Consistent Phrase Relevance Measures

Wen-tau Yih
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
scottyih@microsoft.com

Christopher Meek
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
meek@microsoft.com

ABSTRACT

Measuring the relevance between a document and a phrase is fundamental to many information retrieval and matching tasks including on-line advertising. In this paper, we explore two approaches for measuring the relevance between a document and a phrase aiming to provide consistent relevance scores for both in and out-of document phrases. The first approach is a similarity-based method which represents both the document and phrase as term vectors to derive a real-valued relevance score. The second approach takes as input the relevance estimates of some in-document phrases and uses Gaussian Process Regression to predict the score of a target out-of-document phrase. While both of these two approaches work well, the best result is given by a Gaussian Process Regression model, which is significantly better than the similarity-based approach and 10% better than a baseline similarity method using bag-of-word vectors.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Abstracting methods; H.4.m [**Information Systems**]: Miscellaneous

General Terms

Algorithms, experimentation

Keywords

keyword extraction, relevance measures, online advertising, Gaussian process regression

1. INTRODUCTION

The problem of estimating the relevance between a phrase and a document is a well-studied problem in the information retrieval community where the typical goal is to rank documents by their relevance to a query. In this paper we consider the "inverse" problem of query ranking where, instead of ranking documents, our goal is to provide *consistent* relevance measures for both in and out-of document phrases and to rank phrases by their relevance to a document.

To motivate the need for consistent measures for phrase–document relevance we briefly review two applications that benefit from extending an in-document relevance measure to handle out-of-document phrases. Both applications we consider in this paper are related to online advertising where the selection of ads to be shown is primarily driven by phrases (bid keywords) selected by an advertiser. For a review of on-line advertising see [2].

In contextual advertising, phrases in a web page may be identified as relevant keywords by a keyword extraction algorithm, and used to select an appropriate contextual ad [22]. This approach, however, limits the potential matching bid keywords to phrases that appear in the document. By extending a relevance measure from an in-document relevance measure we can broaden the range of potentially relevant ads for any particular context. Proposals for alternative phrases might come from different sources including from query suggestion systems [7] and from the content publisher. For example, an online sports magazine may want the advertising platform to consider showing ads for generic keywords such as *MLB* or *NFL* on all its pages, regardless of whether the added keyword occurs in the document or not. Another application of such measures in the domain of sponsored search advertising is automated relevance verification of bid keywords. In this application, one uses the measures to verify the relevance of keywords to an ad landing page when an advertiser bids on a set of keywords. By excluding some unrelated keywords one can enhance the search user experience by reducing the number of irrelevant ads.

Part of the challenge of the query ranking problem is to provide a *consistent* relevance measures for both in and out-of document phrases. Naively using documents as queries and applying typical relevance ranking functions such as cosine with TFIDF weighting would lead to a poor measure. For example, if a document contains the phrase "Major League Baseball" but not "MLB", these phrases would have very different scores even if they are synonymous. We propose two approaches for solving this problem: one similarity-based and the other regression-based. The first approach extends recent work on measuring similarity between two short-text segments to measure relevance between a phrase and a document. The technique can be thought of as a query expansion technique [21] based on pseudo-relevance feedback [20] using the World Wide Web as the external data source. In contrast to traditional query expansion, however, our goal is to use query expansion to identify a set

of related words to represent the semantics of the "query" phrase. Ideally, two synonyms such as "MLB" and "Major League Baseball" will be represented by very similar vectors and thus have similar relevance scores when compared against the same document.

One potential drawback of the similarity-based approach is that document-specific information, such as whether the phrase appears in the anchor text or whether the phrase occurs in the title cannot be exploited to derive a more accurate relevance measure for in-document phrases. This potential weakness is the motivation for our second approach where we use a regression model to leverage the results of an accurate in-document phrase relevance measure to predict out-of-document phrase relevance. Conceptually, this approach first uses an in-document phrase relevance module to judge the relevance scores for some in-document phrases. We then use a measure of similarity between in and out-of document phrases to predict the relevance of an out-of-document phrase. For instance, if our in-document phrase is "Major League Baseball" and our similarity function indicates that the out-of-document phrase "MLB" is synonymous, then the predicted relevance scores for the two phrases would be the same. In our experiments, the similarity between the in and out-of document phrases is based on the query expansion similarity technique mentioned above and the regression technique is Gaussian Process Regression (GPR).

We evaluate our approaches in an online advertising scenario where we judge the relevance scores of keywords to an ad landing page. The quality of our relevance functions are assessed using several metrics: AUC, precision at k, accuracy, F_1 and cross-entropy. We found that both approaches work well in our experiments, but the best result is given by the regression approach, which performs significantly better than the similarity approach and 10-12% better in all metrics than the baseline approach that uses a bag-of-words scheme to measure relevance.

The rest of this paper is structured as follows. Sec. 2 gives some background of the fundamental techniques used by our approaches, including measuring in-document phrase relevance using keyword extraction and query expansion using the Web as the document source. The two main approaches, the similarity-based and regression-based methods are introduced in Sec. 3 and 4, respectively. Sec. 5 presents the experimental evaluation and some analysis of the results. Finally, we introduce other related work in Sec. 6 and conclude the paper in Sec. 7.

2. BACKGROUND

Given a phrase ph (a short sequence of one or more words) and a document d, we would like a real-valued measure of their relevance. For some applications that require probabilistic relevance measures, we would like to specifically estimate the probability that an annotator would label the phrase ph as relevant to the document d. We propose two approaches to this problem: similarity-based and regression-based methods. The similarity-based approach first represents the target phrase as a term vector via query expansion using the Web as the external document source. A real-valued similarity score is then assigned by comparing this vector with the term vector derived from the target document. When the probability of being relevant is preferred, the estimation is done by calibrating the raw similarity score through a learned sigmoid function. The regression-based approach, on the other hand, separates the in and out-of document phrases. It follows the intuition that the relevance of an in-document phrase can be better estimated directly by a separate component that uses document-specific information that relates to the phrase. For an out-of-document phrase, if it is *similar* to an in-document phrase, the relevance scores of these two phrases should be close to each other.

Both our approaches rely on three underlying techniques – measuring in-document phrase relevance, projecting documents into a vector space and expanding a phrase to a pseudo-document via query expansion. We briefly describe these techniques in this section.

2.1 Measure In-document Phrase Relevance

For a phrase ph that occurs *in* the document d, an in-document phrase relevance measure maps them to a real number and can be denoted as $Rel(ph, d) \rightarrow \mathcal{R}$. This problem is a well-studied problem in information retrieval. For example, when searching for the relevant documents given a query, typically documents that contain the query phrase are selected and ranked based on how relevant they are to the (in-document) query. Standard approaches from the information retrieval community include the basic TFIDF formulas and the BM25 [16, 17] ranking function. Another application of the in-document phrase relevance measure is keyword extraction, which is a fundamental technology for contextual advertising such as Google's AdSense, Yahoo's Contextual Match and MSN AdCenter's Content Ads. *Keyword extraction* (KEX) [6, 22] takes as input a target document and outputs short phrases in the document that are relevant to the content as the selected keywords. The output keywords can then be used to match bid keywords to show relevant ads on the target web page. Although a keyword extraction system usually only outputs a short list of keywords, internally KEX considers all short phrases in the target document as candidates and evaluates their relevance, often based on some probabilistic models such as naive Bayes [6] and logistic regression [22].

In this work, we experiment with several different in-document phrase relevance measures, including using keyword extraction ($Rel_{\texttt{KEX}}$), a TFIDF formula ($Rel_{\texttt{TFIDF}}$) and a uniform weighting ($Rel_{\texttt{BIN}}$). For $Rel_{\texttt{KEX}}$, we built a keyword extraction component following the approach described in [22], which is a state-of-the-art keyword extraction system trained using more than 10 categories of features. Among them, the three most important features are: term frequency (TF), document frequency (DF) and search query log. TF can be treated as a document-dependent feature which provides a rough estimate of on how important the target phrase is, relative to the whole document. DF and search query log can be viewed as document-independent features. The former downweights stopwords and common phrases in the document collection and the latter upweights phrases that are frequently queried. Because this model is trained using logistic regression, its output can be used as probability directly. For $Rel_{\texttt{TFIDF}}$, the exact TFIDF formula we use is the following:

$$w_i = tf_i \times \log(N/df_i), \tag{1}$$

where N is the total number of documents when counting document frequency. Finally, $Rel_{\texttt{BIN}}$ simply assigns 1 as the relevance score of an in-document phrase. This simple strategy is basically used as a baseline approach.

2.2 Project Documents into Vectors

In our approaches, both documents and phrases are represented in vectors for further processing. We map a document d to a sparse term vector, where each term is a word, associated with a weight that indicates the relevance of this word to d. The term-weighting function used for this vector projection process is one of the in-document phrase relevance measures described in Sec. 2.1, followed by L_2 normalization. The detailed steps are:

1. Construct the term vector $\mathbf{v}$, where each element is w_j (the j-th word in the document d), with a real-valued weight $Rel(w_j, d)$.

2. Apply L_2 normalization on $\mathbf{v}$. That is, the final output vector, $Vec(d)$, is $\mathbf{v}/||\mathbf{v}||$.

We denote the projection function according to the term-weighting function. For example, $Vec_{\texttt{KEX}}$, $Vec_{\texttt{TFIDF}}$ and $Vec_{\texttt{BIN}}$ mean that the in-document phrase relevance functions, $Rel_{\texttt{KEX}}$, $Rel_{\texttt{TFIDF}}$ and $Rel_{\texttt{BIN}}$ are used for term-weighting, respectively.

When mapping a phrase to a vector, a query expansion technique, described in the next section, is first used to construct a pseudo-document, before applying the above vector projection method.

2.3 Query Expansion

When measuring the similarity between a phrase and a document (used in the similarity-based methods) or measuring the similarity between two phrases (used in the regression-based methods), the major difficulty is that there can be very little orthographic similarity between the phrases. As reported in the previous work [18, 10], naively relying on the co-occurrence of the words in the target phrases leads to unreliable and low-quality similarity measures. Because of this difficulty, one important step when handling the input phrase in our approaches is to represent it as a *pseudo-document* using *query expansion* [13, 21]. Generally speaking, query expansion is a procedure that treats the target phrase as a search query and represents it as a set of semantically related words. In this work, we use the Web as the document source and follow the idea of pseudo-relevance feedback [20] for this expansion. We denote the process of mapping a phrase to a pseudo-document as $WebQE$, which consists of the following steps.

1. Let $D_n(ph)$ be the set of top n documents returned by a search engine when using phrase ph as the query.

2. Construct a pseudo-document $WebQE(ph)$ by concatenating the title and short summary of each document $d_i \in D_n(ph)$.

In other words, we treat the top n search results of the titles and summaries as relevant text to the phrase. In the experiments, we set n to 50.

3. SIMILARITY-BASED APPROACHES

The similarity-based approach of measuring the relevance of a phrase to a document can be viewed as an extension of recent work of similarity measures between short text segments [18, 23]. In this approach, both the phrase and document are represented as non-negative vectors and their cosine score is used as the raw relevance score. When a probabilistic relevance measure is preferred, we learn a parameterized sigmoid function using held-out data and map the raw score to a probability. The sigmoid function can also be used as a tool to combine multiple relevance scores and yield not only a better probability estimation but also superior ranking results. We next describe these steps in more detail.

3.1 Measure Phrase Document Similarity

Although recently proposed similarity measures of short text segments, such as [18, 23], are designed for comparing two short text segments or phrases, their ideas can be naturally extended to measure the similarity between a phrase and a document, which we used as the relevance measure. When measuring the similarity between two phrases ph_1 and ph_2, these methods first apply query expansion and a term-weighting function to represent the input phrases as two vectors. Their inner-product is simply output as the similarity score. Using our notations, the similarity score can be formulated as $Vec(WebQE(ph_1)) \cdot Vec(WebQE(ph_2))$. When comparing a phrase ph and a document d, we can follow the same strategy and use the inner-product, $Vec(WebQE(ph)) \cdot Vec(d)$, as the relevance score.

As mentioned briefly in Sec. 2.2, we experiment with three different term-weighting functions.

1. $\mathbf{SimBin}(ph, d) = Vec_{\texttt{BIN}}(WebQE(ph)) \cdot Vec_{\texttt{BIN}}(d)$
   All the words in the document are weighted equally. So are the words in the pseudo-document of the phrase.

2. $\mathbf{SimTFIDF}(ph, d) = Vec_{\texttt{TFIDF}}(WebQE(ph)) \cdot Vec_{\texttt{TFIDF}}(d)$
   The term-weighting function is the TFIDF formula in Eq. 1.

3. $\mathbf{SimKEX}(ph, d) = Vec_{\texttt{KEX}}(WebQE(ph)) \cdot Vec_{\texttt{KEX}}(d)$
   The term-weighting function is KEX.

Among them, SimBin is the simplest and the inner-product can be reduced to set operations. As a comparison, SimKEX is the most complicated, but potentially has a better term-weighting function because information other than TF and DF is also used.

3.2 Map Relevance Scores to Probabilities

Although the similarity measures can be used as ranking functions to judge the relevance between the target phrase and document, these real-valued numbers, despite being between 0 and 1, are poorly calibrated and cannot be used as probabilities when they are needed. Calibrating the predictions of a non-probabilistic model such as SVMs or boosted trees has been studied extensively in the machine learning community [11]. One of the most popular methods is proposed by Platt, where he advocates using a sigmoid function to map the real-valued output f of the model to posterior probabilities [12]:

$$P(y = 1|f) = \frac{1}{1 + \exp(\alpha f + \beta)}, \tag{2}$$

where α and β are parameters tuned using the maximum likelihood estimation from a separate training set. Platt's scaling was originally designed for SVMs where f is the margin and plays a similar role of the log odds. We use the same function of log odds to map the raw score $s \in [0, 1]$ to f when

applying this monotonic transformation[1]:

$$f = log(s)/log(1-s) \quad (3)$$

3.3 Combine Multiple Relevance Scores

Because the sigmoid function (Eq. 2) is a monotonic function, using it to map the original relevance score to probability does not change the ordering. Although the quality of the probability estimation will be improved, other ranking metrics such as precision or AUC will remain the same. However, this functional form can also be used as tool to combine multiple similarity-based methods (e.g., SimBin, SimTFIDF and SimKex) and improve the ranking as well.

Suppose we have m relevance scores, $s_1, s_2, \cdots, s_m$, output by different similarity-based methods. Let $f_1, f_2, \cdots, f_m$ be the values after mapping these scores using Eq. 3. Namely, $f_i = \log(s_i)/\log(1-s_i)$. The probability is computed using a similar sigmoid function:

$$P(y=1|f_1, f_2, \cdots, f_m) = \frac{1}{1+\exp(\sum_i^m \alpha_i f_i + \beta)}, \quad (4)$$

where α_i and β are the parameters to learn.

Because the parameter space is quite small, standard gradient descent methods can be easily used as the learning method. We use this method to combine the three similarity-based methods described in Sec. 3.1 in the experiments and denote it as **SimCombine**. In Section 5, we will give more detail on how we constructed a training set for our evaluations of these combination and calibration methods.

4. THE REGRESSION APPROACHES

The *regression*-based approaches follow the intuition that if an out-of-document phrase is semantically similar to an in-document phrase, then the relevance scores of these two phrases should be close to each other. In this approach, we first apply an in-document phrase relevance measure, such as KEX, on the target document to get a list of top N keywords, associated with the relevance scores. The second step is to judge whether an out-of-document phrase is similar to these top in-document phrases and predict the scores that are consistent. A principled way to do this is through regression. In this setting, each phrase ph_i extracted by the in-document relevance measure is represented by a sparse term vector via query expansion $\mathbf{x}_i = Vec(WebQE(ph_i))$, as well as the target phrase, denoted by ph_{N+1} and $\mathbf{x}_{N+1} = Vec(WebQE(ph_{N+1}))$. Correspondingly, the relevance scores of these in-document phrases are denoted as $y_1, \cdots, y_N$. Given the N pairs of $(y_i, \mathbf{x}_i)$ derived from the N in-document phrases and their scores, our goal is to predict y_{N+1}, the relevance score of the target phrase ph_{N+1}. We use Gaussian Process Regression as the regression model, which enjoys several advantages that make it particularly suitable to this task. In this section, we first give a short introduction to Gaussian Process Regression and then describe how we use it to solve our problem.

4.1 Gaussian Process Regression

Gaussian Process Regression (GPR) [9, 14] is a nonparametric model that uses a Gaussian Process (GP) as the prior probability distribution over a function space. A GP is a stochastic process $y(\mathbf{x})$ over a multi-dimensional input space $\mathbf{x}$ that has the following defining property: for any finite selection of points $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$, the corresponding marginal density $P(y(\mathbf{x}_1), \cdots, y(\mathbf{x}_N))$ is a (multi-variate) Gaussian. A Gaussian Process (GP) is fully described by two statistics: the mean $\mu(\mathbf{x})$ and the covariance (i.e., kernel) function on each pair of examples $K(\mathbf{x}, \mathbf{x}')$. Because a random phrase that does not appear in the document tends to be irrelevant to the document, we therefore assume the GP over the relevance function has zero mean.

To use GPR is fairly straightforward: we only need to specify the kernel function and the Gaussian noise term. Given N examples and their observed values $(y_1, \mathbf{x}_1)$, $\cdots$, $(y_N, \mathbf{x}_N)$, and the testing example $\mathbf{x}_{N+1}$, the predicted mean value for y_{N+1} is

$$y_{N+1} = \mathbf{k}^{\mathbf{T}}(\mathbf{K} + \sigma_n^2 \mathrm{I})^{-1}\mathbf{y},$$

where $\mathbf{k}$ is the vector of covariances (given by the specified kernel function $K(\mathbf{x}_{N+1}, \mathbf{x}_i)$) between the test example $\mathbf{x}_{N+1}$ and the N training examples, $\mathbf{K}$ is the N-by-N covariance matrix, where each element (i, j) is $K(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{y}$ is the vector of N observed values $y_1, y_2, \cdots, y_N$, and σ_n^2 is the variance of the Gaussian noise. The computational complexity of solving this equation is $O(N^3)$ for the matrix inversion.

GPR has several advantages that are particularly suitable to our task. For example, although typically a document can have thousands of short phrases, most of them are not related to the topic of the document. As a result, there are typically fewer than 20 phrases (i.e., $N < 20$) with probabilities higher than a threshold that can be chosen as meaningful keywords. For problems of this size, GPR has been shown empirically effective and the inference problem, despite being cubic in the number of input examples (i.e., number of selected in-document keywords), is of modest computational cost, which makes it an appropriate model for our task. GPR is also a very general regression model that subsumes other natural families of regression models. For instance, it can be shown that GPR is equivalent to Bayesian linear regression when a linear kernel function is used.

4.2 Kernel Functions

There are a huge variety of choices of kernel functions. In this paper we use the three most common ones: *linear kernel*, *polynomial kernel* and *radial basis kernel*.

Given two vectors $\mathbf{x}$ and $\mathbf{x}'$, the linear kernel function is simply the inner product of these vectors plus a bias term:

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{x} \cdot \mathbf{x}' + \sigma_0^2$$

For simplicity, we set σ_0^2 to 0 in the experiments, which makes this kernel function a homogeneous linear kernel. In fact, short-text similarity measures that use the inner-product of two vectors [18, 23] are also homogeneous linear kernels.

As a straightforward extension, the polynomial kernel is a positive-integer power of linear kernel:

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}' + \sigma_0^2)^p,$$

where p is a positive-integer. It is well known that a polynomial kernel function maps the original vector into a higher dimensional space, where the parameter p decides the degree. Since polynomial kernels have proved quite effective in high-dimensional classification problems [19], we also in-

[1]When s is 0 or 1, we use ϵ and $1-\epsilon$ respectively, where ϵ is a very small number, to avoid the numerical problem.

clude this kernel function in the experiments and set the bias term σ_0^2 to 0.

Unlike linear and polynomial kernels, a radial basis kernel function (RBF) is an exponential function that takes the difference of the two input vectors with some scaling.

$$K(\mathbf{x}, \mathbf{x}') = \exp(-|\mathbf{x} - \mathbf{x}'|^2/\sigma^2)$$

An RBF kernel has the effect of mapping the original vector into an infinitely high dimensional space. We use this kernel with several different scaling parameters.

5. EXPERIMENTS

In this section, we present the experimental evaluation of our phrase–document relevance measures. We describe our evaluation data, the evaluation metrics used, and the results of the similarity-based and regression-based methods using various kernels and term-weighting functions.

5.1 Data

The evaluation dataset was constructed using log files for a commercial search engine from a three month period in 2007. We randomly selected the landing pages of clicked ads from the logs along with the user queries that lead to the ad clicks. After removing duplicated and non-English pages, we retained 867 pages in our document set. For each document query pair, we use a set of existing query suggestion algorithms to generate alternative keywords. The alternative queries vary in relevance to the original document with some of them quite similar and others only just loosely related. After this process, we had 10 to 13 candidate keywords for each of the documents yielding a dataset consisting of 9,319 keyword–document pairs. We then manually judged whether a keyword is relevant to its corresponding document labeling 4,381 (47.0%) pairs as *relevant* and 4,938 (53.0%) pairs as *irrelevant*. Most of the keywords (81.9%) were out-of-document keywords.

5.2 Evaluation metrics

The quality of a relevance measure is usually hard to evaluate using one simple metric since quality is typically a function of the specific application. Therefore, we use five different metrics for evaluation: *AUC*, *precision at k*, *accuracy*, F_1 and *cross-entropy*. The first two consider the situation when a real-valued measure (probability or not) is used as a ranking function. Accuracy and the F_1 score evaluate the relevance measure when it is used as a binary classifier. Finally, when the relevance measure is a probability measure, cross-entropy is a suitable metric to evaluate its quality.

A standard method for evaluating ranking and classification methods is to use receiver operating characteristic (ROC) curves. These curves plot true-positive rate versus false-positive rate at various thresholds for each method and allow alternative methods to be compared at varying levels of false positives. Due to space limitations, we do not present ROC curves but rather present results for the area under the ROC curve (AUC), a standard real-valued summary of the ROC curve. The AUC score is equivalent to the accuracy of the method's ability to predict the relative relevance for each pair of the phrases, and can be derived using the Wilcoxon-Mann-Whitney statistic [3].

Another common metric for a ranking scenario is the *precision at k*, which calculates the accuracy of the top-ranked k elements. Unlike the AUC score, which treats each pair of phrases as equally important, the precision at k metric only measures the quality of the top ranked items and ignores the rest. We set k to 3 when using this metric in the experiments.

Table 1: Evaluation results of similarity-based relevance measures

	AUC	Prec@3	Acc.	F_1	Entropy
SimBin	0.702	0.704	0.651	0.620	0.939
SimTFIDF	**0.726**	**0.769**	0.663	0.636	**0.887**
SimKEX	**0.726**	**0.727**	0.654	**0.642**	**0.882**
SimCombine	**0.752**†	**0.774**	**0.681**†	**0.665**†	**0.864**†

Compared to the above ranking-related evaluation metrics, the overall prediction accuracy is a simple and direct way to judge a relevance measure when it is used for classifying whether a pair of phrase and document is relevant. We calculate the accuracy of a probabilistic relevance measure when the decision threshold is 0.5. This is to simulate the scenario in which the relevance measure is used for a new dataset and the best decision point cannot be tuned in advance. Another commonly used evaluation metric in the information retrieval community is the F_1 score, which is the harmonic mean of precision and recall. As with accuracy, we choose to calculate the F_1 score at probability 0.5 rather than tuning the decision threshold.

Finally, when the relevance measure is a probability measure, *cross-entropy* is a natural and standard metric to evaluate the quality of the probability estimate. For a pair of document d and phrase ph, if $p(ph|d)$ represents the estimated probability that an annotator labels ph as relevant to d, the cross-entropy of such prediction is

$$\begin{array}{ll} -\log_2 P(ph|d) & \textit{if } ph \textit{ is labeled relevant to } d \\ -\log_2(1 - P(ph|d)) & \textit{if } ph \textit{ is labeled irrelevant to } d \end{array}$$

The ideal cross-entropy is 0 and a lower cross-entropy means a better probability estimate. When comparing the probability estimates from different methods, we report the averaged cross-entropy.

5.3 Cross-Validation

In order to measure the statistical significance of performance differences between methods we use cross-validation to generate ten sets of results for each metric. Documents are randomly split into ten disjoint subsets. Each of the subsets is treated as a test set. The similarity-based approach, which requires training data to fit the sigmoid calibration function, uses the nine remaining subsets for that purpose. The results presented are the average of the 10 results. Statistical significance is determined by using a student's paired-t test on these individual scores and the result is considered to be statistically significant when the p-value is lower than 0.05.

5.4 Similarity-based Approaches

The results for the similarity-based methods on each of the different metrics is presented in Table 1. For all the metrics except cross-entropy, higher numbers mean better results. Results are presented in boldface if the method improves performance for the highlighted metric relative to SimBin. A SimCombined result is annotated with a † if the SimCombined method improves performance for the highlighted

metric relative to all other similarity-based methods.

Not surprisingly, SimBin, which constructs the term vectors based on a simple bag-of-words strategy, does not perform as well as other methods. In fact, as we can see in Table 1, SimBin is inferior to all other methods in all the evaluation metrics, except for accuracy and the F_1 score. On the other hand, SimTFIDF and SimKEX appear to perform equally well. SimTFIDF is higher in precision at 3 and accuracy while SimKEX has better F_1 and cross-entropy, and both methods have the same AUC scores. However, except for precision at 3, none of these differences are statistically significant. As a result, using KEX for weighting the terms when constructing the vectors does not seem to have clear advantages in this method. Finally, the combined approach performs the best compared to all other individual methods. Except for precision at 3 when compared with SimTFIDF, SimCombine is statistically significantly better than all other methods. This indicates that the individual methods behave differently and are complementary to each other when combined together.

5.5 Regression-based Approaches

For the regression-base methods, there are three components that control the quality of the final result – the in-document phrase relevance measure, the query expansion method and the kernel function.

Given a document d and a phrase ph, if ph is in the document, then its relevance score is measured by the in-document phrase relevance measure and will not be changed by the regression model. If ph is out of the document d, then a small set of in-document phrases and their relevance scores are extracted and their in-document phrase relevance scores are measured and used as the training data for regression. In both cases, the quality of the in-document phrase relevance measure influences the quality of the final results.

Due to its method of training, the KEX algorithm is tuned for predicting the relevance of only the most relevant phrases, which can be problematic for the regression approach. To alleviate this problem we reevaluate the relevance of KEX-extracted keywords using SimKEX to improve the relevance estimates for a larger set of keywords. The results presented here use this approach which we found improves over simply using KEX.

The second component that affects the regression approach is how we transform the original phrase into a vector representation; namely, the query expansion method. We experiment with three different term-weighting functions for query expansion in this set of experiments: *Binary*, *TFIDF* and *KEX*. This is analogous to the three similarity-based methods (i.e., SimBin, SimTFIDF and SimKEX) that we tested previously.

Finally, the choice of the kernel function can impact the results. We experiment with six kernel functions: linear kernel (Linear), polynomial kernel with degree 2 and 3 (Poly-2 and Poly-3), and RBF kernel with scaling factors 0.5, 1 and 2 (RBF-0.5, RBF-1 and RBF-2). In all the regression experiments, the variance of Gaussian noise, σ_n^2, is set to 0.1, suggested by early experiments although the regression result is not sensitive to this parameter.

We first show the results of using different kernel functions in the Gaussian Process Regression model. We found that the relative performance difference between different kernels is roughly consistent across different term-weighting functions in query expansion. Thus, to simplify the presentation, we present results using TFIDF as the term-weighting function for different kernels.

Table 2: Results of regression-based relevance measures using different kernels (TFIDF term-weighting)

	AUC	Prec@3	Acc.	F_1	Entropy
Linear	0.766	**0.780**	0.694	0.679	0.846
Poly-2	**0.773**	**0.780**	0.697	**0.696**	**0.845**
Poly-3	**0.774**	**0.778**	0.693	**0.701**	**0.852**
RBF-0.5	**0.773**	**0.779**	**0.704**	0.680	**0.835**
RBF-1	0.756	**0.775**	0.691	0.655	0.864
RBF-2	0.733	**0.766**	0.675	0.632	0.895

Table 3: Results of regression-based relevance measures using different term-weighting functions in query expansion (RBF-0.5 kernel)

	AUC	Prec@3	Acc.	F_1	Entropy
SimCombine	0.752	0.774	0.681	0.665	0.864
Bin	0.725	0.778	0.681	0.610	0.886
TFIDF	**0.773**	0.779	**0.704**	0.680	**0.835**
KEX	**0.772**	0.766	**0.706**	**0.681**	**0.836**

Table 2 shows the results for all the metrics. The numbers in boldface for each column are the best group in the corresponding metric. The difference between methods in and outside the group is statistically significant, but the differences among the methods in the group are statistically insignificant. The statistical significance test is conducted in the same way as described in Sec. 5.3. It is difficult to conclude which kernel function is the best since the answer depends on the exact evaluation metric. However, the general trend seems to indicate that for RGF kernels, lower scaling factor leads to better performance, and higher order polynomial kernels are better than the linear kernel. As a result, RGF-0.5, Poly-2 and Poly-3 are the best configurations and perform comparably.

When fixing the kernel function as RBF-0.5 and changing the term-weighting function in query expansion, TFIDF and KEX again perform equally well and the differences are statistically insignificant. Using the binary term-weighting scheme performs the worst and the differences when compared to the other two methods are all statistically significant, except for precision at 3.

To make it easy to compare the regression-based methods with the similarity-based methods, we copy the row of SimCombine in Table 1 and put it in Table 3. When the result is statistically significantly better than SimCombine, we put the number in boldface. As shown in the table, both TFIDF and KEX term-weighting with RBF-0.5 kernel perform better than the best similarity-based approach in most of the evaluation metrics.

5.6 Example

To illustrate that the in and out-of document phrases are assigned consistent scores, we present a typical example of the predicted relevance scores for a set of keywords for a page. This example is shown in Figure 1. These results are for an ad landing page that promotes the credit report ser-

in-doc?	*keyword*	*score*
✓	truecredit	0.879
	credit bureau report	0.749
✓	transunion	0.705
	credit report services	0.704
	equifax credit bureau	0.652
	equifax credit report	0.649
✓	credit bureaus	0.637
	exquifax	0.516
	equfax	0.498
	trans union canada	0.469
✓	id theft	0.138

Figure 1: Keywords for a credit report company; the relevance scores are predicted by the regression method with TFIDF term-weighting and RBF-0.5 kernel

vice of a company. The keywords are scored and ranked using the regression method with TFIDF term-weighting and RBF-0.5 kernel. Only four of the keywords occur in the document and the system arguably produces reasonable scores for both in and out-of document keywords with several of the out-of-document keywords getting scored above less relevant in-document keywords. In addition, closely related keywords seem to have similar scores. For instance, "exquifax" and "equfax" are both misspelled words of a big credit report company (Equifax), and their scores are both near 0.5.

6. RELATED WORK

Measuring the in-document phrase relevance is one of the most important problems in information retrieval, where the goal is to rank documents with respect to a query. Traditional ranking functions combine the term frequencies and inverse document frequencies of the words in the query (e.g., BM25 [16]). State-of-the-art systems incorporate additional information about the document structure such as the multiple weighted fields as in BM25F [17]. In a similar fashion, the keyword extraction system [22] used in this paper, judges phrase relevance using extended document features related to the phrase in question.

In both cases, the phrase relevance measures rely on document-dependent information such as term-frequency, which makes it difficult to judge out-of-document phrases. One approach to this problem is query expansion [13, 21, 8, 24]. Generally speaking, query expansion is a procedure that extends the original query by adding words that are related to it. Often query expansion techniques provide a weighted collection of words, where the weight reflects the relevance of the new word to the old query. One popular method of finding related words is through the process of pseudo-relevance feedback [20], which assumes that words in the top ranking documents are relevant to the query and words in other documents are not.

Our similarity-based methods extend recent work of measuring similarity between short text segments, which implements a variation of query expansion and pseudo-relevance feedback using the Web as the document source [18, 10, 23]. As discussed earlier, the main difference between our approach and their work is that we compare phrases and documents and also suggest a sigmoid transformation for mapping the raw score to a probability of relevance. As a comparison, using a regression model to make the out-of-document phrase relevance measure consistent with the existing in-document phrase relevance scores, to the best of our knowledge, is novel.

An alternative to the above corpus-based methods is to directly use click-through logs of a search engine to assess relevance. When users click on a link returned by the search engine, it can be viewed as a vote that the query term is relevant to the landing page. With the click-through data as the information source, several researchers demonstrate that one can improve query suggestions and relevance [5, 25] and others focus on improving search results [1, 4]. One of the practical challenges of these approaches are the positional biases associated with clicks. In addition, it is unclear whether the fact that search results tend to contain the query words would limit their ability to provide accurate relevance judgements on out-of document phrases. Finally, these approaches may not be able to handle rarely queried phrases.

As for the online advertising application, our work can be viewed as a natural extension of keyword extraction from Web documents [22] to the goal of providing a consistent relevance judgement for potential advertising keywords that do not occurred in the document. While search and contextual advertising have traditionally focussed on keyword-driven ad placement, it is worth noting that using keywords to find relevant ads is not the only strategy for contextual advertising. The other paradigm that has been studied extensively is to judge the relevance between the ad and Web page directly. For example, Ribeiro-Neto *et al.* [15] investigated various strategies of representing both the ad and the Web document in vectors using extracted keywords and derived a cosine score to measure the ad relevance. In another recent paper by Broder *et al.*, they introduced a text classifier that maps the input text to its semantic category based on a pre-defined taxonomy. By using both syntactic and semantic information, they managed to enhance the relevance measure between the ad and Web page [2].

7. CONCLUSIONS

In this paper, we investigated two approaches for providing consistent relevance measures for both in and out-of document phrases. For similarity-based methods, we experimented with different configurations and found that combing them using a sigmoid function can outperform any individual method. However, when the appropriate kernel function and term-weighting scheme are used, the Gaussian Process Regression model performs the best, and is significantly better than the similarity-based methods and 10% better than a baseline method using bag-of-words vectors.

Despite its suboptimal performance, the similarity-based approach still enjoys some advantages in practice, such as its simplicity and low computational cost. Because the main operations of this approach are just the inner-product and the sigmoid transformation which is required only when a probabilistic measure is needed, it is very easy to implement it efficiently. In addition, the vector space representation of commonly used phrases can also be pre-computed further reducing the computational cost. Using the better relevance measure from the Gaussian Process Regression model requires an increased computational cost. While the vector space representation of commonly used phrases can be pre-

computed, the kernel function of each pair of input phrases, including the in-document phrases and the target out-of-document phrase, has to be computed. While caching the results for popular pairs of phrases might improve performance, pre-computing all the kernel computations is likely to be infeasible. In addition, the $O(N^3)$ computational complexity may be too high for systems that need to respond in real-time, even for a small set of input vectors.

In the future, we plan to explore several of the potential research directions suggested by this work. We found that the best configuration for similarity-based methods was the combined method, SimCombine. It is interesting to note that any of the similarity based methods can be used as an in-document relevance measure. Given the strong performance of SimCombine, it would be interesting to see if using it as a new term-weighting function would improve the performance of either the similarity-based or regression-based approaches. Another direction for improvement is model combination of our two basic approaches. Model combination is a popular and effective strategy for improving performance as illustrated by the good performance of SimCombine. Additional gains in improvement could come from application of other model combination methods especially when used to combine the similarity-based and regression-based methods. There are also a variety of interesting research directions in the application of our approaches to online advertising. In several on-line advertising applications, the goal is to improve the relevance of ads shown to users. This goal does not directly match the goal of the methods described in this paper, which are focussed on whether bid keywords are relevant to content. It would be interesting to (1) investigate the degree to which keyword relevance measures can improve ad relevance and (2) how to best extend these methods to directly measure the relevance of an ad to a content page.

8. ACKNOWLEDGMENTS

We thank Ewa Dominowska and Asela Gunawardana for their help in some preliminary experiments for this project. We also are grateful to John Platt and Susan Dumais for their useful comments and suggestions.

9. REFERENCES

[1] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR '06*, 2006.

[2] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. In *SIGIR '07*, 2007.

[3] C. Cortes and M. Mohri. Confidence intervals for the area under the roc curve. In *Advances in Neural Information Processing Systems 17*, 2005.

[4] C. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR '07*, 2007.

[5] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Probabilistic query expansion using query logs. In *Proceedings of the 11th international conference on World Wide Web (2002)*, pages 325–332, 2002.

[6] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. Domain-specific keyphrase extraction. In *Proc. of IJCAI-99*, pages 668–673, 1999.

[7] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *Proc. of WWW-06*, 2006.

[8] V. Lavrenko and W. B. Croft. Relevance-based language models. In *SIGIR '01*, pages 120–127, 2001.

[9] D. J. C. MacKay. Introduction to Gaussian processes. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, pages 133–166. Kluwer, 1998.

[10] D. Metzler, S. Dumais, and C. Meek. Similarity measures for short segments of text. In *Proc. of ECIR-07*, 2007.

[11] A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *ICML*, pages 625–632, 2005.

[12] J. Platt. Probabilities for support vector machines. In A. Smola, P. Bartlett, B. Scolkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT press, 2000.

[13] Y. Qiu and H.-P. Frei. Concept based query expansion. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–169, New York, NY, USA, 1993. ACM Press.

[14] C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. The MIT Press, Cambridge, MA, USA, 01 2006.

[15] B. Ribeiro-Neto, M. Cristo, P. B. Golgher, and E. S. de Moura. Impedance coupling in content-targeted advertising. In *SIGIR-05*, pages 496–503, 2005.

[16] S. Robertson and S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *SIGIR '94*, pages 345–354, 1994.

[17] S. Robertson, H. Zaragoza, and M. Taylor. Simple BM25 extension to multiple weighted fields. In *CIKM '04*, pages 42–49, 2004.

[18] M. Sahami and T. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proc. of WWW-06*, 2006.

[19] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[20] K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - parts 1 & 2. *Information Processing and Management*, 36(6), 2000.

[21] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *SIGIR '96*, pages 4–11, 1996.

[22] W. Yih, J. Goodman, and V. Carvalho. Finding advertising keywords on web pages. In *Proc. of WWW-06*, 2006.

[23] W. Yih and C. Meek. Improving similarity measures for short segments of text. In *Proc. of AAAI '07*, 2007.

[24] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01*, pages 334–342, 2001.

[25] Q. Zhao, S. C. H. Hoi, T.-Y. Liu, S. S. Bhowmick, M. R. Lyu, and W.-Y. Ma. Time-dependent semantic similarity measure of queries using historical click-through data. In *Proc. of WWW '06*, 2006.