Did You Say U2 or YouTube?

Inferring Implicit Transcripts from Voice Search Logs

Milad Shokouhi
Microsoft
milads@microsoft.com

Umut Ozertem
Microsoft
umuto@microsoft.com

Nick Craswell
Microsoft
nickcr@microsoft.com

ABSTRACT

Web search via voice is becoming increasingly popular, taking advantage of recent advances in automatic speech recognition. Speech recognition systems are trained using audio transcripts, which can be generated by a paid annotator listening to some audio and manually transcribing it. This paper considers an alternative source of training data for speech recognition, called implicit transcription. This is based on Web search clicks and reformulations, which can be interpreted as validating or correcting the recognition done during a real Web search. This can give a large amount of free training data that matches the exact characteristics of real incoming voice searches and the implicit transcriptions can better reflect the needs of real users because they come from the user who generated the audio. On an overall basis we demonstrate that the new training data has value in improving speech recognition. We further show that the in-context feedback from real users can allow the speech recognizer to exploit contextual signals, and reduce the recognition error rate further by up to 23%.

Categories and Subject Descriptors

Information Systems [**Information Retrieval**]: Information retrieval query processing

Keywords

Speech retrieval, speech recognition, personalized search

1. INTRODUCTION

Automatic speech recognition (ASR) systems have become substantially more accurate in recent years. For instance, in May 2015 Google announced an 8% error rate for word recognition, down from 23% in 2013,[1] mostly thanks to breakthroughs in deep learning and availability of large-scale training data and infrastructure.

The improved quality of ASR along with widespread adoption of mobile devices for daily tasks have significantly boosted the

---

[1] http://bit.ly/1JbBnax

number of search queries by voice. In October 2014, Google official blog reported that 55% of teens and 41% of adults use voice search at least once a day.[2] In an interview with Wall Street Journal in November 2014, Baidu Chief Scientist Andrew Ng revealed that 10% of their queries come through speech.[3] He also anticipated that by 2019 half of queries will be on speech or images.[4]

Speech recognition models are trained with large samples of speech utterances and their ground-truth transcripts. These transcripts are often generated by human annotators that listen to the utterance audio and transcribe it manually. Human transcripts are expensive to obtain and are subject to various biases and noise. For instance, the audio may not be always understandable due to background noise or speaker's pronunciation. In addition, there might be several plausible spellings and interpretations of an utterance which can make the task of finding the *correct* user intent challenging – if not impossible sometimes (e.g. `pictures of whales` and `pictures of wales` sound almost identical in spite of their different meaning).

In this paper, we propose a new technique for mining utterance-transcript pairs from users implicit feedback recorded in search logs. The labels mined by our approach can be collected at large-scale, and we show that they can be effective in training ASR models. Furthermore, our *implicit transcripts* capture valuable contextual information that is often missing from manually transcribed data.

To generate our implicit transcripts, we consider the output of recognizer for all *successful* voice queries as ground-truth. A query is defined as successful if it has at least one satisfied (SAT) click by the user. Fox et al. [8] referred to clicks with dwell times $\geq$ 30 seconds as SAT and showed that such clicks are highly correlated with search satisfaction. Suppose that a user submits `restaurants in cambridge` as query, the recognizer correctly transcribes the query, and the user responds by a SAT click on one of the search results. In such a case, we assign `restaurants in cambridge` as implicit transcript for the submitted voice query. Now consider an alternative scenario in which the recognizer misrecognizes `restaurants` as `restaurant`. The user is likely to be presented with near-identical set of results and as long as he/she responds with a SAT click, we will still consider the overall interaction as *positive* and regard the output of recognizer as implicit transcript for the utterance.

The implicit transcripts mined from SAT searches do not capture unsuccessful cases in which the ASR engine failed to recognize the correct query from the user's utterance. Consequently, training an ASR model with such labels is unlikely to lead to any gains on top of the default recognition engine. To remedy this issue, we

---

[2] http://bit.ly/1V4XBj6
[3] http://on.wsj.com/1xLxmEc
[4] http://bit.ly/1uS8k3J

Table 1: A real search session example from Bing logs on February 2nd, 2015. The second column shows the user input and the third column indicates the input type. The submitted query is shown on column four and the last column lists the top five suggestions in n-best list (empty for typed queries). The user keeps reformulating his/her query by voice until eventually switches to keyboard for typing. The underline symbol indicates cases where a *rejected* suggestion appears in the n-best lists of later queries. The **bold** text represents cases where a typed query submitted later appears in the n-best lists of previous voice queries. We use large font size to visualize the increasing emphasis (e.g. pause and stress) that was put on each word by the user which could be heard from listening to the recorded audio of voice queries.

Timestamp	User Query	Input Type	Submitted Query	n-best
3:14:19 PM	different kinds of graphs	🎤	different kind of girl	"different kind of girl", "different kind of grass", "define kind of girl", "different kind of girls","different kind of growth"
3:14:25 PM	different kinds of graphs	🎤	different kinds of grass	"different kinds of grass","different kind of grass","different kinds of girls", "different kind of girl","different kinds of girl"
3:14:33 PM	different kinds of graphs	🎤	different kinds of grass	"different kinds of grass",**"different kinds of graphs"**,"different kind of grass","different kinds of graf","different kinds of groff"
3:14:45 PM	different kinds of graphs	⌨	different kinds of graphs	–

rely on *voice-to-text* (V2T) reformulations in search logs for mining *negative* examples. Previous work suggest that such reformulations are correlated with users' dissatisfaction about the output of speech recognizer [15, 19, 32]. Hence, to mine our implicit transcripts we also focus on cases where a user initially issues a search query by voice but eventually switches to keyboard for *correction*. We use the final query submitted by keyboard as the implicit transcript of previous unsuccessful voice queries in the same *session*.[5]

Table 1 illustrates a real example of V2T reformulation sampled from the query logs of Bing search engine on February 2nd, 2015. The first column contains the timestamps recorded for submitted queries. The second column presents the *actual* query submitted by the user. The third column specifies the input type (speech vs. keyboard); the fourth column contains the query that is received by the search backend (recognizer's output for voice queries), and the last column includes the n-best candidates (available only for voice queries). We vary the font-size to indicate the loudness and stress put on each word by the user which could be heard from listening to the audio. The search session starts when the user issues the query `different kinds of graphs` by voice. The speech recognizer misrecognizes the query as `different kind of girl` and issues the wrong query to the backend. The user – clearly dissatisfied with the results – repeats the voice query this time putting more emphasis on `graphs` which was misrecognized as `girl` on the first attempt. The voice query is misrecognized again, this time to `different kind of grass`. Please note that the top candidates in the n-best list returned by the recognizer contain `different kind of girl`, regardless of the fact that it was implicitly *rejected* by the user in the previous query. The user continues by repeating the voice query this time louder and clearly frustrated but the recognizer fails to correctly recognize the query yet again. It misrecognizes the query as `different kinds of grass` once more, despite the previous unsuccessful (*abandoned*) query by the user. Interestingly, the correct query (`different kinds of graphs`) does appear in the n-best list at position two. It comes second only to `different kinds of grass`, a candidate which was already implicitly rejected by the user. After 26

seconds of unsuccessful interactions, the user switches to keyboard and types `different kinds of graphs`. It is clear in this example that the final query typed by the user could be used as the transcript for previous failed voice queries in the session. It is also evident that the user interactions during the first two queries could perhaps be used to rank the *correct* suggestion in n-best on top.

In this paper we aim to learn from positive and negative user interactions recorded in voice search logs to mine *implicit transcripts* that can be used to train ASR models for voice queries (*first contribution*). We evaluate the effectiveness of models using both manual and implicit transcripts and show that they can significantly improve the accuracy of speech recognition for voice queries mainly thanks to availability of large-scale training data (*second contribution*). Furthermore, our experiments suggest that analogous to training personalized search rankers by implicit click labels [8], we can deploy the implicit transcripts mined from search logs to personalize the speech recognition output for voice search queries and improve the recognition accuracy further by up to 23% (*third contribution*).

We continue by presenting a brief background on speech recognition before introducing our approach in the following sections.

2. SPEECH RECOGNITION

Speech recognition systems aim to find the most likely word sequence for an input speech waveform. For large-vocabulary *continuous* (multi-word) recognition scenarios such as voice search this is often formalized as follows [9, 25],

$$\hat{\omega} = \operatorname*{argmax}_{!}\{P(\omega|o)\} = \operatorname*{argmax}_{!}\{\frac{P(o|\omega)P(\omega)}{P(o)}\}$$

where, ω represents a sequence of words (of any length), and o denotes the input audio typically computed over 10-25 millisecond windows known as *frames*. $P(o|\omega)$ is commonly referred to as *acoustic model* and represents the likelihood of utterance o given the word sequence ω. The acoustic model specifies the relationship between the audio signal and the basic units of speech which are usually *phones*.[6] The acoustic models are trained over large sets of audio data and their corresponding transcriptions. The second term $P(\omega)$ is generally called *language model* and represents the prior

---

[5] For simplicity, we follow the common approach of drawing session boundaries after observing intervals of inactivity longer than 30 minutes [4] although recent work [11] that suggests such boundaries may not be optimal.

[6] There are about 40 phones in English. For example the phonetic alphabet representation of 'cat' is /k/ /æ/ /t/.
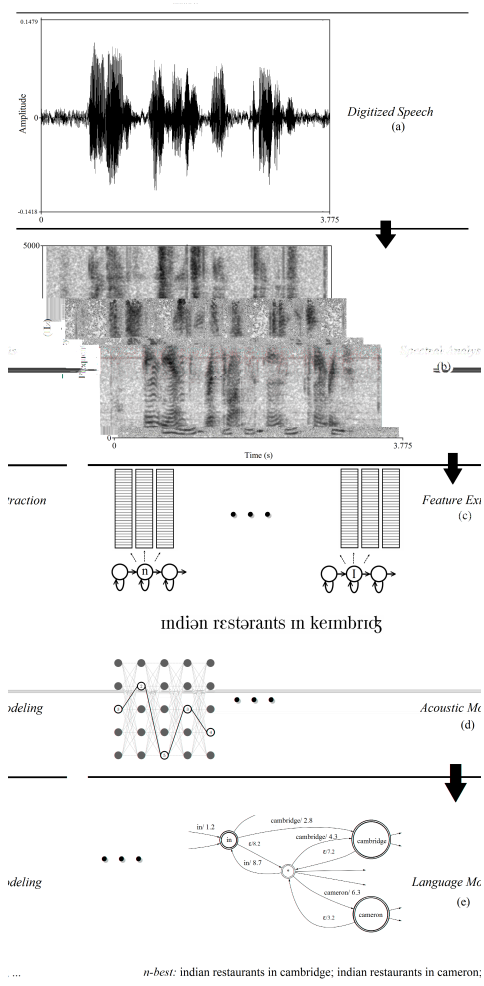
Figure 1: An overview of speech recognition process. (a) The user input is received in form of a speech waveform. (b) The digitalized speech signal undergoes spectral analysis and segmentation. (c) Each audio frame is represented by a set of features such as Mel-frequency cepstral coefficients. (d) Acoustic models map the acoustic features to the most likely phone sequence and concatenate phone units to form words. (e) Language models are used to find the most likely sequence of words and generate the n-best list.

probability of the word sequence ω. $P(\omega)$ is usually generated using statistical language models based on N-grams that are computed over large collections from documents and other sources. $P(o)$ is a constant and is ignored as it does not affect the optimization.

A simplified[7] overview of speech recognition process is depicted in Figure 1. The user's utterance (query in our case) is received as a speech waveform by the recognizer. The waveform is split into *frames* often computed every 10-25 milliseconds (ms) using an overlapping window of 5-10 ms [9]. These features are usually generated based on mel-frequency cepstral coefficients (MFCCs) [7] by applying Fast Fourier transforms to the signal. Acoustic models are commonly trained based on Hidden Markov Models [26], or more recently based on deep neural nets [14]. In Figure 1, 'ɪndiən rɛstərants ɪn keɪmbrɪdʒ' is the phonetic representation of the submitted voice query (`indian restaurants in`

---

[7]We ignore several common components such as pronunciation modeling and speaker adaptation for brevity.

`cambridge`). The most likely sequence estimated based on the acoustic model are identified using Viterbi algorithm. Finally, statistical language models – possibly generated over different sources – are used to compute the probability of each word sequence [18]. The recognizer outputs a list of candidates in descending order of $P(o|\omega)P(\omega)$ score, which is generally referred to as n-best list. In this paper, we demonstrate how we can leverage users' implicit feedback to learn models that *rerank* the n-best candidates, and produce more accurate speech recognition output.

Evaluation metrics. Speech recognition systems are typically evaluated at the word level and their performance is measured by word error rate (WER) [23]. WER measures the number of deletions ($\mathcal{D}$), substitutions ($\mathcal{S}$) and insertions ($\mathcal{I}$) that are required to convert the recognition output to the reference transcript. That is,

$$WER = \frac{\mathcal{D} + \mathcal{S} + \mathcal{I}}{\mathcal{L}}$$

where, the number of words in the reference is specified by $\mathcal{L}$. WER is essentially the length normalized Levenshtein edit-distance [20] value between the recognizer's output (top suggestion in n-best) and the ground-truth transcript, and measures the accuracy of speech recognition in isolation and independent of retrieval quality.

While speech recognition and document retrieval quality have been reported to be correlated [2, 6] there are certain recognition errors such as missing common stopwords and misrecognition of plural forms that tend to have minor effect on retrieval quality. To address this shortcoming, WebScore [28] has been suggested as an alternative metric that focuses mostly on user satisfaction. WebScore compares the top k results returned for the recognized and correct form of the query, and is originally defined as "how many times the search result as queried by the recognition hypothesis varies from the search result as queried by a human transcription" [28]. We deploy a normalized version of WebScore that measures the rate of overlap in the first page of results returned for the recognition hypothesis and transcript (#common URLs / #unique URLs). A *top-heavy* variant of WebScore that takes the position of URLs into account may be more appropriate for Web search scenarios. However, we report the results for the original *unweighted* version to be consistent with previous approaches in literature.

Both WebScore and WER only consider the recognition hypothesis (top suggestion in n-best) for evaluation. This is motivated by the fact that the top suggestion is what will be issued as query and unless it is correctly recognized, the retrieval quality is likely to be poor. However, for the purpose of ranker training, a system that ranks the correct suggestion higher even when it is not not on top (e.g. second position) should be penalized less heavily than a system that ranks it lower (e.g. position five). Therefore, we also measure the mean reciprocal rank (MRR) of the correct suggestion in n-best lists. The reciprocal rank is set to zero when the correct suggestion (transcript) does not appear in n-best list.

In the next section we describe how user's implicit feedback can be turned into training labels for optimizing such models.

3. IMPLICIT TRANSCRIPTS

Speech recognition systems depend on transcriptions of audio data for training, validation and testing. The transcriptions usually come from a manual process, involving a paid human transcriber listening to an audio utterance and producing a text transcript. This allows us to generate transcripts for any available utterance – potentially multiple times from different transcribers if needed. However, since transcribers are paid, collecting manual transcripts at large

scale is costly. The manual transcription process can also introduce noise and bias, where the transcript differs from what the original user had in mind. This could be because noise in the user's environment or their accent makes the audio difficult to understand. Even for a clear utterance, homophones such as `wales`/`whales` and `phish`/`fish` can make it impossible for the transcriber to be sure of the user's intent. Another source of mismatch is the world knowledge of the transcriber, particularly in transcribing named entities. In one case, five independent transcribers gave us the same transcript `hagie lee's bakery` for an utterance, but based on the user's follow-up behavior and clicks it is clear they were looking for Haegele's Bakery. Most transcribers do not know about Philadelphia bakeries, which makes it difficult for them to understand the true user intent.

We propose the generation of implicit transcripts based on user interaction logs. Unlike manual transcripts, implicit transcripts do not require us to pay a human transcriber, but instead are based on an engine's existing logs. Positive user interaction with the results can suggest that the output of recognizer has been acceptable for the user. Otherwise users may indicate their dissatisfaction by reformulating [12], which provides an alternate transcript.

Since the voice utterance and implicit behavior come from the same user, the implicit behavior offers evidence of the user's true intent. Even if the audio is too unclear for a manual transcription, the real user still knows their intent and can provide feedback. For homophones, which are indistinguishable to manual transcribers, the implicit transcript can uncover the truth. For example, for a given utterance recognized as `pictures of whales`, a SAT click on a page about 'blue whale photos' can be regarded as ASR success, while a V2T reformulation to `pictures of wales` may suggest the opposite. This allows us to collect reliable learning targets for queries that have homophones. Similarly we benefit from the user's world knowledge, when their behavior indicates that they were saying `haegele's bakery`.

Positive labels. Previous work [8] suggests that SAT clicks can be regarded as a sign of user satisfaction with search results. For voice queries, we conjecture that satisfaction with search results indicates that the output of speech recognizer must have been *acceptable*. We note that the recognition output may not exactly match the user's utterance. There might be lexical differences that still lead to good search results. However, in general the user is unlikely to follow up with a SAT click unless the recognition output is semantically correct [13]. Therefore, for all the voice queries that received at least one SAT click in the logs, we regard the output of recognizer as an implicit transcript of the utterance.

Negative labels. After issuing a voice query, if the user does not interact with the search results and issues a text query in a short time frame (within same session), this is evidence that the user was dissatisfied with results of the first query [13]. If the second query led to a SAT click and also was on the n-best ASR candidates for the first query, we take this as evidence that the user has resorted to the keyboard to tell us their true intent. We use this second typed query as an implicit transcript for the first (voice) query. Had the ASR system ranked this candidate at the top of its n-best, the user's SAT click would have been possible without reformulation.

Negative labels can also give us transcripts outside the n-best of the voice query, and even include words that are outside the vocabulary of the ASR system. However, in this paper we limit the labeling to reformulations in the n-best, to keep the learning as a reranking problem, which will be detailed in the following section.

Table 2: The quality of implicit transcripts for positive and negative labels. The ground-truth transcripts are obtained manually by crowdsourcing.

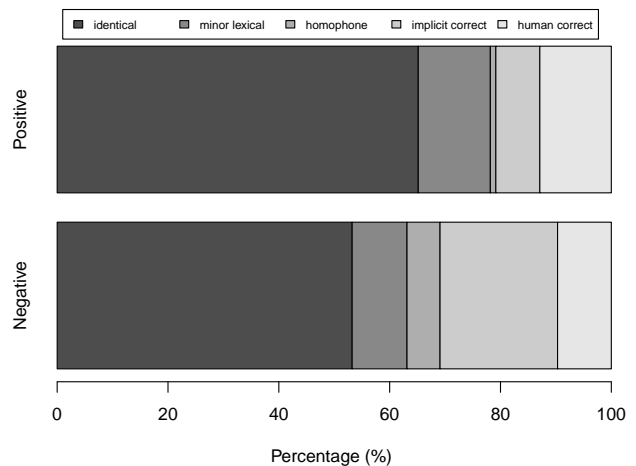| Dataset		WER	WebScore
	all data	**0.11**	**0.80**
	identical	0.00	1.00
positive	minor lexical	0.23	0.63
labels	homophone	0.43	0.24
	implicit correct	0.35	0.29
	human correct	0.37	0.29
	all data	**0.20**	**0.65**
	identical	0	1
negative	minor lexical	0.31	0.55
labels	homophone	0.52	0.09
	implicit correct	0.47	0.11
	human correct	0.35	0.27

Figure 2: The categorization of positive and negative implicit labels when compared against human transcripts.

Quality of implicit transcripts. To evaluate the quality of our implicit transcripts, we collected a random sample of voice queries (*impressions*) submitted to Bing search engine during November 2014 and transcribed them implicitly. We then subsample 1000 queries with positive and another of the same size with negative labels. The audio recorded for each of these queries was presented to professional annotators from Clickworker[8] – a crowdsourcing partner with Microsoft. The annotators were instructed to listen to the audio and transcribe it. Each voice query was transcribed by five annotators independently and the consensus obtained from crowdsourcing were then compared to our implicit transcripts mined from the logs. For the positive label dataset, all manual judges agree for 73% of queries, and for the negative labels this drops to 58%. It is not surprising that the negative labels yield a harder dataset. Positive labels contain many navigational queries, whereas negative samples are coming from queries where the production speech recognizer has difficulty.

To compare human and implicit transcriptions we start with measuring WER and WebScore of implicit transcriptions, using the human transcriptions as ground-truth. Since there are cases where implicit transcripts can actually be better than manual transcripts, we

---

[8]http://www.clickworker.com

also manually annotate the 2000 pairs of transcripts, to understand the different agreement and disagreement cases and how often they occur. The results are presented in Table 2 and Figure 2.

WER and WebScore for positive and negative labels (bolded lines in Table 2) reveal that there is some disagreement between the consensus of human transcribers and the implicit label, especially for the negative set. Distributions in Figure 2 show that the implicit labels are identical to the consensus human transcription for 66% of the positive and and 54% of the negative set.[9] We organized an annotation task among the three authors to categorize the cases that the implicit transcription and consensus human transcription disagree, and group disagreements into four categories:

- **minor lexical**: The query pair is semantically identical. Plurals, punctuation such as apostrophes and some word breaking that does not change the meaning fall into this category.

- **homophone**: Both queries make sense and whichever is the right one is not distinguishable from the audio. However, it is important to note that in homophone cases the user interaction logs are consistent with the implicit transcript interpretation of the intent.

- **implicit correct**: The implicit transcription is correct and the human transcription is not.

- **human correct**: The human transcription is correct and the implicit transcription is not.

The distribution of annotations for disagreement categories is presented in Figure 2 along with the portion where human and implicit transcriptions were identical. The distributions reveal that the implicit transcriptions are not only abundant and free, but also of pretty good quality. For nearly 88% of the positive set and the 90% of the negative set (everything except "human better" label), the implicit transcripts are as good, or better than the consensus of 5 independent human annotators. Table 2 shows a breakdown of WER and WebScore on these categories.

We see that minor lexical differences between implicit and human transcripts are responsible for a significant part of the overall recognition errors. Nearly 30% of all non-identical cases for the positive set and 17% for the negative set can be attributed to errors that do not prevent the user from reaching a satisfactory search result. WebScore figures for this category show that there is reasonable overlap of results between the implicit and human transcripts. Also note that the main reason that these queries made it into our label sets is that the user indeed was not negatively affected by the minor mismatch in the recognition result and had a SAT click. Examples of minor lexical differences in the annotated set are plurality (e.g. `cups song` vs. `cup song`), stopwords (e.g. `a city in southern oregon` vs. `city in southern oregon`), and split-join variations that do not change the meaning of the query (e.g. `dailymotion` vs. `daily motion`).

Homophones are inherently ambiguous, and very hard to transcribe correctly without context. In our annotation we identified the transcription pairs that are indistinguishable via audio as a separate category, but we suspect in many such cases the implicit transcription is correct. The implicit transcript has the evidence of a SAT click or a typed transcript in its favor, which came from the user who issued the voice query, while the manual transcript has no such

---

[9]Interestingly, if we also consider disagreements in the human transcribers into account and compare how often the implicit transcript matches any of the five human transcripts, the agreement goes up to 80% and 71% for the positive and negative set, respectively.

evidence. Homophones are indeed another factor that affected the overall WER and WebScore numbers as a significant source of disagreement. We observe that human transcribers often pick the more popular transcript; they are less likely to know `the weeknd` (the singer) or `houzz` (the home improvement website), they tend to transcribe these as `the weekend` or `house`. Since this is a systematic problem with manual transcripts, collecting the consensus of multiple transcriptions may only reinforce it.

Quite often the human transcriptions are wrong altogether, more likely for queries that contain named entities. They either do not spell out a named entity correctly such as `excellon patch` vs. `exelon patch` (the correct name of the product), or they do not even realize this is an named entity and transcribe the query to – a sequence of – similar sounding words in their vocabulary, such as `shaq in a pool` vs. `shaqtin a fool`, the TV show. These cases were annotated as 'implicit correct' and significantly affect the overall WER and WebScore values. If this proportion is excluded, overall WER would go down to 7.7% for positive set, and 10% for negative set. The negative set is affected more by this, because it contains more of these harder queries with named entities.

Queries in the last category are those that the human transcription is better than the implicit, which constitutes 12% of positive and 10% of negative labels. These mostly come from two sources. For the positive set, an incorrect speech recognition output can still return relevant results thanks to search engine's automatic spelling correction and other query rewriting models. For instance, if the recognition output is `cabellas` rather than `cabela's`, results about Cabela's are in the page, so it is not surprising to see a SAT click rather than a reformulation. For the negative set, the main reason for a wrong implicit transcription is users switching their queries slightly. For example, if user's voice query is `iphone 6`, and the speech recognizer has no errors, the user may still reformulate with no SAT click to `iphone 6s` followed by a SAT click. If `iphone 6s` was in the n-best candidates of the first query, we will incorrectly take the second query as an implicit transcript.

Overall, despite some cases where the human transcription is better, implicit transcripts are a plentiful source of free transcription data. With an optimistic interpretation, we believe that sometimes the implicit transcript offers a correct interpretation of the user's true intent in cases where manual transcripts can not. Even with a pessimistic interpretation, where we assume all disagreements are faults in the implicit label, they still may provide sufficient evidence to be of use as training data.

4. LEARNING TO RERANK N-BEST

The analyses in the previous section confirmed the quality of implicit transcripts and revealed their potential as training data for optimizing speech recognizers. In addition, we observed that implicit transcripts tend to match the user's search intent semantically in cases where there are minor lexical misrecognitions.

However, it is not trivial to train an end to end ASR system (as illustrated in Figure 1) with implicit transcripts. For instance, the acoustic model can be confused during training when it faces cases in which some of the audio frames do not correspond to anything in target word sequence. For example, consider a scenario where the user said `caltrain station in san carlos`, but the output of the recognizer (and hence the submitted query and implicit target) was `caltrain station san carlos`. The user is likely to be satisfied with the results returned for the recognized and submitted query. However, if we use the utterance audio and implicit target to train an acoustic model, there will be an interval in the utterance that corresponds to the word which is missing from the target. Thus, we set up our optimization framework with

implicit targets as a *reranking* problem. In the first stage, the default speech recognizer generates its candidate hypotheses (n-best). Our reranking models trained by implicit transcripts then receive this original list and rerank them based on available features. Therefore, the default ASR model acts as a candidate generation step, and the top candidate of the reranker will be the submitted query. We refer to our n-best reranking optimization framework as $\mathcal{N}ero$. An overview of $\mathcal{N}ero$'s architecture is presented in Figure 3.

Formal definition. Given an utterance o and the n-best results η returned by the default speech recognizer π, the goal is to learn an optimal reranking model π^* that reranks η – if necessary – in order to rank the *best* recognition candidate on top. That is,

$$\pi^*(\eta) = F(\pi(\eta), \Theta) \quad and, \quad E(\pi^*(\eta)) \geq E(\pi(\eta)) \quad (1)$$

where, F and Θ respectively represent the reranking function and its feature set, and E is the optimization function that uses implicit transcripts as its target. Our training procedure is inspired by previous work on learning reranking models for personalization [3, 8, 30, 31]. Since Fox et al. [8] demonstrated that SAT-clicks can be regarded as a proxy for relevance, many approaches (e.g. [3, 29]) have adopted implicit labels inferred from such clicks for training personalized rankers. The training process is similar across all these techniques; a set of search impressions are sampled where each impression consists of a user query along with a unique identifier and its click statistics. For training and evaluation, the user sessions and contextual features and interactions are simulated and *replayed*. Documents with SAT-clicks are regarded as pseudo-relevant and others are treated as non-relevant. The reranking models are trained to rerank the search results based on available features, so that the SAT click documents appear on top positions. Similar techniques have been proposed for personalizing auto-completion [30] and proactive zero-query recommendations [31]. In the former, the submitted query is considered as ground-truth for sampled prefixes and in the latter SAT clicks and SAT *views*[10] are used to generate implicit labels. Our work is inspired by all these previous methods and uses a similar technique for reranking n-best lists.

User logs & transcripts. We conducted our experiments on the query logs of Bing search engine. We sampled voice queries that were submitted between December 1st, 2014 and February 14th, 2015. We use $\mathcal{Q}$ to refer to this sample and subsequently subsample it to generate different experimental sets with implicit and explicit transcripts. We use the procedure described earlier in Section 3 to generate implicit labels based on SAT clicks and voice-to-text reformulations. We discarded all queries in $\mathcal{Q}$ that could not be labeled due to lack of SAT clicks or reformulations[11] and ended up with $1,204,057$ labeled utterances (queries) in total. We used the utterances sampled between December 1st, 2014 and January 31st, 2015 ($858,008$ in total) for training and validation, and the remaining utterances sampled during the first half of February 2015 ($346,049$ in total) for testing. We use $\mathcal{R}$ to represent implicit transcripts that are used to label n-best candidates in these datasets.

Our experiments in Section 3 verified the high quality of implicit transcripts and confirmed that they can be deployed as a reasonable proxy for manual transcripts. However, we observed that for a small percentage of queries, manual and implicit transcripts noticeably differ. Besides, the great majority of previous work in the literature

---

[10]Cards views with viewport duration of ≥ 30 seconds on mobile screen are classified as SAT views.
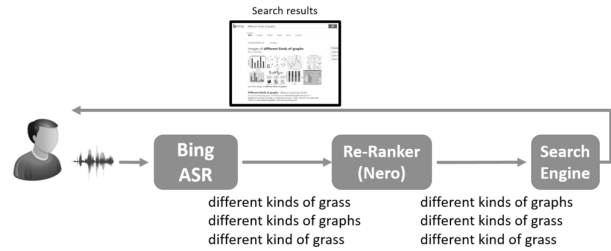[11]The discarded queries accounted for 48% of impressions.

Figure 3: The overall architecture of our n-best list reranking system (Nero). The original n-best list is generated by the default speech recognizer and is reranked according to available features. The top-ranked candidate in the reranked n-best is submitted as the final search query.

have been evaluated against manually transcribed data. Hence, we also generate an evaluation dataset based on manual transcripts. For this purpose, we randomly selected $1,327$ impressions (query utterances) from the testing subset of $\mathcal{Q}$, and crowdsourced their transcripts as explained in Section 3. We denote our manually transcribed test data by $\mathcal{R}$.

Training. As in many previous approaches on training personalized rankers [3, 29–31], we adopted LambdaMART [35] as our preference learning model. LambdaMART is a learning to rank approach based on gradient boosted decision trees, and has been reported to be one of the most effective optimization models for training supervised rankers [5]. An n-best suggestion is considered as *relevant* (label = 1) if it is identical to the implicit transcript and *nonrelevant* (label = 0) otherwise. We performed parameter sweep on our validation dataset over the number of trees {100, 300, 500}, learning rate {0.5, 0.75, 1.0, 1.5}, minimum number of instances per leaf {10, 50, 100}, and number of leaves per tree {10, 20, 50}. We applied the best combination of parameters on our test dataset to generate all experimental results reported in the following sections.

In the following section we evaluate $\mathcal{N}ero$ by both implicit and explicit transcripts, and compare its effectiveness against a state-of-the-art industry ASR production system as baseline.

5. RESULTS

In the first set of experiments we leverage a set of standard speech recognition features to optimize $\mathcal{N}ero$. These are specified as *speech recognition raw features* (♠) under Table 3. For now please ignore the remaining feature sets as we will return to them later. The raw features include the output of acoustic model and a number of language modeling scores computed over various corpora.

In all experiments, we use the speech recognizer of Bing search engine as our baseline. The baseline system consists of a decoder that produces a set of recognition hypotheses (n-best) that we use for reranking. The baseline language model is trained on diverse sources of data such as mobile and desktop query logs. Its acoustic model is developed based on a sequence-trained context-dependent deep neural net model with a front-end of 29 log filter bank features and their first and second derivatives. Our baseline is trained on a large collection of manual transcripts and benefits from a rich set of features and a multi-tier production ranking architecture.

Table 4 summarizes the evaluation results for $\mathcal{N}ero$ measured by three different metrics on two sets of transcripts. The gains and losses are computed against the production speech recognizer of

Table 3: The list of features used in our experiments for training the speech recognition models. The features are grouped into three categories depending on how they have generated. Here, o represents the speech utterance received by the recognizer, c denotes a suggested candidate in the n-best list, and c_0 specifies the top suggestion in n-best. We use $\Omega = \{q_1, q_2, \cdots q_n\}$ to represent all (if any) previous search queries that were submitted by the user in the same session. Ω_v and Ω_t respectively contain subsets of queries in Ω that were submitted by voice and text. We use $\hat{q}$ and $\underline{q}$ to represent clicked and abandoned queries.

Name	Description
Speech recognition raw features ($\spadesuit$)	
$\Phi(o, c)$	The acoustic model score of c.
$\Theta(o, c)$	The language model score of c.
$\Theta_q(o, c)$	The language model score of c computed over a collection of query logs.
$\Theta_t(o, c)$	The language model score of c computed over a collection of web document titles.
$\Theta_a(o, c)$	The language model score of c computed over a collection of web anchor text.
$\Theta_b(o, c)$	The language model score of c computed over a collection of web document body texts.
Query-log reformulation features ($\blacklozenge$)	
$\Sigma_{V2T}(c_0, c)$	The total number of times c_0 was reformulated to c in the logs in a voice to text transition.
$\Sigma_{All}(c_0, c)$	The total number of times c_0 was reformulated to c in the logs.
$\Sigma_{All}(\underline{c_0}, \hat{c})$	The number of times c_0 was reformulated to c in the logs in which query c led to a click and c_0 had no clicks.
$\Sigma_{All}(c_0, \hat{c})$	The number of times c_0 was reformulated to c in the logs in which query c led to a click.
$\Sigma_{Users}(c_0, c)$	The number of users who have reformulated their query from c_0 to c in the logs.
Contextual features ($\clubsuit$)	
$\mathcal{N}*(c)$	#times that query c has been issued previously in the session at any form (text or voice).
$\mathcal{N}_v(c)$	#times that query c has been issued previously in the session by voice.
$\mathcal{N}$	

Table 4: The relative performance of $\mathcal{N}ero$ compared to Bing speech recognizer. The implicit and manual transcript sets for evaluation are respectively represented by $\mathcal{R}$ and $\mathcal{R}$ (the training is always done on implicit transcripts). The relative gains and losses are respectively represented by ▲ and ▼ (lower is better for WER and worse for MRR and WebScore). Statistical significant differences detected by the paired t-test ($p < 0.001$) are denoted by †.

Transcripts	ΔWER	ΔMRR	ΔWebScore
Manual ($\mathcal{R}$)	▼ $+7.22\%$	▼ $-4.33\%^\dagger$	▼ -3.26%
Implicit ($\mathcal{R}$)	▲ $-40.28\%^\dagger$	▲ $+6.75\%^\dagger$	▲ $+3.77\%^\dagger$

Table 5: The impact of the number of implicit labels used for training on effectiveness of $\mathcal{N}ero$. The numbers are compared against a model trained with all available implicit labels. In the top section, the manual transcript set ($\mathcal{R}$) is used for mining the ground-truth transcripts, while in the bottom section implicit transcripts ($\mathcal{R}$) are used for evaluation. The losses are represented by ▼ and those that are statistically significant (paired t-test, $p < 0.001$) are distinguished by †. Note that the increase in WER is considered loss.

Sample Size	ΔWER	ΔMRR	ΔWebScore
Manual transcripts ($\mathcal{R}$)			
10%	▼ $+25.23\%$	▼ $-2.32\%^\dagger$	▼ $-2.71\%^\dagger$
1%	▼ $+62.64\%^\dagger$	▼ $-5.61\%^\dagger$	▼ $-6.31\%^\dagger$
0.1%	▼ $+107.00\%^\dagger$	▼ $-13.91\%^\dagger$	▼ $-13.47\%^\dagger$
Implicit transcripts ($\mathcal{R}$)			
10%	▼ $+14.64\%^\dagger$	▼ $-1.75\%^\dagger$	▼ $-3.24\%^\dagger$
1%	▼ $+55.49\%^\dagger$	▼ $-7.18\%^\dagger$	▼ $-8.28\%^\dagger$
0.1%	▼ $+107.04\%^\dagger$	▼ $-15.21\%^\dagger$	▼ $-12.43\%^\dagger$

Contextual speech recognition. We drew connection between $\mathcal{N}ero$ and previous work on personalized rerankers earlier in Section 4. We described that $\mathcal{N}ero$'s implicit transcripts are analogous to SAT clicks, submitted queries, and SAT views which are respectively used as training labels for optimizing personalized document search [3, 29], personalized auto-completion [30], and contextual proactive card rerankers [31]. Inspired by these previous approaches, we enhance $\mathcal{N}ero$ by adding features that can capture the user context for more effective reranking. Going back to the example at the beginning of this paper (Table 1), $\mathcal{N}ero$ should learn that if the user has already implicitly *rejected* a recognition output in the session (e.g. different kinds of grass), it may be unlikely that he or she is satisfied with it in the following queries. The complete list of contextual features used in our experiments can be found in Table 3, distinguished by ♣. The feature list includes signals that are designed to capture the user's (dis)satisfaction with the results (e.g. time since last query, or number of clicked documents for previous query). There are also several features that try to capture the topic of the session and compare each n-best suggestion with recently submitted queries by the user (e.g. character trie-gram similarity with the last query).

Collecting personalized judgments – and in our case transcripts – has been acknowledged to be costly, subject to noise and nontrivial [3, 30, 31]. Therefore, we follow the commonly used strategy in previous work and only evaluate our personalized rerankers by implicit transcripts ($\mathcal{R}$). Our reranking framework is generic and capable of integrating different types of features. Therefore, we also experiment with a set of query histogram features aggregated over historical logs. These features are denoted by ♦ in Table 3 and can

Table 6: The impact of additional feature sets on the performance of $\mathcal{N}ero$ measured by three different metrics on implicit transcripts ($\mathcal{R}$). A $\mathcal{N}ero$ model trained only with speech recognition raw features (♠) is used as the baseline for comparison. The query-log reformulation and contextual feature sets are respectively represented by ♦ and ♣ symbols. All numbers represent gains and are statistically significant according to the paired t-test ($p < 0.001$).

Feature sets	ΔWER	ΔMRR	ΔWebScore
$\mathcal{N}ero$(♠♦)	▲ -11.51%	▲ $+0.76\%$	▲ $+1.58\%$
$\mathcal{N}ero$(♠♣)	▲ -12.77%	▲ $+2.37\%$	▲ $+0.74\%$
$\mathcal{N}ero$(♠♦♣)	▲ -23.09%	▲ $+3.03\%$	▲ $+2.23\%$

Table 7: The list of $\mathcal{N}ero$'s top 5 most important features when trained over the complete feature set by implicit labels. The feature weights are all normalized with respect to the feature with the highest weight. The full description of features can be found in Table 3.

Feature	Weight	Feature group
$\Theta_q(o, c)$	1.00	speech recognition raw features (♠)
$\Sigma_{V2T}(c_0, c)$	0.98	query-log reformulation features (♦)
$\Phi(o, c)$	0.84	speech recognition raw features (♠)
$\Sigma_{All}(c_0, c)$	0.40	query-log reformulation features (♦)
$\mathcal{N}_v(c)$	0.38	contextual features (♣)

also be used as a reference point for quantifying the gains achieved by contextual ranking.

The evaluation results in Table 6 show that $\mathcal{N}ero$ can benefit from integrating both sets of contextual (♣) and query-log (♦) features. The gains are more substantial for contextual features and all differences are statistically significant compared to a vanilla baseline model that is trained solely by speech features. The best performance is achieved when all features are used suggesting that each feature set models a unique part of the space that is not covered by others. The feature weights learned by $\mathcal{N}ero$ over the complete feature set confirm that all feature groups contribute to the gains. Inspecting the reranked n-best lists by the complete model individually and comparing them against the vanilla ranker (♠) reveals several instances in which $\mathcal{N}ero$ leverages contextual and query-log features for better recognition. For example, instead of twice misrecognizing lady ella as lady la and definition of flour as definition of flower, $\mathcal{N}ero$ recognizes the correct query on the second attempts. Likewise $\mathcal{N}ero$ corrects elegant show to ellen shown after observing another earlier *rejection* on the first voice query of the session (ariens show).

Table 7 includes the list of the top five most influential features. The feature weights is computed by summing the contribution of the feature (in terms of decrease in mean-squared error) across all splits and all trees in the model. It can be seen that all feature groups have at least one representative in this set. The top feature and the third feature are respectively the language modeling (based on a query language model) and the acoustic modeling scores assigned to the suggestion, and they both belong to the speech feature set (♠). The second and fourth features in the list are computed based on historical statistics aggregated over query logs (♠). The last feature in the list is contextual (♣) and measures the number of times the current suggestion is recognized (repeated) as a voice query in the previous impressions of the session.

To summarize, the experiments in this section confirm that $\mathcal{N}ero$ is capable of integrating various contextual and generic features, which leads to further significant improvements in recognition accuracy.

6. RELATED WORK

We leveraged user interaction logs such as clicks and reformulations to infer implicit transcripts that we later use to train speech recognition systems for voice queries. Hence, our work is related to several previous studies on spoken document retrieval, speech recognition, search reformulations and implicit labels.

Spoken document retrieval. Early research on effective document ranking for spoken queries dates back to two decades ago. From 1997 through 2000, the US national institute of standards and technology (NIST) organized a spoken document retrieval track (SDR) as a part of annual TREC[13] conference. The goal of the track was to measure the impact of speech recognition errors on retrieval effectiveness. Contrary to the experiments reported in this paper, the SDR track focused on retrieving spoken documents for typed queries, but not vice versa. Garofolo et al. [10] summarized the first three years of TREC SDR track and concluded that SDR is a *solved problem*. The authors confirmed that ASR errors and retrieval effectiveness are highly correlated. However, they pointed out that the ASR errors – even as high as 40% – account for only 10% drop in search effectiveness. Singhal and Pereira [33] suggested that even when the speech recognition WER is as high as 65%, the retrieval quality can be maintained at a reasonable rate using document expansion techniques. It is important to note that the majority of test queries in SDR tracks were long (e.g. averaging 14.7 words for 1998 track). Later on, experiments by Crestani [6] revealed that the impact of ASR errors is indeed much higher for shorter voice queries. Similarly, Barnett et al. [2] confirmed that for 30% recognition error, the average loss for "very short" queries (2–4 words) is about 31%.[14] This is notable as the average length of voice queries has been reported to be between 2.5–3.8 words [28, 36] which falls in the same range.

Reformulation & correction. Search reformulations have been the subject of several studies. Search users often reformulate their queries when their information needs are not completely satisfied with the current results [12, 17], and when their voice queries are misrecognized [15, 32]. In this paper, we are mostly interested in the latter category in which the users try to *correct* the speech recognition errors. Automatic detection of speech recognizer mistakes is essential for improving the performance of ASR systems and has been well-studied in the speech recognition literature.

Levow [22] trained decision-tree classifiers based on acoustic and prosodic features (e.g. pause and stress) to identify *corrections*. They found an overall increase in both utterance and pause duration when users repeat to correct a misrecognized query. Orlandi et al. [24] relied on prosodic features to detect correction utterances and consequently adapted the language models for better recognition.

In the context of web search, recognizing correction reformulations is relatively under-studied. The majority of *voice-to-voice* reformulations – where the original voice query is immediately followed by another voice query – are not corrections [21]. *Voice-to-text* reformulations are more likely to capture corrections [15, 32] but that is not always the case [19]. Levitan and Elson [21] showed that correction retries of voice queries can be predicted with about 81% accuracy. To train their classifier, they used features that measured similarity, correctness confidence, and the estimated recognizability of voice queries. Sarma and Palmer [27] deployed co-occurrence word statistics to identify misrecognized words in ASR outputs.

Kou et al. [19] reported that between 30%–40% of voice to text reformulations in search logs that happen within a time-span of 30 seconds or less are correction queries. That is, the typed reformulated query is the exact transcription of the original spoken utterance. They trained a logistic regression classifier based on word-based, character-based, phoneme and acoustic features to mine correction pairs and achieved 90% precision at 75% recall.

Implicit relevance labels. Optimizing retrieval models by implicit labels dates back at least to introduction of pseudo-relevance (PRF) in the 70s [1]. In PRF, the top-ranked documents returned for a query are used to refine and resubmit the query on the fly before presenting the results to the user. The top-ranked results used by PRF can be regarded as a *system* feedback which is later used to improve search results. Joachims [16] demonstrated that *user* feedback in the form of clickthrough can be also used directly to enhance retrieval models.

Implicit labels have been frequently used to train personalized rankers in web search [3, 29, 30, 34]. Collecting editorial labels manually for training personalized rankers is challenging; different users may interpret the same query differently or might be interested in separate aspects and facets of it. Fox et al. [8] demonstrated that SAT clicks can be considered as a reasonable proxy for relevance. Since then training personalized reranking models based on SAT-click labels has become a common practice [3, 29, 34] and has been extended to other areas such as auto-completion [30], and zero-query recommendations [31]. The implicit transcripts inferred by our model to rate each n-best candidate, are analogues to implicit labels in these methods.

7. CONCLUSIONS

In this paper we proposed a novel technique for mining implicit transcripts from the user interactions stored in voice search logs. We verified the quality of our implicit transcripts and confirmed that it is at least as good as manually transcribed data in almost 90% of cases. We used the implicit transcripts mined from search logs to train a supervised n-best reranker ($\mathcal{N}ero$). Our experimental results suggest that $\mathcal{N}ero$ significantly improves the quality of speech recognition as measured by three different metrics. We also demonstrated that $\mathcal{N}ero$ can benefit from integrating other feature sets and can produce contextual reranking in the presence of contextual features.

There are several directions for future work. Our experiments were conducted over English speaking users in the US and the effectiveness of $\mathcal{N}ero$ in other languages is yet to be explored. It would be also interesting to enhance $\mathcal{N}ero$ with additional features based on user demographics and long-term history. Last but not least, joint optimization over both implicit and manual transcripts may lead to further improvements in recognition accuracy and consequently better search results.

References

[1] R. Attar and A. S. Fraenkel. Local feedback in full-text retrieval systems. *Journal of the ACM*, 24(3):397–417, 1977.

[2] J. Barnett, S. Anderson, J. Broglio, M. Singh, R. Hudson, and S. Kuo. Experiments in spoken queries for document retrieval. In *Eurospeech*. Citeseer, 1997.

[3] P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, and X. Cui. Modeling the impact of short-and

---

[13]
[14] The retrieval effectiveness was measured by precision, which is defined as the proportion of returned documents that were relevant.

long-term behavior on search personalization. In *Proc. SIGIR*, pages 185–194. ACM, 2012.

[4] L. D. Catledge and J. E. Pitkow. Characterizing browsing strategies in the world-wide web. *Computer Networks and ISDN systems*, 27(6):1065–1073, 1995.

[5] O. Chapelle and Y. Chang. Yahoo! learning to rank challenge overview. In *Yahoo! Learning to Rank Challenge*, pages 1–24, 2011.

[6] F. Crestani. Word recognition errors and relevance feedback in spoken query processing. In *Flexible Query Answering Systems*, pages 267–281. Springer, 2001.

[7] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366, 1980.

[8] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems*, 23(2):147–168, 2005.

[9] M. Gales and S. Young. The application of hidden markov models in speech recognition. *Foundations and trends in signal processing*, 1(3):195–304, 2008.

[10] J. Garofolo, C. Auzanne, and E. Voorhees. The TREC spoken document retrieval track: A success story. *NIST Special Publication SP*, 500(246):107–130, 2000.

[11] A. Halfaker, O. Keyes, D. Kluver, J. Thebault-Spieker, T. Nguyen, K. Shores, A. Uduwage, and M. Warncke-Wang. User session identification based on strong regularities in inter-activity time. In *Proc. WWW*, pages 410–418, 2015.

[12] A. Hassan, X. Shi, N. Craswell, and B. Ramsey. Beyond clicks: Query reformulation as a predictor of search satisfaction. In *Proc. CIKM*, pages 2019–2028. ACM, 2013.

[13] A. Hassan Awadallah, R. Gurunath Kulkarni, U. Ozertem, and R. Jones. Characterizing and predicting voice query reformulation. In *Proc. CIKM*, pages 543–552. ACM, 2015.

[14] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[15] J. Jiang, W. Jeng, and D. He. How do users respond to voice input errors? lexical and phonetic query reformulation in voice search. In *Proc. SIGIR*, pages 143–152. ACM, 2013.

[16] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. SIGKDD*, pages 133–142. ACM, 2002.

[17] R. Jones and K. L. Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proc. CIKM*, pages 699–708. ACM, 2008.

[18] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Proc. ICASSP*, volume 1, pages 181–184. IEEE, 1995.

[19] Z. Kou, D. Stanton, F. Peng, F. Beaufays, and T. Strohman. Fix it where it fails: Pronunciation learning by mining error corrections from speech logs. In *Proc. ICASSP*, pages 4619–4623. IEEE, 2015. ISBN 978-1-4673-6997-8.

[20] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[21] R. Levitan and D. Elson. Detecting retries of voice search queries. In *Proc. ACL*, pages 230–235, 2014.

[22] G.-A. Levow. Characterizing and recognizing spoken corrections in human-computer dialogue. In *Proc. COLING*, pages 736–742. ACL, 1998.

[23] J. H. Martin and D. Jurafsky. Speech and language processing. *International Edition*, 2000.

[24] M. Orlandi, C. Culy, and H. Franco. Using dialog corrections to improve speech recognition. In *ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*, 2003.

[25] L. Rabiner and B.-H. Juang. Fundamentals of speech recognition. 1993.

[26] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[27] A. Sarma and D. D. Palmer. Context-based speech recognition error detection and correction. In *Proc. HLT-NAACL*, pages 85–88. ACL, 2004.

[28] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope. "your word is my command": Google search by voice: A case study. In *Advances in Speech Recognition*, pages 61–90. Springer, 2010.

[29] P. Serdyukov, G. Dupret, and N. Craswell. Log-based personalization: The 4th web search click data (WSCD) workshop. In *Proc. WSDM*, pages 685–686. ACM, 2014.

[30] M. Shokouhi. Learning to personalize query auto-completion. In *Proc. SIGIR*, pages 103–112. ACM, 2013.

[31] M. Shokouhi and Q. Guo. From queries to cards: Re-ranking proactive card recommendations based on reactive search history. In *Proc. SIGIR*, pages 695–704. ACM, 2015.

[32] M. Shokouhi, R. Jones, U. Ozertem, K. Raghunathan, and F. Diaz. Mobile query reformulations. In *Proc. SIGIR*, pages 1011–1014. ACM, 2014.

[33] A. Singhal and F. Pereira. Document expansion for speech retrieval. In *Proc. SIGIR*, pages 34–41. ACM, 1999.

[34] R. W. White, P. N. Bennett, and S. T. Dumais. Predicting short-term interests using activity-based search context. In *Proc. CIKM*, pages 1009–1018. ACM, 2010.

[35] Q. Wu, C. J. Burges, K. M. Svore, and J. Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, 2010.

[36] J. Yi and F. Maghoul. Mobile search pattern evolution: the trend and the impact of voice queries. In *Proc. WWW*, pages 165–166. ACM, 2011.