Using Query Performance Predictors to Reduce Spoken Queries

Jaime Arguello[1], Sandeep Avula[1], and Fernando Diaz[2]

[1] University of North Carolina at Chapel Hill
[2] Microsoft Research

Abstract. The goal of query performance prediction is to estimate a query's retrieval effectiveness without user feedback. Past research has investigated the usefulness of query performance predictors for the task of reducing verbose textual queries. The basic idea is to automatically find a shortened version of the original query that yields a better retrieval. To date, such techniques have been applied to TREC topic descriptions (as surrogates for verbose queries) and to long textual queries issued to a web search engine. In this paper, we build upon an existing query reduction approach that was applied to TREC topic descriptions and evaluate its *generalizability* to the new task of reducing spoken query transcriptions. Our results show that we are able to outperform the original spoken query by a small, but significant margin. Furthermore, we show that the terms that are omitted from better-performing sub-queries include extraneous terms not central to the query topic, disfluencies, and speech recognition errors.

1 Introduction

Speech-enabled search allows users to formulate queries using spoken language. The search engine transcribes the spoken query using an automatic speech recognition (ASR) system and then runs the textual query against the collection. While speech is a natural means of communicating an information need, spoken queries pose a challenge for speech-enabled search engines, for two reasons: (1) spoken queries are longer than textual queries and may include terms that are not central to the query topic [5], and (2) spoken queries may have speech recognition errors that can cause a significant drop in retrieval performance [10].

In this paper, we focus on the task of automatically reducing spoken query transcriptions in order to improve retrieval performance. We evaluate an approach that extends the algorithm proposed by Kumaran and Carvalho [11], which was originally evaluated using TREC topic descriptions as surrogates for verbose textual queries. Our approach proceeds in three steps. First, given a spoken query transcription, we generate a set of candidate sub-queries to consider (including the original spoken query). Second, we use a regression model to predict each sub-query's retrieval performance compared to the original. Finally, we use a *weighted* rank fusion method to combine the rankings from the top-k sub-queries with the greatest predicted performance.

The regression model is trained to predict the difference in performance between a candidate sub-query and the original query as a function of a set of features. Following prior work, we experimented with three types of features: *pre-retrieval* query performance features, *post-retrieval* query performance features, and drift features. Our query performance features estimate the candidate sub-query's effectiveness. On the other hand, to avoid drifting too far from the original query topic, our drift features capture the relatedness between a candidate sub-query and the original. We present an evaluation on 5,000 spoken queries that were obtained using a crowdsourced study and transcribed using three freely available ASR systems provided by AT&T, IBM, and WIT.AI.

This paper makes the following contributions. First, we propose an extension of an existing query reduction approach and achieve comparable results on the task of reducing TREC topic descriptions. Second, we evaluate the *generalizablity* of our approach to the new task of reducing spoken queries. Third, we describe the types of spoken query terms that are dropped in order to improve retrieval performance, which suggest unique challenges and opportunities for improving spoken query retrieval. Finally, we describe our collection of 5,000 spoken queries which are based on the 250 TREC 2004 Robust Track topics and are therefore associated with a reusable IR test collection. Our spoken query transcriptions are available for others to extend our research.[3]

2 Related Work

Our work builds on three areas of prior research: (1) query performance prediction, (2) automatically reducing verbose queries, and (3) using query performance predictors to improve spoken query retrieval.

Query performance prediction: Query performance predictors estimate a query's effectiveness without user feedback. Pre-retrieval measures capture evidence such as the query's specificity, topical coherence, and estimated rank stability [8]. Query specificity measures consider the query terms' inverse document frequency (IDF) and inverse collection term frequency (ICTF) values [6, 9, 22]. Other specificity measures include the *query-scope*—proportional to the number of documents with at least one query term—and *simplified clarity*—equal to the KL-divergence between the query and collection language models [9]. Topical coherence can be measured using the degree of co-occurrence between query terms [8]. Finally, the rank stability can be estimated using the query terms' variance of TF.IDF weights across documents in the collection [22].

Post-retrieval measures capture evidence such as the topical coherence of the top results, the actual rank stability, and the extent to which similar documents obtain similar retrieval scores. The *clarity* score measures the KL-divergence between the language model of the top documents and a background model of the collection [6]. Rank stability methods perturb the query [20, 24], the documents [23], or the retrieval system [2], and measure the degree of change in the output ranking. Finally, the auto-correlation score from Diaz [7] considers the extent to which documents with a high text similarity obtain similar retrieval scores.

[3] `https://ils.unc.edu/~jarguell/ecir2017/`

Reducing verbose queries: Kumaran and Carvalho [11] focused on automatically reducing TREC topic descriptions. They used learning-to-rank (LTR) to predict the sub-query with the best performance and used query performance predictors as features. The authors focused on a heuristically-chosen sample of all possible sub-queries and found a 6.8% improvement in average precision on the TREC 2004 Robust Track collection. Balasubramanian *et al.* [3] evaluated a similar technique on verbose queries issued to a commercial web search engine and considered only sub-queries with $n-1$ terms. Xue *et al.* [19] focused on reducing TREC topic descriptions and trained a sequential model to label each query-term as 'keep' or 'do not keep' using query performance predictors as features. The authors found greater improvements by combining the predicted sub-query with the original. Xue and Croft extended this idea by combining sub-queries in a weighted fashion, setting the mixing parameters based on the LTR output [18]. Zhao and Callan trained a classifier to predict a query term's importance by combining performance predictors with features such as the query-term's rareness, abstractness, and ambiguity [21]. Their results found greater improvements for more verbose queries (i.e., TREC descriptions vs. titles).

Improving spoken query retrieval: Prior work has also considered improving spoken query recognition using evidence similar to some of the query performance predictors mentioned above. Mamou *et al.* [14] focused on re-ranking the ASR system's n-best list using term co-occurrence statistics in order to favor transcribed queries with semantically related terms. Li *et al.* [13] combined language models generated from different query-click logs to bias the ASR output in favor of previously run queries with clicks. Peng *et al.* [15] focused on re-ranking the n-best list using post-retrieval evidence such as the number of search results and the number of exact matches in the top results. Arguello *et al.* [1] used a wide-range of pre- and post-retrieval query performance predictors to re-rank the ASR system's n-best list.

3 Data Collection

Our spoken queries were collected as part of a user study reported in a previous paper. We provide a general description of the study and the ASR systems used, and refer the reader to Arguello *et al.* [1] for additional details.

User Study: Spoken queries were collected using Amazon Mechanical Turk (MTurk). Participants were given a search task description and were asked to produce a recording of how they would request the information from a speech-enabled search engine. Each MTurk Human Intelligence Task (HIT) proceeded as follows. First, participants were given a set of instructions and a link to a video explaining the HIT. Participants were then asked to click a "start" button to open the main voice recording page in a new browser tab. While loading, the main voice recording page asked participants to grant access to their computer's microphone. Participants were required to grant access in order to continue. The main voice-recording page provided participants with three items: (1) a "view task" button that displayed the search task description in a pop-up window, (2) Javascript widgets to record the spoken query and save the recording as a WAV file on their computer, and (3) an HTML form to upload the saved

WAV file to our server. The search task was displayed in a pop-up window to prevent participants from reading the search task description while producing their recording.

Each MTurk HIT was priced at $0.15 USD. We restricted our HITs to workers with a 95% acceptance rate or greater and to workers within the U.S. Finally, in order to gather spoken queries from a wide range of participants, each worker was allowed to complete a maximum of 100 HITs (2% of all HITs). In total, we collected spoken queries from 167 participants.

Search Tasks: We developed 250 search tasks based on the 250 topics from the TREC 2004 Robust Track. We used the TREC description and narrative as guidelines and situated each task in a background scenario that gave rise to the information need. We collected 20 spoken queries per search task for a total of 5,000 spoken queries. An example search task and spoken query are provided below.

> **TREC Topic ID and Title:** 303 - Hubble Telescope Achievements
> **TREC Description:** Identify positive accomplishments of the Hubble telescope since it was launched in 1991.
> **Search Task Description:** You recently saw a picture of space taken by the Hubble telescope and now you are curious about the scientific advances made possible by the Hubble telescope since its launch in 1991. Find information about the positive accomplishments of the Hubble telescope, which include the ability to gather new and better-quality data that has led to new discoveries, theories, and areas of inquiry.
> **Example Spoken Query:** "What scientific advances have been made as a result of the Hubble telescope?"

ASR Systems: In this work, we treated the ASR system as a "black box" and used three freely available speech-to-text APIs provided by AT&T, IBM, and WIT.AI. All three APIs accept a WAV file as input and return the most confident transcription in JSON format.

Spoken Queries vs. TREC Topic Descriptions: In this work, we test the *generalizability* of a query reduction approach on TREC topic descriptions and spoken query transcriptions. Thus, we were interested in the differences between TREC topic descriptions and the spoken queries produced by our participants. We focus on the query transcriptions produced by the AT&T API.

Our spoken queries are different than the 250 TREC topic descriptions from the 2004 Robust Track in two important ways. First, our spoken queries are shorter. Including stopwords, our spoken queries have an average of 10.11 ± 4.81 words, while TREC topic descriptions have an average of 16.76 ± 8.89 words. Excluding stopwords, our spoken queries have an average of 5.055 ± 2.22 words, while TREC topic descriptions have an average of 9.12 ± 5.32.[4] Both TREC topic descriptions and our spoken queries were about 45% stopwords.

Second, when issued as queries against the TREC 2004 Robust Track collection, TREC topic descriptions produced better retrievals than our spoken

[4] We used the SMART stopword list.

queries. TREC topic descriptions achieved an average precision of 0.240, while our spoken queries achieved an average precision of 0.113.

Taken together, these two trends suggest that reducing spoken queries may be more difficult than reducing TREC topic descriptions. Our spoken queries had fewer topical terms and a lower baseline performance.

4 Algorithm

The goal of our algorithm is to select sub-queries that perform better than the original query transcription. Our approach is similar to the one proposed by Kumaran and Carvalho [11], and proceeds in three steps: (1) generate a candidate set of sub-queries to consider (including the original), (2) predict the retrieval performance of each candidate sub-query, and (3) combine the retrievals from the top-k sub-queries with the highest predicted performance.

Step 1: A query with n terms has $2^n - 1$ sub-queries (excluding the null query). We considered a much smaller subset of sub-queries using the following heuristics. First, we only considered sub-queries with 3-6 terms. Second, we only considered sub-queries with at least one noun. Third, to favor topically coherent queries, we only considered the 25 sub-queries with the highest average mutual information between query-term pairs. Finally, we included the original query in the candidate set. Similar heuristics were used in prior work [11].

Step 2: To perform the second step, we trained a regression model to predict each candidate sub-query's absolute increase or decrease in retrieval performance compared to the original query. We trained support vector regression models using the LibLinear toolkit.[5] At test time, we simply selected the candidate sub-queries with the greatest predicted performance. As described in more detail below, we measured retrieval performance in terms of P@10, NDCG@30, and average precision (AP). We trained different regression models for different metrics. Each sub-query was represented as a vector of features (Section 5), and feature values were normalized to zero-min and and unit-max separately for each candidate set of sub-queries. In other words, we used each feature's min and max values from the set of sub-queries associated with the *same* spoken query.

Step 3: Finally, to perform the third step, we used a *weighted* version of the Reciprocal Rank Fusion (RRF) method [4] to combine the document rankings from the top-k sub-queries with the *greatest* predicted performance. Let $\mathcal{R}_i$ denote the document ranking from the i^{th} sub-query with the greatest predicted performance, and let $\mathcal{R}_i(d)$ denote the rank of document d in $\mathcal{R}_i$. Documents retrieved by the top-k sub-queries were scored as follows:

$$\text{score}(d) = \sum_{i=1}^{k} \frac{1}{i} \times \frac{1}{t + \mathcal{R}_i(d)}. \tag{1}$$

Parameter t mitigates the impact of highly ranked documents that are outliers, and we set it to $t = 60$ based on prior work [4].

[5] `https://www.csie.ntu.edu.tw/~cjlin/liblinear/`

5 Features

We used three types of features: pre-retrieval query performance predictors, post-retrieval query performance predictors, and drift features. The numbers in parentheses indicate the number of features in each group.

Pre-retrieval Features (5): Prior work shows that well-performing queries tend to contain discriminative terms that appear in only a few documents. We included four features aimed to capture this type of evidence. Following prior work, we included the average inverse document frequency (IDF) value across query terms [9, 6, 22]. The *query-collection similarity* (QCS) score measures the extent to which the query terms appear many times in only a few documents [22]. The *query scope* score is inversely proportional to the number of documents with at least one query term [9]. Finally, the *simplified clarity* score measures the KL-divergence between the query and collection language models [9].

Prior work also shows that a query is more likely to perform well if the query terms describe a coherent topic. We included one feature to capture this type of evidence. Our point-wise mutual information (PMI) feature measures the average degree of co-occurrence between query-term pairs [8].

Post-Retrieval Features (6): A query is more likely to perform well if the top-ranked documents describe a coherent topic. We included five features aimed to capture this type of evidence. The *clarity* score measures that KL-divergence between the language model of the top results and the collection [6]. The *query feedback* score measures the degree of overlap between the top results before and after query-expansion [24]. A greater overlap suggests that the original query is more on-topic. Finally, we considered the *normalized query commitment* (NQC) score, which measures the standard deviation of the top document scores. We included three NQC scores: the standard deviation of the top document scores, the standard deviation of the scores *above* the mean top-document score, and the standard deviation of the scores *below* the mean top-document score [16].

Prior work also shows that in an effective retrieval, similar documents have similar retrieval scores [7]. We included one feature to model this type of evidence. The autocorrelation score from Diaz [7] measures to extent to which results with a high cosine similarity have similar scores.

Drift Features (2): The aim of our drift features was to favor sub-queries that do not drift too far from the original. We included two features to measure this type of evidence. Our *relevance model similarity* feature computes the similarity between the language model of the top results from the original query and the candidate sub-query. Following Lavrenko *et al.* [12], relevance models were estimated by combining the language models of the top-10 results weighted by their retrieval scores. The relevance model similarity was computed using the Bhattacharyya correlation. Finally, we measured the Jaccard coefficient between the top-10 results from the original and candidate sub-query. All drift feature values were 1.0 for the original query, which was included in the candidate set.

6 Evaluation Methodology

Retrieval performance was measured by issuing queries against the TREC 2004 Robust Track collection. We used Lucene's implementation of the query-likelihood model with Dirichlet smoothing ($\mu = 1000$), and used the Krovetz stemmer and the SMART stopword list. We evaluated in terms of P@10, NDCG@30, and average precision (AP).

Models were evaluated using 20-fold cross-validation. In order to train and test using spoken queries from *different* TREC topics, all 20 spoken queries for the same topic were assigned to the same fold. We report average performance across held-out folds and measured statistical significance using the approximation of Fisher's randomization test described in Smucker *et al.* [17]. We used the same cross-validation folds in all our experiments. Thus, when measuring statistical significance, the randomization was applied to the 20 pairs of performance values for the two models being compared.

We compare against two baseline approaches: (1) selecting the best-performing candidate sub-query (oracle) and (2) running the original spoken query transcription (original). Parameter k and SVM regression parameter c were tuned by doing a second level of cross-validation.

7 Results

Our evaluation results are presented in Table 1. We present results using the ASR output from our three speech-to-text APIs: AT&T (Table 1a), IBM (Table 1b), and WIT.AI (Table 1c). Additionally, we applied our approach to the task of reducing TREC topic descriptions (Table 1d). We present results in terms of average precision (AP), P@10 and NDCG@30.

The rows labeled original show the performance of the original spoken query in Tables 1a-1c and the original TREC topic description in Table 1d. The rows labeled all show the performance of our models using *all* features. The rows labeled no.x show the performance of our models using all features *except* for pre-retrieval query performance features (no.pre), post-retrieval query performance features (no.post) and drift features (no.drift). The rows labeled oracle show the performance of the best candidate sub-query. This is not a "true" oracle experiment because we did not consider every possible sub-query. However, it determines whether Step 1 in our approach was able to select sub-queries that perform better than the original.

In Step 3 of our approach, we combined the rankings from the top-k sub-queries with the greatest predicted performance using Equation 1. We were interested in evaluating the contribution of this step to retrieval performance. To this end, we considered three additional alternatives: (1) selecting the *single* sub-query with the greatest predicted performance ($k = 1$), (2) combining the rankings from *all* candidate sub-queries in a weighted fashion as described in Equation 1 ($k = \max$), and (3) combining the rankings from all candidate sub-queries in an *unweighted* fashion by omitting factor $1/i$ from Equation 1 ($k = \max$, unweighted).

Table 1: Results using TREC topic descriptions and the spoken query transcriptions generated using the AT&T, IBM, and WIT.AI APIs. The percentages indicate percent improvement over the original query (original). A ▲ and ▼ denotes a significant increase and decrease in performance compared to original, respectively. We report significance at the $p < .05$ level using Bonferroni correction.

	AP	P@10	NDCG@30
original	0.113	0.206	0.197
all	0.119 (5.31%)▲	0.210 (2.25%)▲	0.203 (3.12%)▲
all (k=1)	0.116 (2.65%)▲	0.207 (0.67%)	0.200 (1.63%)
all (k=max)	0.118 (4.42%)▲	0.208 (1.27%)	0.202 (2.37%)
all (k=max, unweighted)	0.109 (-3.54%)▼	0.199 (-2.99%)▼	0.191 (-3.13%)▼
no.pre	0.118 (4.42%)▲	0.208 (1.16%)	0.202 (2.32%)▲
no.post	0.115 (1.77%)	0.207 (0.79%)	0.200 (1.31%)
no.drift	0.118 (4.42%)▲	0.206 (0.01%)	0.199 (1.15%)
oracle	0.146 (29.20%)▲	0.285 (38.56%)▲	0.258 (31.20%)▲

(a) AT&T spoken query transcriptions

	AP	P@10	NDCG@30
original	0.165	0.293	0.282
all	0.173 (4.85%)▲	0.300 (2.16%)▲	0.290 (2.77%)▲
all (k=1)	0.170 (3.03%)▲	0.296 (0.82%)	0.286 (1.35%)
all (k=max)	0.173 (4.85%)▲	0.300 (2.32%)▲	0.290 (2.75%)▲
all (k=max, unweighted)	0.160 (-3.03%)▼	0.288 (-2.04%)▼	0.276 (-2.23%)▼
no.pre	0.173 (4.85%)▲	0.299 (2.00%)▲	0.290 (2.72%)▲
no.post	0.168 (1.82%)	0.296 (0.86%)	0.286 (1.20%)
no.drift	0.171 (3.64%) ▲	0.295 (0.65%)	0.285 (0.77%)
oracle	0.211 (27.88%)▲	0.395 (34.60%)▲	0.361 (27.74%)▲

(b) IBM spoken query transcriptions

	AP	P@10	NDCG@30
original	0.183	0.321	0.308
all	0.191 (4.37%)▲	0.327 (1.77%)	0.317 (2.70%)▲
all (k=1)	0.188 (2.73%)▲	0.324 (0.77%)	0.312 (1.16%)
all (k=max)	0.190 (3.83%)▲	0.326 (1.50%)	0.316 (2.42%)▲
all (k=max, unweighted)	0.177 (-3.28%)▼	0.314 (-2.26%)▼	0.302 (-2.15%)▼
no.pre	0.192 (4.92%)▲	0.326 (1.44%)	0.316 (2.52%)▲
no.post	0.185 (1.09%)	0.323 (0.42%)	0.312 (1.14%)
no.drift	0.190 (3.83%)▲	0.323 (0.61%)	0.311 (1.02%)
oracle	0.228 (24.59%)▲	0.422 (31.26%)▲	0.385 (24.99%)▲

(c) WIT.AI spoken query transcriptions

	AP	P@10	NDCG@30
original	0.240	0.403	0.384
all	0.252 (5.00%)▲	0.417 (3.49%)	0.393 (2.37%)
all (k=1)	0.245 (2.08%)	0.403 (-0.05%)	0.387 (0.69%)
all (k=max)	0.245 (2.08%)	0.403 (0.02%)	0.384 (0.00%)
all (k=max, unweighted)	0.225 (-6.25%)▼	0.380 (-5.67%)▼	0.361 (-6.04%)▼
no.pre	0.255 (6.25%)▲	0.411 (2.00%)	0.395 (2.70%)
no.post	0.240 (0.00%)	0.397 (-1.42%)	0.379 (-1.38%)
no.drift	0.251 (4.58%)▲	0.403 (-0.11%)	0.388 (1.08%)
oracle	0.301 (25.42%)▲	0.544 (34.87%)▲	0.491 (27.68%)▲

(d) TREC Topic Descriptions

The results in Table 1 show seven important trends. First, overall retrieval performance was better for the IBM and WIT.AI APIs than the AT&T API. As it turns out, the AT&T API had more ASR errors, possibly because it uses a language model less well-suited for queries or for the topics associated with the 2004 Robust Track collection. Our goal was not to compare speech-to-text APIs. However, as described below, our results suggest that we can improve retrieval performance for spoken queries with varying degrees of ASR error.

Second, across all APIs and evaluation metrics, our models using all features (all) performed at the same level or significantly better than the baseline of running the original query (original). Improvements were higher in terms of AP than P@10 and NDCG@30, suggesting that our approach was able to retrieve more relevant documents beyond the top-10 results. In terms of AP, performance improvements compared to the original query were in the 4-5% range. We observed similar trends on TREC topic descriptions. On the task of reducing TREC topic descriptions, Kumaran and Carvalho [11] reported a 6.8% improvement in AP on the same TREC 2004 Robust Track collection. In our case, we observed a 5.0% improvement when using all features (all) and a 6.25% improvement when ignoring pre-retrieval features (no.pre).

Third, our approach (all) outperformed the alternative of selecting the *single* sub-query with the greatest predicted performance ($k = 1$). In all cases, setting $k = 1$ resulted in a drop in retrieval performance. This result suggests that combining the rankings from the top sub-queries yields a more robust solution.

Fourth, our results show that combining the rankings from all candidate sub-queries in a weighted fashion ($k = \max$) is a reasonable alternative. In most cases, setting $k = \max$ resulted in only a slight drop in performance. This result shows that our approach is not very sensitive to parameter k. In retrospect, this makes sense, as factor $1/i$ in Equation 1 places much more emphasis on the top sub-queries than the bottom ones.

Fifth, combining rankings from all sub-queries in an *unweighted* fashion ($k = \max$, unweighted) resulted in a large drop in performance. In fact, in all cases, the drop in performance compared to the original query was statistically significant. This result shows that effectively reducing spoken queries (and TREC topic descriptions) is not simply a matter of combining sub-queries without first estimating their retrieval performance. In other words, this result validates Steps 2 and 3 of our approach.

Sixth, our feature ablation results suggest that pre-retrieval query performance features were the least predictive and that post-retrieval features were the most predictive. Omitting pre-retrieval features (no.pre) resulted in the lowest drop in performance. In most cases, no.pre still performed significantly better than the original baseline. In contrast, omitting post-retrieval features (no.post) resulted in the largest drop in performance. In all cases, no.post was statistically equal to the original baseline. This result is consistent with prior work that shows that, while post-retrieval features are more computationally expensive, they provide valuable evidence [11, 19].

The final trend worth noting is that there is still room for improvement. Across all APIs and metrics, the oracle significantly outperformed the original query (original) and all our models by a large margin.

8 Discussion

Sub-query Effectiveness: Based on our results, it is clear that some candidate sub-queries perform better than others. For example, combining the rankings from all candidate sub-queries in a *weighted* fashion (based on their predicted performance) outperformed combining the rankings in an *unweighted* fashion. A natural follow-up question is: On average, what percentage of the candidate sub-queries outperformed the original query? For our spoken queries, the average percentage of candidate sub-queries that outperformed the original query were: AT&T= 29.22% ± 22.22%, IBM= 31.74% ± 21.38%, and WIT.AI= 31.58% ± 20.80%. Similarly, for TREC topic descriptions, the average percentage of better-performing sub-queries was 30.65%±22.22%. Across all datasets, most candidate sub-queries did not outperform the original. Thus, any method that uses sub-queries to reduce verbose queries needs to be selective about which sub-queries to focus on.

Reducing Spoken Queries: Our results in Table 1 show that we can improve retrieval performance by dropping terms from the original spoken query. We were interested in better understanding what are the types of original query terms that are omitted from a better-performing sub-query. To answer this question, we counted the number of times each term was omitted from a candidate sub-query that outperformed the original query in terms of AP. For this and the next analysis, we focus on the recognition output from the AT&T API.

The following are the top-50 most frequently dropped terms: information (2189), find (934), country (660), show (592), states (535), united (510), people (471), current (343), affect (313), list (275), negative (274), um (270), america (263), world (238), government (237), company (234), effects (229), con (226), recent (226), place (222), pro (221), type (218), call (216), industry (210), work (209), history (209), case (202), conditions (190), tax (189), international (184), worldwide (176), activity (172), treatment (170), human (163), news (159), project (158), happen (158), instance (156), law (156), impact (156), involve (154), nineteen (148), made (147), side (146), system (145), increase (142), group (142), number (139), document (138), and search (138).

Interestingly, we see three types of terms. First, we see several imperative verbs and nouns associated with 'requesting information' (e.g., find, show, list, search, information, document). Second, we see at least one disfluency (e.g., um). Third, we see terms describing *extra-topical* dimensions of the information need. For example, we see terms that suggest the desire for information about a specific time frame (e.g., history, recent, current, news), as well as terms that suggest the desire for information about a particular perspective (e.g., negative, pro, con). This last category is particularly interesting. Such terms may be problematic for search systems because they may not frequently appear in relevant documents. For instance, a document discussing historic or recent events may not actually

contain the terms 'history' or 'recent'. Future work might consider whether such extra-topical terms are more popular in spoken versus textual queries.

Finally, we expected that dropping speech recognition errors would yield better-performing sub-queries. Indeed, we found evidence of this in our results. For example, we found cases where the spoken term 'lyme' was misrecognized as 'line' and omitted from better-performing sub-queries. Other example pairs (x,y) where the spoken term x was misrecognized as y and subsequently omitted from a better-performing sub-query include: (apirin, aprin); (beatify, beautify), (cult, colt); (export, expert); (fatal, foetal); (france, francis); (czech, check); (melanoma, melonoma); (nobel, noble); (pisa, pizza); (vegetation, visitation); (role, roll); and (soil, swell).

9 Conclusion

We presented an approach for reducing spoken queries. Our approach is an extension of the algorithm proposed by Kumaran and Carvalho [11], which was applied to the task of reducing TREC topic descriptions. We were able to closely approximate the level of performance reported in Kumaran and Carvalho [11] and tested the generalizability of our approach on the new task of reducing spoken queries.

Our results suggest three major trends. First, our approach yielded small, but significant improvements over the baseline of running the original transcription as the query. Second, combining the rankings from the top-k sub-queries in a weighted fashion yielded the best performance—it performed better than simply selecting the single sub-query with the greatest predicted performance and better than combining all candidate sub-queries in an *unweighted* fashion. Finally, post-retrieval query performance features were more predictive than pre-retrieval query performance features and drift features.

A post-hoc analysis found that the types of terms that are omitted from a better-performing sub-query include a combination of: (1) terms that are not central to the query topic (e.g., find, information), (2) disfluencies (e.g., um, eh), (3) terms that describe extra-topical dimensions of the information need, and (4) speech recognition errors.

Our findings point to several directions for future work. First, our results suggest several additional features that might be useful for predicting sub-query performance. For instance, non-topical terms such as 'find' and 'information' might tend to appear towards the beginning of a spoken query. Thus, features that characterize the relative positions of the dropped query terms might improve sub-query prediction performance. Also, ASR systems sometimes include term confidence values in the output transcription. Features that characterize the ASR confidence values of the dropped terms might also be useful. Finally, future work should consider whether terms associated with extra-topical dimensions of the information need, such as terms that convey temporal constraints ('historic', 'recent') or perspective constraints ('pros', 'cons'), are more common in spoken versus textual queries.

Acknowledgments. This work was supported in part by NSF grant IIS-1451668. Any opinions, findings, conclusions, and recommendations expressed in this paper are the authors' and do not necessarily reflect those of the sponsors.

References

1. J. Arguello, S. Avula, and F. Diaz. Using query performance predictors to improve spoken queries. In *ECIR*, 2016.
2. J. A. Aslam and V. Pavlu. Query hardness estimation using jensen-shannon divergence among multiple scoring functions. In *ECIR*, 2007.
3. N. Balasubramanian, G. Kumaran, and V. R. Carvalho. Exploring reductions for long web queries. In *SIGIR*, 2010.
4. G. V. Cormack, C. L. A. Clarke, and S. Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *SIGIR*, 2009.
5. F. Crestani and H. Du. Written versus spoken queries: A qualitative and quantitative comparative analysis. *JASIST*, 57(7), 2006.
6. S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *SIGIR*, 2002.
7. F. Diaz. Performance prediction using spatial autocorrelation. In *SIGIR*, 2007.
8. C. Hauff. *Predicting the Effectiveness of Queries and Retrieval Systems.* dissertation, Univeristy of Twente, 2010.
9. B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. In *SPIRE*, 2004.
10. J. Jiang, W. Jeng, and D. He. How do users respond to voice input errors?: Lexical and phonetic query reformulation in voice search. In *SIGIR*, 2013.
11. G. Kumaran and V. R. Carvalho. Reducing long queries using query quality predictors. In *SIGIR*, 2009.
12. V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR*, 2001.
13. X. Li, P. Nguyen, G. Zweig, and D. Bohus. Leveraging multiple query logs to improve language models for spoken query recognition. In *ICASSP*, 2009.
14. J. Mamou, A. Sethy, B. Ramabhadran, R. Hoory, and P. Vozila. Improved spoken query transcription using co-occurrence information. In *INTERSPEECH*, 2011.
15. F. Peng, S. Roy, B. Shahshahani, and F. Beaufays. Search results based n-best hypothesis rescoring with maximum entropy classification. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013.
16. A. Shtok, O. Kurland, D. Carmel, F. Raiber, and G. Markovits. Predicting query performance by query-drift estimation. *TOIS*, 30(2), 2012.
17. M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *CIKM*, 2007.
18. X. Xue and W. B. Croft. Modeling subset distributions for verbose queries. In *SIGIR*, 2011.
19. X. Xue, S. Huston, and W. B. Croft. Improving verbose queries using subset distribution. In *CIKM*, 2010.
20. E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow. Learning to estimate query difficulty: Including applications to missing content detection and distributed information retrieval. In *SIGIR*, 2005.
21. L. Zhao and J. Callan. Term necessity prediction. In *CIKM*, 2010.
22. Y. Zhao, F. Scholer, and Y. Tsegay. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *ECIR*, 2008.
23. Y. Zhou and W. B. Croft. Ranking robustness: A novel framework to predict query performance. In *CIKM*, 2006.
24. Y. Zhou and W. B. Croft. Query performance prediction in web search environments. In *SIGIR*, 2007.