Simulating Early-Termination Search for Verbose Spoken Queries

Jerome White
IBM Research
Bangalore, KA India
jerome.white@in.ibm.com

Douglas W. Oard
University of Maryland
College Park, MD USA
oard@umd.edu

Nitendra Rajput
IBM Research
New Delhi, India
rnitendra@in.ibm.com

Marion Zalk
University of Melbourne
Melbourne, VIC Australia
m.zalk@student.unimelb.edu.au

Abstract

Building search engines that can respond to spoken queries with spoken content requires that the system not just be able to find useful responses, but also that it know when it has heard enough about what the user wants to be able to do so. This paper describes a simulation study with queries spoken by non-native speakers that suggests that indicates that finding relevant content is often possible within a half minute, and that combining features based on automatically recognized words with features designed for automated prediction of query difficulty can serve as a useful basis for predicting when that useful content has been found.

1 Introduction

Much of the early work on what has come to be called "speech retrieval" has focused on the use of text queries to rank segments that are automatically extracted from spoken content. While such an approach can be useful in a desktop environment, half of the world's Internet users can access the global information network only using a voice-only mobile phone. This raises two challenges: 1) in such settings, both the query and the content must be spoken, and 2) the language being spoken will often be one for which we lack accurate speech recognition.

The Web has taught us that the "ten blue links" paradigm can be a useful response to short queries. That works because typed queries are often fairly precise, and tabular responses are easily skimmed. However, spoken queries, and in particular open-domain spoken queries for unrestricted spoken content, pose new challenges that call for new thinking about interaction design. This paper explores the potential of a recently proposed alternative, in which the spoken queries are long, and only one response can be played at a time by the system. This approach, which has been called Query by Babbling, requires that the user ramble on about what they are looking for, that the system be able to estimate when it has found a good response, and that the user be able to continue the search interaction by babbling on if the first response does not fully meet their needs (Oard, 2012).

One might question whether users actually will "babble" for extended periods about their information need. There are two reasons to believe that some users might. First, we are particularly interested in ultimately serving users who search for information in languages for which we do not have usable speech recognition systems. Speech-to-speech matching in such cases will be challenging, and we would not expect short queries to work well. Second, we seek to principally serve users who will be new to search, and thus not yet conditioned to issue short queries. As with Web searchers, we can expect them to explore initially, then to ultimately settle on query strategies that work well enough to meet their needs. If longer queries work better for them, it seems reasonable to expect that they would use longer queries. Likewise, if systems cannot effectively use longer queries to produce useful results, then people will not use them.

To get a sense for whether such an interaction modality is feasible, we performed a simulation

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1270–1280,
Seattle, Washington, USA, 18-21 October 2013.

study for this paper in which we asked people to babble on some topic for which we already have relevance judgments results. We transcribe those babbles using automatic speech recognition (ASR), then note how many words must be babbled in each case before an information retrieval system is first able to place a relevant document in rank one. From this perspective, our results show that people are indeed often able to babble usefully; and, moreover, that current information retrieval technology could often place relevant results at rank one within half a minute or so of babbling even with contemporary speech recognition technology.

The question then arises as to whether a system can be built that would recognize when an answer is available at rank one. Barging in with an answer before that point wastes time and disrupts the user; barging in long after that point also wastes time, but also risks user abandonment. We therefore want a "Goldilocks" system that can get it just about right. To this end, we introduce an evaluation measure that differentially penalizes early and late responses. Our experiments using such a measure show that systems can be built that, on average, do better than could be achieved by any fixed response delay.

The remainder of this paper is organized as follows: We begin in Section 2 with a brief review of related work. Section 3 then describes the design of the ranking component of our experiment; Section 4 follows with some exploratory analysis of the ranking results using our test collection. Section 6 completes the description of our methods with an explanation of how the stopping classifier is built; Section 7 then presents end-to-end evaluation results using a new measure designed for this task. Section 8 concludes the paper with some remarks on future work.

2 Background

The rapid adoption of remarkably inexpensive mobile telephone services among low-literacy users in developing and emerging markets has generated considerable interest in so-called "spoken forum" projects (Sherwani et al., 2009; Agarwal et al., 2010; Medhi et al., 2011; Mudliar et al., 2012). It is relatively straightforward to collect and store spoken content regardless of the language in which it is spoken; organizing and searching that content is, however, anything but straightforward. Indeed, the current lack of effective search services is one of the key inhibitors that has, to date, limited spoken forums to experimental settings with at most a few hundred users. If a "spoken web" is to achieve the same degree of impact on the lives of low-literacy users in the developing world that the World Wide Web has achieved over the past decade in the developed world, we will need to develop the same key enabler: an effective search engine.

At present, spoken dialog systems of conventional design, such as Siri, rely on complex and expensive language-specific engineering, which can easily be justified for the "languages of wealth" such as English, German, and Chinese; but perhaps not for many of the almost 400 languages that are each spoken by a million or more people.[1] An alternative would be to adopt more of an "information retrieval" perspective by directly matching words spoken in the query with words that had been spoken in the content to be searched. Some progress has been made on this task in the MediaEval benchmark evaluation, which has included a spoken content matching task each year since 2011 (Metze et al., 2012). Results for six low-resource Indian and African languages indicate that miss rates of about 0.5 can be achieved on individual terms, with false alarm rates below 0.01, by tuning acoustic components that had originally been developed for languages with reasonably similar phonetic inventories. Our goal in this paper is to begin to explore how such capabilities might be employed in a complete search engine for spoken forum content, as will be evaluated for the first time at MediaEval 2013.[2] The principal impediment to development in this first year of that evaluation is the need for relevance judgments, which are not currently available for spoken content of the type we wish to search. That consideration has motivated our design of the simulation study reported in this paper.

[1] http://www.ethnologue.com/statistics/size

[2] http://www.multimediaeval.org/mediaeval2013/qa4sw2013/

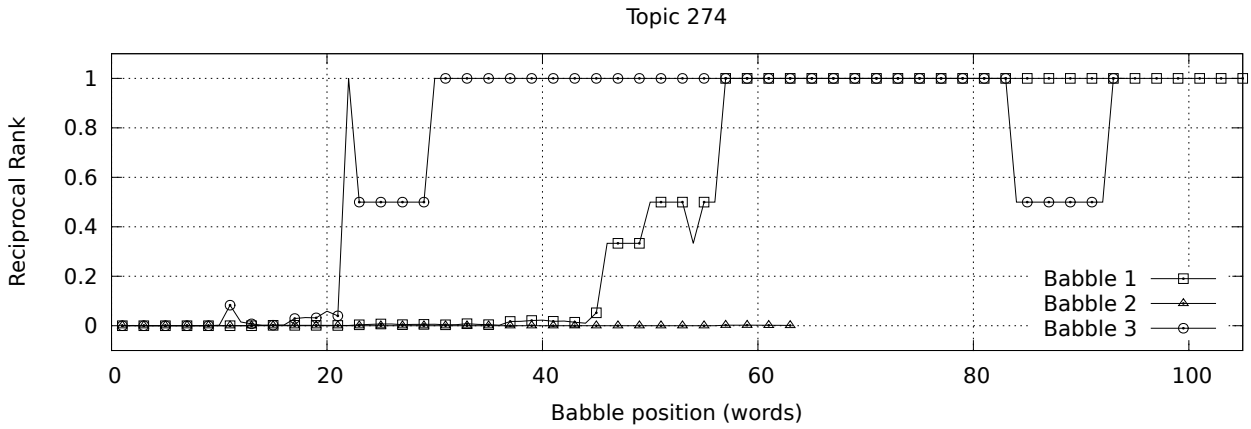

Figure 1: Reciprocal ranks at for each query making up a given babble. When retrieving results, a babbler either "latches" on to a relevant document (Babble 1), moves back-and-forth between relevant documents (Babble 3), or fails to elicit a relevant document at all (Babble 2).

3 Setup and Method

The approach taken in this paper is to simulate, as closely as possible, babbling about topics for which we a) already have relevance judgments available, and b) have the ability to match partial babbles with potential answers in ways that reflect the errors introduced by speech processing. To this end, we chose to ask non-native English speakers to babble, in English, about an information need that is stimulated by an existing English Text Retrieval Conference (TREC) topic for which we already have relevance judgments. An English Automatic Speech Recognition (ASR) system was then used to generate recognized words for those babbles. Those recognized words, in turn, have been used to rank order the (character-coded written text) news documents that were originally used in TREC, the documents for which we have relevance judgments. Our goal then becomes twofold: to first rank the documents in such a way as to get a relevant document into rank one; and then to recognize when we have done so.

Figure 1 is a visual representation of retrieval results as a person babbles. For three different babbles prompted by TREC Topic 274, it shows the reciprocal rank for the query that is posed after each additional word is recognized. We are primarily interested in cases where the reciprocal rank is one.[3] In these three babbles we see all cases that the retrieval system must take into account: babbles that never yield a relevant first-ranked document (Babble 2); babbles that eventually yield a relevant first-rank document, and that continue to do so as the person speaks (Babble 1); and babbles that alternate between good and bad results as the speaker continues (Babble 3).

3.1 Acquiring Babbles

Ten TREC-5 Ad Hoc topics were selected for this study: 255, 257, 258, 260, 266, 271, 274, 276, 287, and 297 based on our expectation of which of the 50 TREC 5 topics would be most suitable for prompted babbles. In making this choice, we avoided TREC topics that we felt would require specialized domain knowledge, experience with a particular culture, or detailed knowledge of an earlier time period, such as when the topics had been crafted. For each topic, three babbles were created by people speaking at length about the same information need that the TREC topic reflected. For convenience, the people who created the babbles were second-language speakers of English selected from information technology companies. There were a total of ten babblers; each recorded, in English, babbles for three topics, yielding a total of thirty babbles. We maintained a balance across topics when assigning topic

[3] A reciprocal rank of one indicates that a known relevant document is in position one; a reciprocal rank of 0.5 indicates that the most highly ranked known relevant document is in position two; 0.33 indicates position three; and so on.

Transcribed babble	Text from ASR
So long time back one of my friend had a Toyota Pryus it uses electric and petrol to increase the to reduce the consumption and increase the mileage I would now want to get information about why car operators manufacturers or what do they think about electric vehicles in the US well this is what the stories say that the car lobby made sure that the electric vehicles do not get enough support and the taxes are high by the government but has it changed now are there new technologies that enable to lower cost and also can increase speed for electric vehicles I am sure something is being done because of the rising prices of fuel these days	So long time at one of my friends headed towards the previous accuses electric in petrol to increase the to reduce the consumption and increase the minutes and would now want to get information about why car operator manufacturers on what to think about electric vehicles in the us versus what the story said that the car lobby make sure that the electric vehicles to not get enough support to an attack and I try to comment but has changed now arctic new technologies that enabled to cover costs and also can increase speak for electric vehicles I'm sure some clinton gore carls junior chef

Table 1: Text from an example babble (274-1). The left is transcribed through human comprehension; the right is the output from an automatic speech recognition engine.

numbers to babblers. All babblers had more than sixteen years of formal education, had a strong command on the English language, and had some information about the topics that they selected. They were all briefed about our motivation for collecting this data, and about the concept of query by babbling.

The babbles were created using a phone interface. Each subject was asked to call an interactive voice response (IVR) system. The system prompted the user for a three digit topic ID. After obtaining the topic ID, the system then prompted the user to start speaking about what they were looking for. TREC topics contain a short title, a description, and a narrative. The title is generally something a user might post as an initial Web query; the description is something one person might say to another person who might then help them search; the narrative is a few sentences meant to reflect what the user might jot down as notes to themselves on what they were actually looking for. For easy reference, the system provided a short description—derived from the description and narrative of the TREC topics—that gave the user the context around which to speak. The user was expected to begin speaking after hearing a system-generated cue, at which time their speech was recorded. Two text files were produced from the audio babbles: one produced via manual transcription,[4] and one produced by an ASR system; Table 1 presents an example. The ASR transcripts of the babbles were used by our system as a basis for ranking, and as a basis for making the decision on when to barge-in, what we call the "stopping point." The manual transcriptions were used only for scoring the Word Error Rate (WER) of the ASR transcript for each babble.

TREC Topic		WER	
ID	Title	Mean	SD
255	Environmental protect.	0.434	0.203
257	Cigarette consumption	0.623	0.281
258	Computer security	0.549	0.289
260	Evidence of human life	0.391	0.051
266	Prof. scuba diving	0.576	0.117
271	Solar power	0.566	0.094
274	Electric automobiles	0.438	0.280
276	School unif./dress code	0.671	0.094
287	Electronic surveillance	0.519	0.246
297	Right to die pros/cons	0.498	0.181
	Average	0.527	0.188

Table 2: Average ASR Word Error Rate over 3 babbles per topic (SD=Standard Deviation).

[4] The transcriber is the third author of this paper.

Babble	Words	Judgment at First Rank			Scorable	First Rel	Last Rel	WER
		Relevant	Not Relevant	Unknown				
257-3	74	5	64	5	93%	@13	@66	0.414
276-3	61	7	46	8	87%	@36	@42	0.720
258-1	146	2	118	26	82%	@28	@29	0.528
297-1	117	58	19	40	66%	@56	@117	0.594
274-3	94	57	0	47	61%	@22	@94	0.250
274-1	105	49	13	43	59%	@57	@105	0.437
257-1	191	104	0	87	54%	@52	@188	0.764
271-1	145	42	26	76	48%	@38	@109	0.556
287-2	61	26	0	35	43%	@33	@61	0.889
260-2	93	22	8	63	32%	@69	@93	0.500
276-2	69	11	2	56	19%	@47	@69	0.795
260-3	82	6	8	68	17%	@17	@62	0.370
258-2	94	14	1	79	16%	@24	@60	0.389
297-3	90	4	2	84	7%	@52	@56	0.312
266-2	115	6	0	109	5%	@47	@52	0.745

Table 3: Rank-1 relevance ("Rel") judgments and position of first and last scorable guesses.

3.2 System Setup

The TREC-5 Associated Press (AP) and Wall Street Journal (WSJ) news stories were indexed by Indri (Strohman et al., 2004) using the Krovetz stemmer (Krovetz, 1993), standard English stopword settings, and language model matching. Each babble was turned into a set of nested queries by sequentially concatenating words. Specifically, the first query contained only the first word from the babble, the second query only the first two words, and so on. Thus, the number of queries presented to Indri for a given babble was equivalent to the number of words in the babble, with each query differing only by the number of words it contained. The results were scored using `trec_eval` version 9.0. For evaluation, we were interested in the reciprocal rank; in particular, where the reciprocal rank was one. This measure tells us when Indri was able to place a known relevant document at rank one.

4 Working with Babbles

Our experiment design presents three key challenges. The first is ranking well despite errors in speech processing. Table 2 shows the average Word Error Rate (WER) for each topic, over three babbles. Averaging further over all thirty babbles, we see that about half the words are correctly recognized. While this may seem low, it is in line with observations from other spoken content retrieval research: over classroom lectures (Chelba et al., 2007), call center recordings (Mamou et al., 2006), and conversational telephone speech (Chia et al., 2010). Moreover, it is broadly consistent with the reported term-matching results for low density languages in MediaEval.

The second challenge lies in the scorability of the system guesses. Table 3 provides an overview of where relevance was found within our collection of babbles. It includes only the subset of babbles for which, during the babble, at least one known relevant document was found at the top of the ranked list. The table presents the number of recognized words—a proxy for the number of potential stopping points—and at how many of those potential stopping points the document ranked in position 1 is known to be relevant, known not to be relevant, or of unknown relevance. Because of the way in which TREC relevance judgments were created, unknown relevance indicates that no TREC system returned the document near the top of their ranked list. At TREC, documents with unknown relevance are typ-

ically scored as if they are not relevant;[5] we make the same assumption.

Table 3 also shows how much we would need to rely on that assumption: the "scorable" fraction for which the relevance of the top-ranked document is known, rather than assumed, ranges from 93 per cent down to 5 per cent. In the averages that we report below, we omit the five babbles with scorable fractions of 30 per cent or less. On average, over the 10 topics for which more than 30 per cent of the potential stopping points are scorable, there are 37 stopping points at which our system could have been scored as successful based on a known relevant document in position 1. In three of these cases, the challenge for our stopping classifier is extreme, with only a handful—between two and seven—of such opportunities.

A third challenge is knowing when to interrupt to present results. The ultimate goal of our work is to predict when the system should interrupt the babbler and barge-in to present an answer in which they might be interested. Table 3 next presents the word positions at which known relevant documents first and last appear in rank one ("First Rel"). This are the earliest and latest scorable successful stopping points. As can be seen, the first possible stopping point exhibits considerable variation, as does the last. For some babbles—babble 274-3, for example—almost any choice of stopping points would be fine. In other cases—babble 258-1, for example—a stopping point prediction would need to be spot on to get any useful results at all. Moreover, we can see both cases in different babbles for the same topic despite the fact that both babblers were prompted by the same topic; for example, babbles 257-1 and 257-3, which are, respectively, fairly easy and fairly hard.

Finally, we can look for interaction effects between speech processing errors and scorability. The rightmost column of Table 3 shows the measured WER for each scorable babble. Of the 10 scorable babbles for which more than 30 per cent of the potential stopping points are scorable, three turned out to be extremely challenging for ASR, with word error rates above 0.7. Overall, however, the WER for the 10 babbles on which we focus is 0.56, which is about the same as the average WER over all 30 babbles.

In addition to the 15 babbles shown in Table 3, there are another 15 babbles for which no relevant document was retrievable. Of those, only a single babble—babble 255-2, at 54 per cent scorable and a WER of 0.402—had more than 30 per cent of the potential stopping points scorable.

5 Learning to Stop

There are several ways in which we could predict when to stop the search and barge-in with an answer—in this paper, we consider a machine learning approach. The idea is that by building a classifier with enough information about known good and bad babbles, a learner can make such predictions better than other methods. Our stopping prediction models uses four types of features for each potential stopping point: the number of words spoken so far, the average word length so far, some "surface characteristics" of those words, and some query performance prediction metrics. The surface characteristics that we used were originally developed to quantify writing style—they are particularly useful for generating readability grades of a given document. Although many metrics for readability have been proposed, we choose a subset: Flesch Reading Ease (Flesch, 1948), Flesch-Kincaid Grade Level (Kincaid et al., 1975), Automated Readability Index (Senter and Smith, 1967), Coleman-Liau index (Coleman and Liau, 1975), Gunning fog index (Gunning, 1968), LIX (Brown and Eskenazi, 2005), and SMOG Grading (McLaughlin, 1969). Our expectation was that a better readability value should correspond to use of words that are more succinct and expressive, and that a larger number of more expressive words should help the search engine to get good responses highly ranked.

As post-retrieval query difficulty prediction measures, we choose three that have been prominent in information retrieval research: clarity (Cronen-Townsend et al., 2002), weighted information gain (Zhou and Croft, 2007), and normalized query commitment (Shtok et al., 2012). Although each takes a distinct approach, the methods all compare some aspect of the documents retrieved by a query

[5] On the assumption that the TREC systems together span the range of responses that are likely to be relevant.

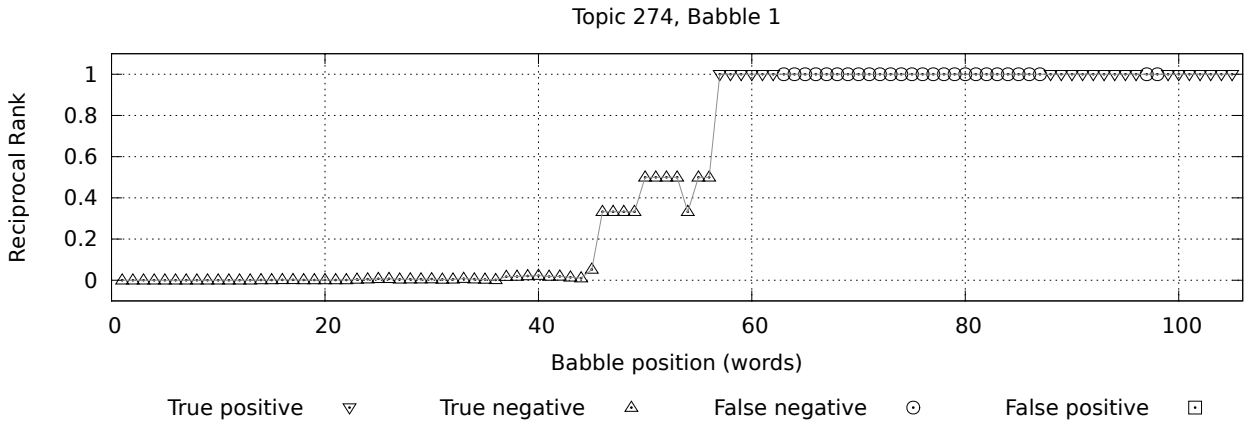

Figure 2: Predictions for babble 274-1 made by a decision tree classifier trained on 27 babbles for the nine other topics. For each point, the mean reciprocal rank is annotated to indicate the correctness of the guess made by the classifier. Note that in this case, the classifier never made a false positive. See Figure 1 for an unannotated version of this same babble.

	Confusion Matrix					
Class.	T_n	F_p	F_n	T_p	F_1	Acy.
Bayes	1288	1259	61	291	0.31	55%
Reg.	2522	25	253	99	0.42	90%
Trees	2499	48	70	282	0.83	96%

Table 4: Cross validation accuracy ("Acy.") measures for stop-prediction classifiers: naive Bayes, logistic regression, and Decision trees.

with the complete collection of documents in the collection from which that retrieval was performed. They seek to provide some measure of information about how likely a query is to have ranked the documents well when relevance judgments are not available. Clarity measures the difference in the language models induced by the retrieved results and the corpus as a whole. Weighted information gain and normalized query commitment look at the scores of the retrieved documents, the former comparing the mean score of the retrieved set with that of the entire corpus; the latter measuring the standard deviation of the scores for the retrieved set.

Features of all four types were were created for each query that was run for each babble; that is after receiving each new word. A separate classifier was then trained for each topic by creating a binary objective function for all 27 babbles for the nine other topics, then using every query for every one of those babbles as training instances. The objective function produces 1 if the query actually retrieved a relevant document at first rank, and 0 otherwise. Figure 2 shows an example of how this training data was created for one babble, and Table 4 shows the resulting hold-one-topic-out cross-validation results for intrinsic measures of classifier accuracy for three Weka classifiers[6]. As can be seen, the decision tree classifier seems to be a good choice, so in Section 7 we compare the stopping prediction model based on a decision tree classifier trained using hold-one-topic-out cross-validation with three baseline models.

6 Evaluation Design

This section describes our evaluation measure and the baselines to which we compared.

6.1 Evaluation Measure

To evaluate a stopping prediction model, the fundamental goal is to stop with a relevant document in rank one, and to do so as close in time as possible to the first such opportunity. If the first guess is bad, it would be reasonable to score a second guess, with some penalty.

Specifically, there are several things that we

[6]Naive Bayes, logistic regression, and decision trees (J48)

would like our evaluation framework to describe. Keeping in mind that ultimately the system will interrupt the speaker to notify them of results, we first want to avoid the interruption before we have found a good answer. Our evaluation measure gives no credit for such a guess. Second, we want to avoid interrupting long after finding the first relevant answer. Credit is reduced with increasing delays after the first point where we could have barged in. Third, when we do barge-in, there must indeed be a good answer in rank one. This will be true if we barge-in at the first opportunity, but if we barge-in later the good answer we had found might have dropped back out of the first position. No credit is given if we barge-in such a case. Finally, if a bad position for first barge-in is chosen, we would like at least to get it right the second time. Thus, we limit ourselves to two tries, awarding half the credit on the second try that we could have received had we barged in at the same point on the first try.

The delay penalty is modeled using an exponential distribution that declines with each new word that arrives after the first opportunity. Let q_0 be the first point within a query where the reciprocal rank is one. Let p_i be the first "yes" guess of the predictor after point q_0. The score is thus $e^{\lambda(q_0-p_i)}$, where λ is the half-life, or the number of words by which the exponential decay has dropped to one-half. The equation is scaled by 0.5 if i is the second element (guess) of p, and by 0.25 if it is the third. From Figure 1, some cases the potential stopping points are consecutive, while in others they are intermittent—we penalize delays from the first good opportunity even when there is no relevant document in position one because we feel that best models the user experience. Unjudged documents in position one are treated as non-relevant.

6.2 Stopping Prediction Baselines

We chose one deterministic and one random baseline for comparison. The deterministic baseline made its first guess at a calculated point in the babble, and continued to guess at each word thereafter. The initial guess was determined by taking the average of the first scorable point of the other 27 out-of-topic babbles.

The random baseline drew the first and second words at which to guess "yes" as samples from a uniform distribution. Specifically, drawing samples uniformly, without replacement, across the average number of words in all other out-of-topic babbles.

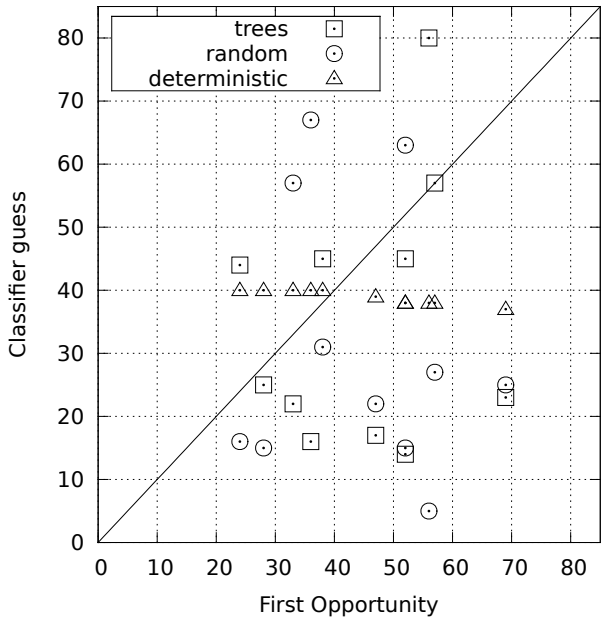

Figure 3: First guesses for various classifiers plotted against the first instance of rank one documents within a babble. Points below the diagonal are places where the classifier guessed too early; points above are guesses too late. All 11 babbles for which the decision tree classifier made a guess are shown.

7 Results

Figure 3 shows the extent to which each classifiers first guess is early, on time, or late. These points falls, respectively, below the main diagonal, on the main diagonal, or above the main diagonal. Early guesses result in large penalties from our scoring function, dropping the maximum score from 1.0 to 0.5; for late guesses the penalty depends on how late the guess is. As can be seen, our decision tree classifier ("trees") guesses early more often than it guesses late. For an additional four cases (not plotted), the decision tree classifier never makes a guess.

Figure 4 shows the results for scoring at most three guesses. These results are averaged over all eleven babbles for which the decision tree classifier made at least one guess; no guess was made on babbles 257-3, 266-2, 260-3, or 274-3. These re-

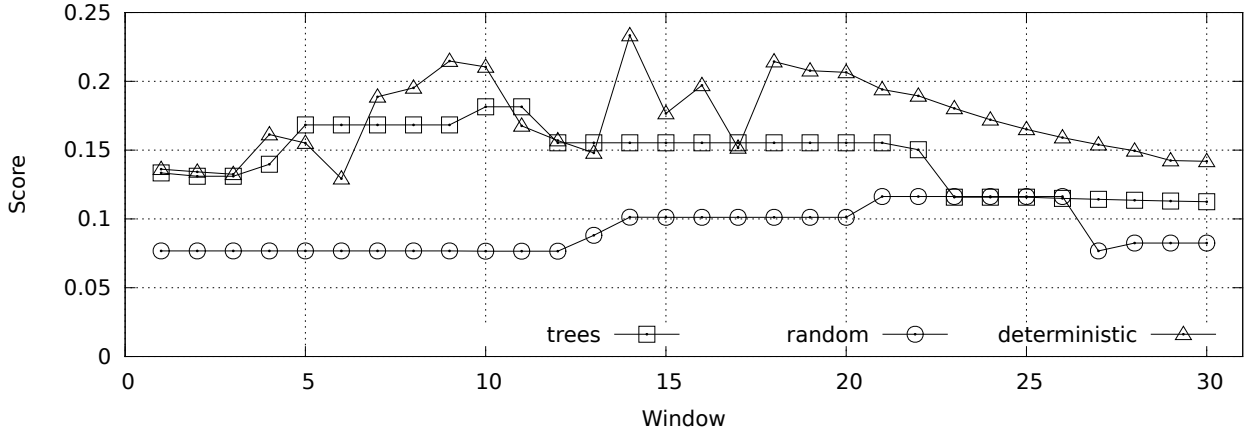

Figure 4: Evaluation using all available babbles in which the tree classifier made a guess.

sults are shown for a half-life of five words, which is a relatively steep penalty function, essentially removing all credit after about ten seconds at normal speaking rates. The leftmost point in each figure, plotted at a "window size" of one, shows the results for the stopping prediction models as we have described them. It is possible, and indeed not unusual, for our decision tree classifier to make two or three guesses in a row, however, in part because it has no feature telling it how long is has been since its most recent guess. To see whether adding a bit of patience would help, we added a deterministic period following each guess in which no additional guess would be allowed. We call the point at which this delay expires, and a guess is again allowed, the delay "window."

As can be seen, a window size of ten or eleven—allowing the next guess no sooner than the tenth or eleventh subsequent word—is optimal for the decision tree classifier when averaged over these eleven babbles. The random classifier has an optimal point between window sizes of 21 and 26, but is generally not as good as the other classifiers. The deterministic classifier displays the most variability, but for window sizes greater than 14, it is the best solution. Although it has fewer features available to it—knowing only the mean number of words to the first opportunity for other topics—it is able to outperform the decision tree classifier for relatively large window sizes.

From this analysis we conclude that our decision tree classifier shows promise; and that going forward, it would likely be beneficial to integrate features of the deterministic classifier. We can also conclude that these results are, at best, suggestive—a richer test collection will ultimately be required. Moreover, we need some approach to accommodate the four cases in which the decision tree classifier never guesses. Setting a maximum point at which the first guess will be tried could be a useful initial heuristic, and one that would be reasonable to apply in practice.

8 Conclusions and Future Work

We have used a simulation study to show that building a system for query by babbling is feasible. Moreover, we have suggested a reasonable evaluation measure for this task, and we have shown that several simple baselines for predicting stopping points can be beaten by a decision tree classifier. Our next step is to try these same techniques with spoken questions and spoken answers in a low-resource language using the test collection that is being developed for the MediaEval 2013 Question Answering for the Spoken Web task.

Another potentially productive direction for future work would be to somehow filter the queries in ways that improve the rankings. Many potential users of this technology in the actual developing region settings that we wish to ultimately serve will likely have no experience with Internet search engines, and thus they may be even less likely to fo-

cus their babbles on useful terms to the same extent that our babblers did in these experiments. There has been some work on techniques for recognizing useful query terms in long queries, but of course we will need to do that with spoken queries, and moreover with queries spoken in a language for which we have at lest limited speech processing capabilities available. How best to model such a situation in a simulation study is not yet clear, so we have deferred this question until the MediaEval speech-to-speech test collection becomes available.

In the long term, many of the questions we are exploring will also has implications for open-domain Web search in other hands- or eyes-free applications such as driving a car or operating an aircraft.

Acknowledgments

We thank Anna Shtok for her assistance with the understanding and implementation of the various query prediction metrics. We also thank the anonymous babblers who provided data that was imperative to this study. Finally, we would like to thank the reviewers, whose comments helped to improve the work overall.

References

[Agarwal et al.2010] Sheetal K. Agarwal, Anupam Jain, Arun Kumar, Amit A. Nanavati, and Nitendra Rajput. 2010. The spoken web: A web for the underprivileged. *SIGWEB Newsletter*, pages 1:1–1:9, June.

[Brown and Eskenazi2005] Jonathan Brown and Maxine Eskenazi. 2005. Student, text and curriculum modeling for reader-specific document retrieval. In *Proceedings of the IASTED International Conference on Human-Computer Interaction. Phoenix, AZ*.

[Chelba et al.2007] Ciprian Chelba, Jorge Silva, and Alex Acero. 2007. Soft indexing of speech content for search in spoken documents. *Computer Speech and Language*, 21(3):458–478.

[Chia et al.2010] Tee Kiah Chia, Khe Chai Sim, Haizhou Li, and Hwee Tou Ng. 2010. Statistical lattice-based spoken document retrieval. *ACM Transactions on Information Systems*, 28(1):2:1–2:30, January.

[Coleman and Liau1975] Meri Coleman and TL Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

[Cronen-Townsend et al.2002] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 299–306, New York, NY, USA. ACM.

[Flesch1948] Rudolf Flesch. 1948. A new readability yardstick. *The Journal of applied psychology*, 32(3):221.

[Gunning1968] Robert Gunning. 1968. *The technique of clear writing*. McGraw-Hill New York.

[Kincaid et al.1975] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, DTIC Document.

[Krovetz1993] Robert Krovetz. 1993. Viewing morphology as an inference process. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '93, pages 191–202, New York, NY, USA. ACM.

[Mamou et al.2006] Jonathan Mamou, David Carmel, and Ron Hoory. 2006. Spoken document retrieval from call-center conversations. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 51–58, New York, NY, USA. ACM.

[McLaughlin1969] G Harry McLaughlin. 1969. Smog grading: A new readability formula. *Journal of reading*, 12(8):639–646.

[Medhi et al.2011] Indrani Medhi, Somani Patnaik, Emma Brunskill, S.N. Nagasena Gautama, William Thies, and Kentaro Toyama. 2011. Designing mobile interfaces for novice and low-literacy users. *ACM Transactions on Computer-Human Interaction*, 18(1):2:1–2:28.

[Metze et al.2012] Florian Metze, Etienne Barnard, Marelie Davel, Charl Van Heerden, Xavier Anguera, Guillaume Gravier, Nitendra Rajput, et al. 2012. The spoken web search task. In *Working Notes Proceedings of the MediaEval 2012 Workshop*.

[Mudliar et al.2012] Preeti Mudliar, Jonathan Donner, and William Thies. 2012. Emergent practices around cgnet swara, voice forum for citizen journalism in rural india. In *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development*, ICTD '12, pages 159–168, New York, NY, USA. ACM.

[Oard2012] Douglas W. Oard. 2012. Query by babbling. In *CIKM Workshop on Information and Knowledge Management for Developing Regions*, October.

[Senter and Smith1967] RJ Senter and EA Smith. 1967. Automated readability index. Technical report, DTIC Document.

[Sherwani et al.2009] Jahanzeb Sherwani, Sooraj Palijo, Sarwat Mirza, Tanveer Ahmed, Nosheen Ali, and Roni Rosenfeld. 2009. Speech vs. touch-tone: Telephony interfaces for information access by low literate users. In *International Conference on Information and Communication Technologies and Development*, pages 447–457.

[Shtok et al.2012] Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. 2012. Predicting query performance by query-drift estimation. *ACM Transactions on Information Systems*, 30(2):11:1–11:35, May.

[Strohman et al.2004] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. 2004. Indri: A language model-based search engine for complex queries. In *International Conference on Intelligence Analysis*.

[Zhou and Croft2007] Yun Zhou and W. Bruce Croft. 2007. Query performance prediction in web search environments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 543–550, New York, NY, USA. ACM.