Detecting Retries of Voice Search Queries

Rivka Levitan
Columbia University[*]
rlevitan@cs.columbia.edu

David Elson
Google Inc.
elson@google.com

Abstract

When a system fails to correctly recognize a voice search query, the user will frequently retry the query, either by repeating it exactly or rephrasing it in an attempt to adapt to the system's failure. It is desirable to be able to identify queries as retries both offline, as a valuable quality signal, and online, as contextual information that can aid recognition. We present a method than can identify retries offline with 81% accuracy using similarity measures between two subsequent queries as well as system and user signals of recognition accuracy. The retry rate predicted by this method correlates significantly with a gold standard measure of accuracy, suggesting that it may be useful as an offline predictor of accuracy.

1 Introduction

With ever more capable smartphones connecting users to cloud-based computing, voice has been a rapidly growing modality for searching for information online. Our voice search application connects a speech recognition service with a search engine, providing users with structured answers to questions, Web results, voice actions such as setting an alarm, and more. In the multimodal smartphone interface, users can press a button to activate the microphone, and then speak the query when prompted by a beep; after receiving results, the microphone button is available if they wish to follow up with a subsequent voice query.

Traditionally, the evaluation of speech recognition systems has been carried by preparing a test set of annotated utterances and comparing the accuracy of a system's transcripts of those utterances against the annotations. In particular, we seek to measure and minimize the word error rate (WER) of a system, with a WER of zero indicating perfect transcription. For voice search interfaces such as the present one, though, query-level metrics like WER only tell part of the story. When a user issues two queries in a row, she might be seeking the same information for a second time due to a system failure the first time. When this happens, from an evaluation standpoint it is helpful to break down why the first query was unsuccessful: it might be a speech recognition issue (in particular, a mistaken transcription), a search quality issue (where a correct transcript is interpreted incorrectly by the semantic understanding systems), a user interface issue, or another factor. As a second voice query may also be a new query or a follow-up query, as opposed to a retry of the first query, the detection of voice search retry pairs in the query steam is non-trivial.

Correctly identifying a retry situation in the query stream has two main benefits. The first involves offline evaluation and monitoring. We would like to know the rate at which users were forced to retry their voice queries, as a measure of quality. The second has a more immediate benefit for individual users: if we can detect in real time that a new voice search is really a retry of a previous voice search, we can take immediate corrective action, such as reranking transcription hypotheses to avoid making the same mistake twice, or presenting alternative searches in the user interface to indicate that the system acknowledges it is having difficulty.

In this paper, we describe a method for the *classification of subsequent voice searches* as either retry pairs of a certain type, or non-retry pairs. We identify four salient types of retry pairs, describe a test set and identify the features we extracted to build an automatic classifier. We then describe the models we used to build the classifier and their rel-

[*] This work was done while the first author was an intern at Google Inc.

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 230–235,
Baltimore, Maryland, USA, June 23-25 2014. ©2014 Association for Computational Linguistics

ative performance on the task, and leave the issue of real-time corrective action to future work.

2 Related Work

Previous work in voice-enabled information retrieval has investigated the problem of identifying voice retries, and some has taken the additional step of taking corrective action in instances where the user is thought to be retrying an earlier utterance. Zweig (2009) describes a system switching approach in which the second utterance is recognized by a separate model, one trained differently than the primary model. The "backup" system is found to be quite effective at recognizing those utterances missed by the primary system. Retry cases are identified with joint language modeling across multiple transcripts, with the intuition that retry pairs tend to be closely related or exact duplicates. They also propose a joint acoustic model in which portions of both utterances are averaged for feature extraction. Zweig et al. (2008) similarly create a joint decoding model under the assumption that a discrete sent of entities (names of businesses with directory information) underlies both queries. While we follow this work in our usage of joint language modeling, our application encompasses open domain voice searches and voice actions (such as placing calls), so we cannot use simplifying domain assumptions.

Other approaches include Cevik, Weng and Lee (2008), who use dynamic time warping to define pattern boundaries using spectral features, and then consider the best matching patterns to be repeated. Williams (2008) measures the overlap between the two utterances' n-best lists (alternate hypotheses) and upweights hypotheses that are common to both attempts; similarly, Orlandi, Culy and Franco (2003) remove hypotheses that are semantically equivalent to a previously rejected hypothesis. Unlike these approaches, we do not assume a strong notion of dialog state to maintain per-state models.

Another consequence of the open-domain nature of our service is that users are conditioned to interact with the system as they would with a search engine, e.g., if the results of a search do not satisfy their information need, they rephrase queries in order to refine their results. This can happen *even if the first transcript was correct* and the rephrased query can be easily confused for a retry of a utterance where the recognition failed. For purposes of latently monitoring the accuracy of the recognizer from usage logs, this is a significant complicating factor.

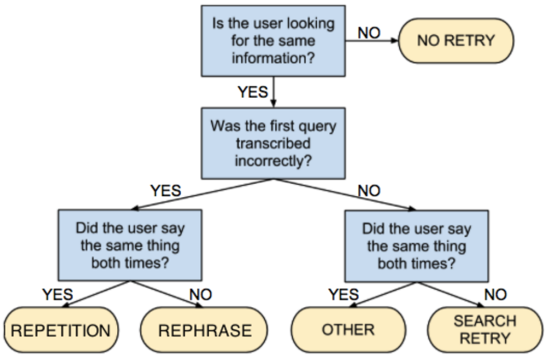

Is the user looking for the same information?
NO
NO RETRY
YES
Was the first query transcribed incorrectly?
YES
NO
Did the user say the same thing both times?
Did the user say the same thing both times?
YES
NO
YES
NO
REPETITION
REPHRASE
OTHER
SEARCH RETRY

Figure 1: Retry annotation decision tree.

3 Data and Annotation

Our data consists of pairs of queries sampled from anonymized session logs. We consider a pair of voice searches (spoken queries) to be a potential retry pair if they are consecutive; we assume that a voice search cannot be a retry of another voice search if a typed search occurs between them. We also exclude pairs for which either member has no recognition result. For the purpose of our analysis, we further restricted our data to query pairs whose second member had been previously randomly selected for transcription. A set of 8,254 query pairs met these requirements and are considered potential retry pairs. 1,000 randomly selected pairs from this set were separated out and annotated by the authors, leaving a test set of 7,254 potential retry pairs. Among the annotated development set, 18 inaudible or unintelligible pairs were discarded, for a final development set of 982 pairs.

The problem as we have formulated it requires a labeling system that identifies repetitions and rephrases as retries, while excluding query pairs that are superficially similar but have different search intents. Our system includes five labels. Figure 1 shows the guidelines for annotation that define each category.

The first distinction is between query pairs with the same *search intent* (”Is the user looking for the same information?”) and those with different search intents. We define search intent as the response the user wants and expects from the system. If the second query's search intent is different, it is by definition **no retry**.

The second distinction we make is between cases where the first query was recognized correctly and those where it was not. Although a query that was recognized correctly may be retried—for example, the user may want to be reminded of information she already received (**other**)—we are only interested in cases where the system is in error.

If the search intent is the same for both queries, and the system incorrectly recognized the first, we consider the second query a retry. We distinguish between cases where the user repeated the query exactly, **repetition**, and where the user rephrased the query in an attempt to adapt to the system's failure, **rephrase**. This category includes many kinds of rephrasings, such as adding or dropping terms, or replacing them with synonyms. The rephrased query may be significantly different from the original, as in the following example:

Q1. Navigate to chaparral ease. ("Navigate to Chiapparelli's.")

Q2. Chipper rally's Little Italy Baltimore. ("Chiapparelli's Little Italy Baltimore.")

The rephrased query dropped a term ("Navigate to") and added another ("Little Italy Baltimore").

This example illustrates another difficulty of the data: the unreliability of the automatic speech recognition (ASR) means that terms that are in fact identical ("Chiapparelli's") may be recognized very differently ("chaparral ease" or "chipper rally's"). In the next example, the recognition hypotheses of two identical queries have only a single word in common:

Q1. I get in the house Google. ("I did it Google")

Q2. I did it crash cool. ("I did it Google")

Conversely, recognition hypotheses that are nearly identical are not necessarily retries. Often, these are "serial queries," a series of queries the user is making of the same form or on the same topic, often to test the system.

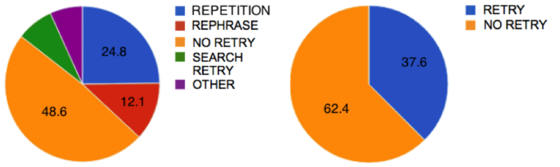
Q1. How tall is George Clooney?
Q2. How old is George Clooney?

Q1. Weather in New York.

Q2. Weather in Los Angeles.

These complementary problems mean that we cannot use naïve text similarity features to identify retries. Instead, we combine features that model the first query's likely accuracy to broader similarity features to form a more nuanced picture of a likely retry.

The five granular retry labels were collapsed into binary categories: search retry, other, and no retry were mapped to NO RETRY; and repetition and rephrase were mapped to RETRY. The label distribution of the final dataset is shown in Figure 2.



(a) Granular labels (b) Collapsed (binary) labels

Figure 2: Retry label distribution.

4 Features

The features we consider can be divided into three main categories. The first group of features, *similarity*, is intended to measure the similarity between the two queries, as similar queries are (with the above caveats) more likely to be retries. We calculate the edit distance between the two transcripts at the character and word level, as well as the two most similar phonetic rewrites. We include both raw and normalized values as features. We also count the number of unigrams the two transcripts have in common and the length, absolute and relative, of the longest unigram overlap.

As we have shown in the previous section, similarity features alone cannot identify a retry, since ASR errors and user rephrases can result in recognition hypotheses that are significantly different from the original query, while a nearly identical pair of queries can have different search intents. Our second group of features, *correctness*, goes up a level in our labeling decision tree (Figure 1) and attempts to instead answer the question: "Was the first query transcribed incorrectly?" We use the confidence score assigned by the recognizer to the first recognition hypothesis as a measure of the system's opinion of its own performance. Since this score, while informative, may be inaccurate, we also consider signals from the user that might indicate the accuracy of the hypothesis. A boolean feature indicates whether the user interacted with any of the results (structured or unstructured) that were presented by the system in response to the first query, which should constitute an implicit acceptance of the system's recognition hypothesis. The length of the interval between the two queries is another feature, since a query that occurs immediately after another is likely to be a retry. We also include the difference and ratio of the two queries' speaking rate, roughly calculated as the number of vowels divided by the audio duration in seconds, since a speaker is likely to hyperarticulate (speak more loudly and slowly) after being misunderstood ((Wade et al., 1992; Oviatt et al., 1996; Levow, 1998; Bell and Gustafson, 1999; Soltau and Waibel, 1998)).

The third feature group, *recognizability*, attempts to model the characteristics of a query that is likely to be misrecognized (for the first query of the pair) or is likely to be a retry of a previous query (for the second query). We look at the language model (LM) score and the number of alternate pronunciations of the first query, predicting that a misrecognized query will have a lower LM score and more alternate pronunciations. In addition, we look at the number of characters and unigrams and the audio duration of each query, with the intuition that the length of a query may be correlated with its likelihood of being retried (or a retry). This feature group also includes two heuristic features intended to flag the "serial queries" mentioned before: the number of capitalized words in each query, and whether each one begins with a question word (who, what, etc.).

5 Prediction task

5.1 Experimental Results

A logistic regression model was trained on these features to predict the collapsed binary categories of NO RETRY (search retry, other, no retry) vs. RETRY (rephrase, repetition). The results of running this model with each combination of the feature groups are shown in Table 1.

Features	Precision	Recall	F1	Accuracy
Similarity	0.54	0.65	0.59	0.72
Correctness	0.53	0.67	0.59	0.73
Recognizability	0.49	0.63	0.55	0.70
Sim. & Corr.	0.67	0.71	0.69	0.77
Sim. & Rec.	0.62	0.70	0.66	0.76
Corr. & Rec.	0.65	0.71	0.68	0.77
All Features	0.70	0.76	0.73	0.81

Table 1: Results of the binary prediction task.

Individually, each feature group peformed significantly better than the baseline strategy of always predicting NO RETRY (62.4%). Each pair of feature groups performed better than any individual group, and the final combination of all three feature groups had the highest precision, recall, and accuracy, suggesting that each aspect of the retry conceptualization provides valuable information to the model.

Of the *similarity* features, the ones that contributed significantly in the final model were character edit distance (normalized) and phoneme edit distance (raw and normalized); as expected, retries are associated with more similar query pairs. Of the *correctness* features, high recognizer confidence, the presence of a positive reaction from the user such as a link click, and a long interval between queries were all negatively associated with retries. The significant *recognizability* features included length of the first query in characters (longer queries were less likely to be retried) and the number of capital letters in each query (as our LM is case-sensitive): queries transcribed with more capital letters were more likely to be retried, but less likely to themselves be retries. In addition, the language model likelihood for the first query was, as expected, significantly lower for retries. Interestingly, the score of the *second* query was lower for retries as well. This accords with our finding that retries of misrecognized queries are themselves misrecognized 60%-70% of the time, which highlights the potential value of corrective action informed by the retry context.

Several features, though not significant in the model, are significantly different between the RETRY and NO RETRY categories, which affords us further insight into the characteristics of a retry. T-tests between the two categories showed that all edit distance features—character, word, reduced, and phonetic; raw and normalized—are significantly more similar between retry query pairs.[1] Similarly, the number of unigrams the two queries have in common is significantly higher for retries. The duration of each member of the query pair, in seconds and word count, is significantly more similar between retry pairs, and each member of a retry pair tends to be shorter than members of a no retry pair. Finally, members of NO RETRY query pairs were significantly more similar in speaking rate, and the relative speaking rate of the second query was significantly *slower* for RETRY pairs, possibly due to hyperarticulation.

5.2 Analysis

Figure 3 shows a breakdown of the true granular labels versus the predicted binary labels. The primary source of error is the REPHRASE category, which is identified as a retry with only 16.5% ac-

[1] T-tests reported here use a conservative significance threshold of $p < 0.00125$ to control for family-wise type I error ("data dredging" effects).

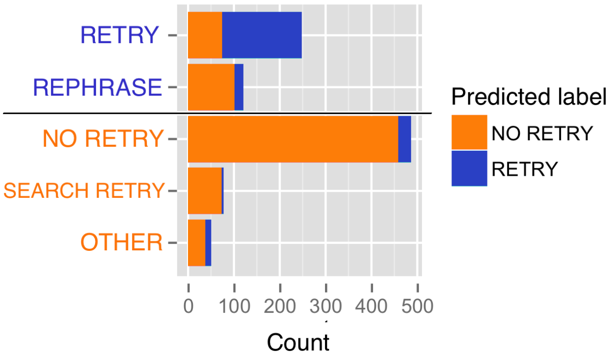


Figure 3: Performance on each of the granular categories.

curacy. This result reflects the fact that although rephrases conceptually belong in the retry category, their characteristics are materially different. Most notably, all edit distance features are significantly greater for rephrases. Differences in duration between the two queries in a pair, in seconds and words, are significantly greater as well. Rephrases also are significantly longer, in seconds and words, than strict retries. The model including only correctness and recognizability features does significantly better on rephrases than the full model, identifying them as retries with 25.6% accuracy, confirming that the similarity features are the primary culprit. Future work may address this issue by including features crafted to examine the similarity between substrings of the two queries, rather than the query as a whole, and by expanding the similarity definition to include synonyms.

To test the model's performance with a larger, unseen dataset, we looked at how many retries it detected in the test set of potential retry pairs (n=7,254). We do not have retry annotations for this larger set, but we have transcriptions for the first member of each query pair, enabling us to calculate the word error rate (WER) of each query's recognition hypothesis, and thus obtain ground truth for half of our retry definition. A perfect model should never predict RETRY when the first query is transcribed correctly (WER==0). As shown in Figure 4, our model assigns a RETRY label to approximately 14% of the queries following an incorrectly recognized search, and only 2% of queries following a correctly recognized search. While this provides us with only a lower bound on our model's error, this significant correlation with an orthogonal accuracy metric shows that we have modeled at least this aspect of retries correctly, and suggests a correlation between retry rate and traditional WER-based evaluation.

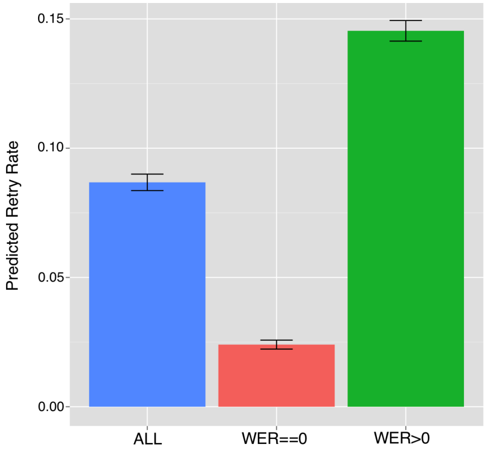


Figure 4: Performance on unseen data. A perfect model would have a predicted retry rate of 0 when WER==0.

6 Conclusion

We have presented a method for characterizing retries in an unrestricted voice interface to a search system. One particular challenge is the lack of simplifying assumptions based on domain and state (as users may consider the system to be stateless when issuing subsequent queries). We introduce a labeling scheme for retries that encompasses rephrases—cases in which the user reworded her query to adapt to the system's error—as well as repetitions.

Our model identifies retries with 81% accuracy, significantly above baseline. Our error analysis confirms that user rephrasings complicate the binary class separation; an approach that models typical *typed* rephrasings may help overcome this difficulty. However, our model's performance today correlates strongly with an orthogonal accuracy metric, word error rate, on unseen data. This suggests that "retry rate" is a reasonable offline quality metric, to be considered in context among other metrics and traditional evaluation based on word error rate.

Acknowledgments

The authors thank Daisy Stanton and Maryam Kamvar for their helpful comments on this project.

References

Linda Bell and Joakim Gustafson. 1999. Repetition and its phonetic realizations: Investigating a swedish database of spontaneous computer-directed speech. In *Proceedings of ICPhS*, volume 99, pages 1221–1224.

Mert Cevik, Fuliang Weng, and Chin-Hui Lee. 2008. Detection of repetitions in spontaneous speech in dialogue sessions. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH 2008)*, pages 471–474, Brisbane, Australia.

Gina-Anne Levow. 1998. Characterizing and recognizing spoken corrections in human-computer dialogue. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 736–742. Association for Computational Linguistics.

Marco Orlandi, Christopher Culy, and Horacio Franco. 2003. Using dialog corrections to improve speech recognition. In *Error Handling in Spoken Language Dialogue Systems*. International Speech Communication Association.

Sharon Oviatt, G-A Levow, Margaret MacEachern, and Karen Kuhn. 1996. Modeling hyperarticulate speech during human-computer error resolution. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 2, pages 801–804. IEEE.

Hagen Soltau and Alex Waibel. 1998. On the influence of hyperarticulated speech on recognition performance. In *ICSLP*. Citeseer.

Elizabeth Wade, Elizabeth Shriberg, and Patti Price. 1992. User behaviors affecting speech recognition. In *ICSLP*.

Jason D. Williams. 2008. Exploiting the asr n-best by tracking multiple dialog state hypotheses. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH 2008)*, pages 191–194, Brisbane, Australia.

Geoffrey Zweig, Dan Bohus, Xiao Li, and Patrick Nguyen. 2008. Structured models for joint decoding of repeated utterances. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH 2008)*, pages 1157–1160, Brisbane, Australia.

Geoffrey Zweig. 2009. New methods for the analysis of repeated utterances. In *Proceedings of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH 2009)*, pages 2791–2794, Brighton, United Kingdom.