\tilde{n}$, then $\mathbf{y}_\mu$ and each $\mathbf{y}_i$ is of length $\tilde{n}$. In some cases, a system did not score a document in the union. Since we are making a Gaussian assumption about our scores, we can sample scores for these unseen documents from the negative tail of the distribution. Specifically, we sample from the part of the distribution lower than the minimum value of in the normalized retrieval. This introduces randomness into our algorithm but we believe it is more appropriate than assigning an arbitrary fixed value.

We optimized the linear regression using the square root of each predictor. We found that this substantially improved fits for all predictors, including the baselines. We considered linear combinations of pairs of predictors (labeled by the components) and all predictors (labeled as β).

5.3 Evaluation

Given a set of retrievals, potentially from a combination of queries and systems, we measure the correlation of the rank ordering of this set by the predictor and by the performance metric. In order to ensure comparability with previous results, we present Kendall's τ correlation between the predictor's ranking and ranking based on average precision of the retrieval. Unless explicitly noted, all correlations are significant with $p < 0.05$.

Predictors can sometimes perform better when linearly combined [9, 11]. Although previous work has presented the coefficient of determination (R^2) to measure the quality of the regression, this measure cannot be reliably used when comparing slight improvements from combining predictors. Therefore, we adopt the *adjusted* coefficient of determination which penalizes models with more variables. The adjusted R^2 allows us to evaluate the improvement in prediction achieved by adding a parameter but loses the statistical interpretation of R^2. We will use Kendall's τ to evaluate the magnitude of the correlation and the adjusted R^2 to evaluate the combination of variables.

6. RESULTS

We present results for our detailed experiments comparing the prediction of language model scores in Table 1. Although the Clarity measure is theoretically designed for language model scores, it consistently underperforms our system-agnostic predictor. Ranking robustness was presented as an improvement to Clarity for web collections (represented in our experiments by the terabyte04 and terabyte05 collections), shifting the τ correlation from 0.139 to 0.150 for terabyte04 and 0.171 to 0.208 for terabyte05. However, these improvements are slight compared to the performance of autocorrelation on these collections. Our predictor achieves a τ correlation of 0.454 for terabyte04 and 0.383 for terabyte05. Though not always the strongest, autocorrelation achieves correlations competitive with baseline predictors. When examining the performance of linear combinations of predictors, we note that in every case, autocorrelation factors as a necessary component of a strong predictor. We also note that the

adjusted R^2 for individual baselines are always significantly improved by incorporating autocorrelation.

We present our generalizability results in Table 2. We begin by examining the situation in column (a) where we are presented with a single retrieval and no information from additional retrievals. For every collection except one, we achieve significantly better correlations than ranked-list Clarity. Surprisingly, we achieve relatively strong correlations for Spanish and Chinese collections despite our naïve processing. We do not have a ranked-list clarity correlation for ent05 because entity modeling is itself an open research question. However, our autocorrelation measure does not achieve high correlations perhaps because relevance for entity retrieval does not propagate according to the cooccurrence links we use.

As noted above, the poor Clarity performance on web data is consistent with our findings in the detailed experiments. Clarity also notably underperforms for several news corpora (trec5, trec7, and robust04). On the other hand, autocorrelation seems robust to the changes between different corpora.

Next, we turn to the introduction of information from multiple retrievals. We compare the correlations between those predictors which do not use this information in column (a) and those which do in column (b). For every collection, the predictors in column (b) outperform the predictors in column (a), indicating that the information from additional runs can be critical to making good predictions.

Inspecting the predictors in column (b), we only draw weak conclusions. Our new predictors tend to perform better on news corpora. And between our new predictors, the hybrid $\rho(\tilde{\mathbf{y}}, \mathbf{y}_\mu)$ predictor tends to perform better. Recall that our $\rho(\tilde{\mathbf{y}}, \mathbf{y}_\mu)$ measure incorporates both spatial and multiple retrieval information. Therefore, we believe that the improvement in correlation is the result of incorporating information from spatial behavior.

In column (c), we can investigate the utility of incorporating spatial information with information from multiple retrievals. Notice that in the cases where autocorrelation, $\rho(\mathbf{y}, \tilde{\mathbf{y}})$, alone performs well (trec3, trec5-spanish, and trec6-chinese), it is substantially improved by incorporating multiple-retrieval information from $\rho(\mathbf{y}, \mathbf{y}_\mu)$ in the linear regression, β. In the cases where $\rho(\mathbf{y}, \mathbf{y}_\mu)$ performs well, incorporating autocorrelation rarely results in a significant improvement in performance. In fact, in every case where our predictor outperforms the baseline, it includes information from multiple runs.

7. DISCUSSION

The most important result from our experiments involves prediction when no information is available from multiple runs (Tables 1 and 2a). This situation arises often in system design. For example, a system may need to, at retrieval time, assess its performance before deciding to conduct more intensive processing such as pseudo-relevance feedback or interaction. Assuming the presence of multiple retrievals is unrealistic in this case.

We believe that autocorrelation is, like multiple-retrieval algorithms, approximating a good ranking; in this case by diffusing scores. Why is $\tilde{\mathbf{y}}$ a reasonable surrogate? We know that diffusion of scores on the web graph and language model graphs improves performance [14, 16]. Therefore, if score diffusion tends to, in general, improve performance, then diffused scores will, in general, provide a good surrogate for relevance. Our results demonstrate that this approximation is not as powerful as information from multiple retrievals. Nevertheless, in situations where this information is lacking, autocorrelation provides substantial information.

The success of autocorrelation as a predictor may also have roots in the clustering hypothesis. Recall that we regard autocorrelation as the degree to which a retrieval satisfies the clustering hypothesis. Our experiments, then, demonstrate that a failure to respect the clustering hypothesis correlates with poor performance. Why might systems fail to conform to the cluster hypothesis? Query-based information retrieval systems often score documents independently. The score of document a may be computed by examining query term or phrase matches, the document length, and perhaps global collection statistics. Once computed, a system rarely compares the score of a to the score of a topically-related document b. With some exceptions, the correlation of document scores has largely been ignored.

We should make it clear that we have selected tasks where topical autocorrelation is appropriate. There are certainly cases where there is no reason to believe that retrieval scores will have topical autocorrelation. For example, ranked lists which incorporate document novelty should not exhibit spatial autocorrelation; if anything autocorrelation should be *negative* for this task. Similarly, answer candidates in a question-answering task may or may not exhibit autocorrelation; in this case, the semantics of links is questionable too. It is important before applying this measure to confirm that, given the semantics for some link between two retrieved items, we should expect a correlation between scores.

8. RELATED WORK

In this section we draw more general comparisons to other work in performance prediction and spatial data analysis.

There is a growing body of work which attempts to predict the performance of individual retrievals [7, 3, 11, 9, 19]. We have attempted to place our work in the context of much of this work in Section 4. However, a complete comparison is beyond the scope of this paper. We note, though, that our experiments cover a larger and more diverse set of retrievals, collections, and topics than previously examined.

Much previous work—particularly in the context of TREC—focuses on predicting the performance of *systems*. Here, each system generates k retrievals. The task is, given these retrievals, to predict the ranking of systems according to some performance measure. Several papers attempt to address this task under the constraint of few judgments [2, 4]. Some work even attempts to use zero judgments by leveraging multiple retrievals for the same query [17]. Our task differs because we focus on ranking retrievals independent of the generating system. The task here is not to test the hypothesis "system A is superior to system B" but to test the hypothesis "retrieval A is superior to retrieval B".

Autocorrelation manifests itself in many classification tasks. Neville and Jensen define *relational autocorrelation* for relational learning problems and demonstrate that many classification tasks manifest autocorrelation [13]. *Temporal autocorrelation* of initial retrievals has also been used to predict performance [9]. However, temporal autocorrelation is performed by projecting the retrieval function into the temporal embedding space. In our work, we focus on the behavior of the function over the relationships between documents.

	τ			adjusted R^2						
	$D_{KL}^{\mathcal{V}}$	P	$\rho(\mathbf{y},\tilde{\mathbf{y}})$	$D_{KL}^{\mathcal{V}}$	P	$\rho(\mathbf{y},\tilde{\mathbf{y}})$	$D_{KL}^{\mathcal{V}}, P$	$D_{KL}^{\mathcal{V}}, \rho(\mathbf{y},\tilde{\mathbf{y}})$	$P\rho(\mathbf{y},\tilde{\mathbf{y}})$	β
trec4	0.353	**0.548**	0.513	0.168	0.363	0.422	0.466	0.420	**0.557**	0.553
trec5	0.311	0.329	**0.357**	0.116	0.190	0.236	0.238	0.244	0.266	**0.269**
robust04	**0.418**	0.398	0.373	0.256	0.304	0.278	0.403	0.373	0.402	**0.442**
terabyte04	0.139	0.150	**0.454**	0.059	0.045	0.292	0.076	**0.293**	0.289	0.284
terabyte05	0.171	0.208	**0.383**	0.022	0.072	0.193	0.120	0.225	0.218	**0.257**

Table 1: Comparison to Robustness and Clarity measures for language model scores. Evaluation replicates experiments from [19]. We present correlations between the classic Clarity measure ($D_{KL}^{\mathcal{V}}$), the ranking robustness measure (P), and autocorrelation ($\rho(\mathbf{y},\tilde{\mathbf{y}})$) each with mean average precision in terms of Kendall's τ. The adjusted coefficient of determination is presented to measure the effectiveness of combining predictors. Measures in bold represent the strongest correlation for that test/collection pair.

			multiple run							
	(a) τ		(b) τ			(c) adjusted R^2				
	D_{KL}	$\rho(\mathbf{y},\tilde{\mathbf{y}})$	D_{KL}^{n}	$\rho(\mathbf{y},\mathbf{y}_\mu)$	$\rho(\tilde{\mathbf{y}},\mathbf{y}_\mu)$	D_{KL}^{n}	$\rho(\mathbf{y},\tilde{\mathbf{y}})$	$\rho(\mathbf{y},\mathbf{y}_\mu)$	$\rho(\tilde{\mathbf{y}},\mathbf{y}_\mu)$	β
trec3	0.201	**0.461**	**0.461**	0.439	0.456	0.444	0.395	0.394	0.386	**0.498**
trec4	0.252	**0.396**	0.455	0.482	**0.489**	0.379	0.263	0.429	0.482	**0.483**
trec5	0.016	**0.277**	0.433	**0.459**	0.393	0.280	0.157	0.375	0.323	**0.386**
trec6	**0.230**	0.227	0.352	**0.428**	0.418	0.203	0.089	0.323	0.325	**0.325**
trec7	0.083	**0.326**	0.341	0.430	**0.483**	0.264	0.182	0.363	**0.442**	0.400
trec8	0.235	**0.396**	0.454	0.508	**0.567**	0.402	0.272	0.490	**0.580**	0.523
robust03	0.302	**0.354**	0.377	0.385	**0.447**	0.269	0.206	0.274	**0.392**	0.303
robust04	0.183	**0.308**	0.301	0.384	**0.453**	0.200	0.182	0.301	**0.393**	0.335
robust05	0.224	**0.249**	0.371	0.377	**0.404**	**0.341**	0.108	0.313	0.328	0.336
terabyte04	0.043	**0.245**	**0.544**	0.420	0.392	**0.516**	0.105	0.357	0.343	0.365
terabyte05	0.068	**0.306**	**0.480**	0.434	0.390	**0.491**	0.168	0.384	0.309	0.403
trec4-spanish	0.307	**0.388**	**0.488**	0.398	0.395	**0.423**	0.299	0.282	0.299	0.388
trec5-spanish	0.220	**0.458**	0.446	**0.484**	0.475	0.411	0.398	0.428	0.437	**0.529**
trec5-chinese	0.092	**0.199**	0.367	0.379	**0.384**	**0.379**	0.199	0.273	0.276	0.310
trec6-chinese	0.144	**0.276**	0.265	0.353	**0.376**	0.115	0.128	0.188	**0.223**	0.199
ent05	-	0.181	**0.324**	0.305	0.282	**0.211**	0.043	0.158	0.155	0.179

Table 2: Large scale prediction experiments. We predict the ranking of large sets of retrievals for various collections and retrieval systems. Kendall's τ correlations are computed between the predicted ranking and a ranking based on the retrieval's average precision. In column (a), we have predictors which do not use information from other retrievals for the same query. In columns (b) and (c) we present performance for predictors which incorporate information from multiple retrievals. The adjusted coefficient of determination is computed to determine effectiveness of combining predictors. Measures in bold represent the strongest correlation for that test/collection pair.

Finally, regularization-based re-ranking processes are also closely-related to our work [8]. These techniques seek to maximize the agreement between scores of related documents by solving a constrained optimization problem. The maximization of consistency is equivalent to maximizing the Moran autocorrelation. Therefore, we believe that our work provides explanation for why regularization-based re-ranking works.

9. CONCLUSION

We have presented a new method for predicting the performance of a retrieval ranking without any relevance judgments. We consider two cases. First, when making predictions in the absence of retrievals from other systems, our predictors demonstrate robust, strong correlations with average precision. This performance, combined with a simple implementation, makes our predictors, in particular, very attractive. We have demonstrated this improvement for many, diverse settings. To our knowledge, this is the first large scale examination of zero-judgment, single-retrieval performance prediction. Second, when provided retrievals from other systems, our extended methods demonstrate competitive performance with state of the art baselines. Our experiments also demonstrate the limits of the usefulness of our predictors when information from multiple runs is provided.

Our results suggest two conclusions. First, our results could affect retrieval algorithm design. Retrieval algorithms designed to consider spatial autocorrelation will conform to the cluster hypothesis and improve performance. Second, our results could affect the design of minimal test collection algorithms. Much of the recent work in ranking systems sometimes ignores correlations between document labels and scores. We believe that these two directions could be rewarding given the theoretical and experimental evidence in this paper.

10. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor. We thank Yun Zhou and Desislava Petkova for providing data and Andre Gauthier for technical assistance.

11. REFERENCES

[1] J. Aslam and V. Pavlu. Query hardness estimation using jensen-shannon divergence among multiple scoring functions. In *ECIR 2007: Proceedings of the 29th European Conference on Information Retrieval*, 2007.

[2] J. A. Aslam, V. Pavlu, and E. Yilmaz. A statistical method for system evaluation using incomplete judgments. In S. Dumais, E. N. Efthimiadis, D. Hawking, and K. Jarvelin, editors, *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 541–548. ACM Press, August 2006.

[3] D. Carmel, E. Yom-Tov, A. Darlow, and D. Pelleg. What makes a query difficult? In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 390–397, New York, NY, USA, 2006. ACM Press.

[4] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 268–275, New York, NY, USA, 2006. ACM Press.

[5] A. D. Cliff and J. K. Ord. *Spatial Autocorrelation*. Pion Ltd., 1973.

[6] M. Connell, A. Feng, G. Kumaran, H. Raghavan, C. Shah, and J. Allan. Umass at tdt 2004. Technical Report CIIR Technical Report IR – 357, Department of Computer Science, University of Massachusetts, 2004.

[7] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Precision prediction based on ranked list coherence. *Inf. Retr.*, 9(6):723–755, 2006.

[8] F. Diaz. Regularizing ad hoc retrieval scores. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 672–679, New York, NY, USA, 2005. ACM Press.

[9] F. Diaz and R. Jones. Using temporal profiles of queries for precision prediction. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 18–24, New York, NY, USA, 2004. ACM Press.

[10] D. A. Griffith. *Spatial Autocorrelation and Spatial Filtering*. Springer Verlag, 2003.

[11] B. He and I. Ounis. Inferring Query Performance Using Pre-retrieval Predictors. In *The Eleventh Symposium on String Processing and Information Retrieval (SPIRE)*, 2004.

[12] N. Jardine and C. J. V. Rijsbergen. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7:217–240, 1971.

[13] D. Jensen and J. Neville. Linkage and autocorrelation cause feature selection bias in relational learning. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 259–266, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[14] O. Kurland and L. Lee. Corpus structure, language models, and ad hoc information retrieval. In *SIGIR '04: Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 194–201, New York, NY, USA, 2004. ACM Press.

[15] M. Montague and J. A. Aslam. Relevance score normalization for metasearch. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 427–433, New York, NY, USA, 2001. ACM Press.

[16] T. Qin, T.-Y. Liu, X.-D. Zhang, Z. Chen, and W.-Y. Ma. A study of relevance propagation for web search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 408–415, New York, NY, USA, 2005. ACM Press.

[17] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 66–73, New York, NY, USA, 2001. ACM Press.

[18] V. Vinay, I. J. Cox, N. Milic-Frayling, and K. Wood. On ranking the effectiveness of searches. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 398–404, New York, NY, USA, 2006. ACM Press.

[19] Y. Zhou and W. B. Croft. Ranking robustness: a novel framework to predict query performance. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 567–574, New York, NY, USA, 2006. ACM Press.