Auto-annotation for Voice-enabled Entertainment Systems

Wenyan Li
Wenyan_Li@comcast.com
Comcast Applied AI Research Lab

Ferhan Ture
Ferhan_Ture@comcast.com
Comcast Applied AI Research Lab

ABSTRACT

Voice-activated intelligent entertainment systems are prevalent in modern TVs. These systems require accurate automatic speech recognition (ASR) models to transcribe voice queries for further downstream language understanding tasks. Currently, labeling audio data for training is the main bottleneck in deploying accurate machine learning ASR models, especially when these models require up-to-date training data to adapt to the shifting customer needs. We present an auto-annotation system, which provides high quality training data without any hand-labeled audios by detecting speech recognition errors and providing possible fixes. Through our algorithm, the auto-annotated training data reaches an overall word error rate (WER) of 0.002; furthermore, we obtained a reduction of 0.907 in WER after applying the auto-suggested fixes.

ACM Reference Format:
Wenyan Li and Ferhan Ture. 2020. Auto-annotation for Voice-enabled Entertainment Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401241

1 INTRODUCTION

Millions of customers use voice-activated intelligent entertainment systems like Comcast X1 and Amazon Alexa every day. Through automatic speech recognition (ASR), voice utterances are transcribed into text queries and fed to downstream natural language understanding modules. Despite many advancements in speech-to-text transcription in the past decade with Deep Neural Networks (DNNs), ASR is still far from perfect in practice. Particularly, audio data labeling has become the primary bottleneck in deploying high-accuracy ASR systems. Labeling data is even more time-consuming and labor-intensive in specialized domains where training data needs to adapt to domain specifics and shifting customer trends. This makes manual annotation and correction for every transcription almost impossible. To alleviate this key problem and to improve the ASR systems, it is crucial to automatically identify and correct transcription errors in an **unsupervised** manner.

Recent research has characterized speech recognition errors and examined both lexical and acoustic features of query reformulation [7], focusing on automatic detection and correction of substitution, deletion and insertions errors in ASR [3]. They are all based on **fully supervised** methods, where features and analyses are extracted from a manually labeled dataset.

This paper presents an auto-annotation system, which detects transcription errors from user voice queries for major entertainment platforms. This system is based on human behavior after users send voice queries, relying only on transcribed text of time-ordered voice queries sessions (similar to [7]) from the ASR output. We are, to the best of our knowledge, the first to tackle this problem without any hand-labeled data.

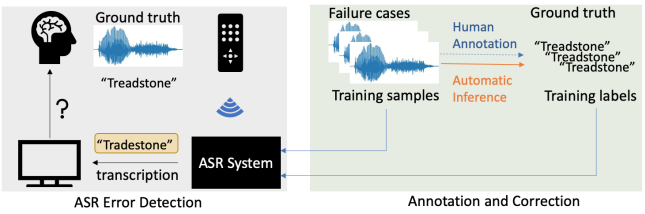

Figure 1: General process of identifying recognition errors in transcription and enhance ASR system with annotated samples.

2 BACKGROUND AND RELATED WORK

The rise of supervised learning has been encouraged by the abundance of data sources, increased investment by companies, and access to computing power at scale. This, among other challenges, has led to relatively little work in unsupervised methods in the field. In this paper, we propose a method to obtain training data for production-level ASR systems with little to no supervision, based on simple rules and clever information processing techniques.

Previous research on automatic detection and classification of ASR errors mainly uses annotated data and focuses on lexical and acoustic features of query reformulation [2, 3, 7]. Hong et al. [4] identified speech recognition errors based on audio-only outputs.

While weakly supervised learning has advanced in multi-label text and image classification tasks [1, 8], existing efforts on using weakly-labeled audio are limited. Weak labels on acoustic audio event detection were first used in [5] through multi-instance learning. On a production-level, Amazon Sagemaker Ground Truth [6] provides automated data labeling in a human-in-the-loop manner, where annotators label ambiguous samples in the active learning process. Unsupervised annotation on ASR transcriptions remains a difficult and under-explored problem.

3 PROBLEM

We want to automatically identify errors and provide reasonable corrections in ASR systems to guide or avoid annotations. For example, in Figure 1, ASR wrongly outputs 'Tradestone' when some users

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401241

ask for the popular TV series 'Treadstone'.[1] To avoid such recognition errors and improve overall user experience, the ASR system needs to be updated frequently with high quality annotated data. This involves classification and annotation on the transcriptions from ASR outputs: finding the true positives, as well as detecting the erroneous ones and providing corresponding corrections. Manually going through such cases is tedious and infeasible at large scale, thus automatic annotation is highly desired.

We follow Tang et al. [7], who analyzed ASR errors and subsequent user behaviors on an entertainment system. While Tang et al. [7] analyzed the errors with human annotated data, we extend on the analysis and use an unsupervised approach to identify such errors and provide reasonable corrections automatically.

4 METHODS

We use two methods to evaluate ASR outputs and provide annotations on the transcriptions. First, we investigate sessions of user voice queries, extract both session-level and query-level features, and identify patterns for query reformulation. This method uses only time-ordered ASR transcriptions of user utterances, so we name it **utterance-based annotation**. Second, we collect button clicks, media tune and app activities to study users' interactions with the platform after they issue a voice request. We call this **interaction-based annotation**.

4.1 Utterance-based Detection and Annotation

Our dataset is constructed from voice query sessions. Following [7], a *voice query session* is defined as a set of time-ordered queries where each query comes from the same device and there are at most 45 seconds between consecutive queries.

4.1.1 Identifying erroneous transcriptions. Tang et al. [7] have shown that WER is positively correlated to session length and users have a high probability of repetition when facing transcription errors. Thus, when classifying on the ASR transcriptions of user utterances, we consider factors that could introduce errors during transcribing on both session and query levels.

On the session level, a session is very likely to have wrong transcriptions if it contains multiple transcriptions with the same content, e.g., user is trying to correct the automatic transcription. *On the query level*, the transcription is suspected to be erroneous when the query (i) is often repeated by users in a session, and (ii) has a short time interval from the previous one before it is repeated.

Furthermore, a query is considered important when it appears in many sessions. As such, for each query, we construct three features: $s(q)$, $L_{rep}(q)$, and $t_m(q)$. $s(q)$ denotes the number of sessions that contains query q, $L_{rep}(q)$ represents the likelihood of query q to be repeated, $t_m(q)$ denotes the median time interval between repeated query q in sessions, and S_{rep} denotes the set of sessions that contain repeated queries. An erroneous transcription, based on these features, is defined as:

$$Err(q) = \begin{cases} 1 & \text{if } s(q) > T_s, L_{rep}(q) > T_{rep}, t_m(q) < T_t, \\ & \text{s.t. } q \in S_{rep}, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

[1] 'Treadstone' is an new popular drama television series, starting from September, 24th on USA network.

where T_s, T_{rep}, and T_t are hyperparameters that denote the threshold of the respective features. Empirically, we want to find erroneous transcriptions that are common among sessions, problematic to users so that they repeat hastily and frequently. We find that $T_s = 300$, $T_{rep} = 0.2$, and $T_t = 13$ are reasonable and robust assumptions.

Based on these parameters, a transcription is considered *erroneous* if it appears in over 300 user sessions, has more than 20% chance of being repeated by a user, with a median of 13 seconds between subsequent repetitions.

4.1.2 Automatic annotation with query reformulation patterns. To reduce human effort on labeling audio (i.e., transcribing from the source audio and correcting erroneous transcriptions), we use query reformulation patterns to infer corrections.

We investigate sessions that contain multiple queries which are *not* exact repeats in transcriptions. In these sessions, users try to reformulate the query with variations in speech rate, pronunciations, etc., that yield different transcriptions from ASR. As shown in [7], the last query in the session has a much lower word error rate on average, thus we consider the last query in a session to more likely be the corrected transcription of the previous query.

For each detected erroneous transcription q_{err} (as defined in Section 4.1.1), we infer its possible correction by selecting sessions $\{s_1, s_2, \ldots, s_n\}$, where each session $s_i = (q_1, q_2, \ldots, q_{t_i})$ contains t_i time ordered user voice queries and meets the following conditions:

(1) the session has more than one query and does not contain repeated transcriptions;
(2) the median time interval between queries is less than the threshold T_t;
(3) $Err(q_{t_i-1}) = 1$.

Then, from each of the session s_i, we extract the last two transcriptions q_{t_i-1} and q_{t_i}, where q_{t_i} is a correction candidate for q_{t_i-1}. We group the extracted (q_{t_i-1}, q_{t_i}) pairs by the erroneous transcription q_{t_i-1}, collecting its possible corrections candidates among all sessions. For a specific erroneous transcription q_{t_i-1}, denoted as q_{err}, we calculate the confidence of each of its unique correction candidate $q_i^{'}$, as in Equation 2, and select the most confident candidate as correction.

$$P(q_i^{'} \mid q_{err}) = \frac{count(q_i^{'}, q_{err})}{count(q_{err})}. \quad (2)$$

While in a single session, it is possible for q_{t_i} to be a wrong correction for q_{t_i-1}. Across all sessions, we often find that the best correction candidate has much higher confidence than the rest of the correction candidates.

4.2 Interaction-based Detection and Annotation

To analyze users' subsequent interactions after a voice request, we obtain button clicks, media tune, and app launch information. We group and split these events into user sessions in a similar way to Section 4.1 with the additional constraints:

(1) The session begins with a voice query.
(2) Each non-voice event occurs within 30 seconds of the last event.

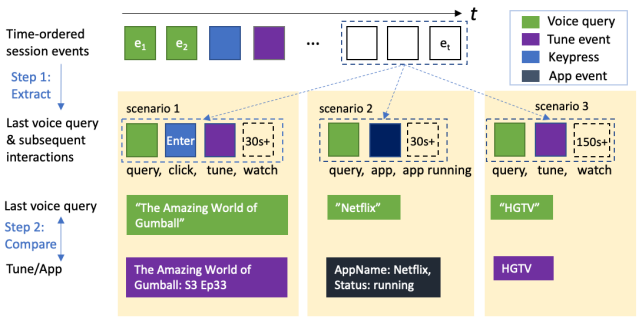

Figure 2: Three scenarios where subsequent user interactions lead to a *confirmed transcription*.

(3) A tuning event or app launch event occurs at the end, within 30 seconds of the previous event.

We hypothesize that users' consecutive behavior (after they utter a command) reflects their underlying intent. Based on this assumption, we look at patterns of behavior among a large user base, and use these patterns to *confirm* the correctness of the ASR transcriptions. We defined[2] three kinds of subsequent user interactions that follows the last voice query in a session (Figure 2):

Scenario 1: User clicks a button, tunes to a program, then keeps watching for at least 30 seconds.

Scenario 2: User launches an app and stays for at least 30 seconds.

Scenario 3: User tunes to a program and stays for at least 150 seconds.

From each session, we extract the last voice query and subsequent events to approximate user's final intent. For the scenarios described above, we say the last query has a *confirmed transcription* based on the confirming action, whether it is a button click, program tune, or app launch.

After we *confirm* the transcription of the last voice query, we also look at the previous query from the same user to see if it points to a potential system error. If this query shares lexical similarity, and there are no following *confirming actions*, we regard the transcription of the former voice query to have high probability of ASR or some downstream error (Figure 3), thereby calling it a *suspicious query*. As the final user interaction is assumed to represent the underlying user intent, we hypothesize that it is a correction of the *suspicious query*. We test this hypothesis in the next section.

5 EVALUATION

We evaluate both utterance-based and interaction-based annotation on thousands of voice queries from real users. Notice that we use all the data in accordance with our institution's privacy policy.

5.1 Evaluation on utterance-based detection and annotation

As described in Section 4.1.1, we predict a transcription is erroneous if it satisfies the conditions in Equation 1. For such queries, Equation 2 determines the most likely correction. To evaluate this

[2]These criteria will vary based on the business domain, user interface and other factors. We merely provide an example that works for the entertainment platform we experimented with.

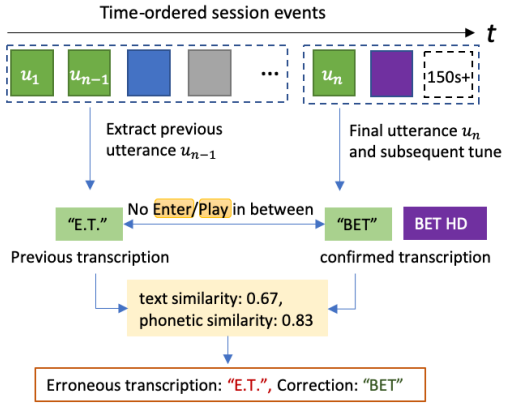

Figure 3: Identifying a correction of an erroneous transcription based on scenario 3.

approach, we extracted voice sessions that contain detected erroneous queries from 4 days of the production system, spread across a period of almost two months (August 14th, September 3rd and 15th, and October 8th, 2020). We ended up with a sample of 474 sessions (1499 utterances in total, among which 351 are unique), containing 925 utterances on 68 unique detected erroneous queries. Trained annotators listened to corresponding audios for each transcription and determined whether ASR was correct. If not, the annotator also determined the correct transcription. Note that it takes hours for human annotators to label hundreds of transcriptions.

We computed precision and recall for error detection: Among queries identified as erroneous, what percentage were actually wrong? Among queries that were determined by human annotators to be mis-transcribed, what percentage were actually recalled by our approach? For correction prediction, we look at the overall accuracy metric: Among queries for which our auto-annotation system made a prediction, what percentage were correct? We also computed these metrics separately for each unique query and report the distribution in Figure 4.

For error detection, we achieved an overall precision of 66.49% and a recall of 76.11%. The accuracy of corrections, on the other hand, was 69.76%. If these auto-suggested corrections were applied as is, the WER of the ASR system would have reduced by 0.907.

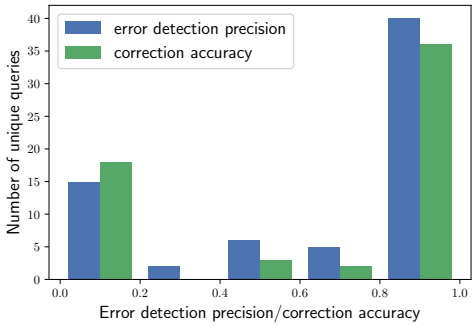

Figure 4: Distribution of error detection precision and correction accuracy

In the upper section of Table 1, we show examples of detected erroneous transcriptions and their corresponding corrections. Note

Table 1: Samples of erroneous transcription detection and automatically suggested annotation

Methods	Suspicious transcriptions	Count	Error detection precision	Suggested annotation	Correction accuracy
Utterance-based Annotation	Academy	15	0.60	Pup Academy	1.00
	CNET Latino	18	0.94	Cinelatino	1.00
	Entouch	17	0.94	In Touch	1.00
	Murder in the First	13	1.00	Murder in the Thrist	1.00
	Joe Josie Wah	17	1.00	JoJo Siwa	1.00
	Antena*	18	0.00	Antenna	0.00
Interaction-based Annotation	Play Select	15	0.53	Player Select	1.00
	Just Go With It	15	1.00	Just Roll With It	1.00
	Find	15	1.00	Friends/Blind Date	1.00
	Stephen	15	1.00	Steven Universe	0.73
	60 Days N	15	0.46	60 Days In	1.00
	The Singer	15	1.00	The Masked Singer/Masked Singer	0.53

Table 2: Samples of annotation failure cases

Detected queries	Human annotation	Automatic annotation
Wonderpark	correct	Wonder Park
Timer	correct	Sleep Timer
The Good Dr.	correct	The Good Doctor

that we are using human annotation as ground truth labels. However, there are some domain-specific errors detected by our algorithm that are difficult for human annotators to catch simply by listening to the audios in isolation. For example, the query 'Antena' is a wrong transcription for 'Antenna', however annotators wouldn't be able to tell the difference between its pronunciation with the correct spelling - 'Antenna'. Similar cases include 'AXS TV' vs. 'Access TV', 'Xenon' vs. 'Zenon'. This shows our method can identify transcriptions that are often missed by human annotators.

Examples shown in Table 2 are queries whose precision fall in range 0.0 to 0.2, where ASR has spelling errors or is actually accurate, however users keep repeating as a result of downstream modules fail to respond. This provides a false signal to our algorithm since the assumption is that these are due to transcription errors. Distinguishing between transcription errors and other downstream errors (e.g., natural language understanding) is outside of the scope of this work, yet an interesting future direction.

5.2 Evaluation on interaction-based detection and annotation

As described in 4.2, we extract *confirmed transcriptions* based on confirming actions (Figure 2). Any prior utterances are then extracted as *suspicious transcriptions* and the final transcription serves as the *suggested annotation* (Figure 3).

We first evaluated the accuracy of the *confirmed transcriptions*. We applied the proposed approach on all sessions in our production system on October 1st. From all the sessions in which we predicted a confirmed transcription, we randomly selected 225 queries in total (15 unique queries, with 15 utterance samples for each query) as an evaluation set. Based on human judgment, only one sample was actually transcribed incorrectly ('Pose' instead of 'Pause'); all other confirmed transcriptions are indeed correct. This translates into an accuracy of 99.6%, achieving an WER of 0.002, for our method of extracting confirmed transcriptions, thus making it a reliable source of high-quality training data for an ASR system.

Second, we evaluated our method for identifying *suspicious transcriptions*, by asking annotators whether they were indeed transcribed incorrectly. Our method achieves an accuracy of 87.1% on this task, thus making it a reliable source of ASR error examples.

Finally, we looked at whether our approach could automatically suggest an annotation for the *suspicious transcription*, thus generating free training data for examples the ASR system actively struggled with. The auto-annotation agreed with human annotations 80.0% of the time (see examples in Table 1, lower section).

6 CONCLUSION AND FUTURE WORK

We present an automated annotation system that provides an unsupervised approach to identify erroneous transcriptions and to suggest possible fixes for detected errors. As our approach only uses sequences of transcriptions and/or other events on user interactions, it can be directly applied to improve annotation efficiency and robustness of ASR systems. Potential extensions include building an end-to-end labeling system for ASR as well as applying these methods to downstream natual language understanding modules.

REFERENCES

[1] Alessandro Bergamo and Lorenzo Torresani. 2010. Exploiting weakly-labeled Web images to improve object classification: a domain adaptation approach. In *Advances in Neural Information Processing Systems 23*.

[2] Rahhal Errattahi, Asmaa El Hannani, and Hassan Ouahmane. 2018. Automatic Speech Recognition Errors Detection and Correction: A Review. *Procedia Computer Science* (2018).

[3] R. Errattahi, A. E. Hannani, H. Ouahmane, and T. Hain. 2016. Automatic speech recognition errors detection using supervised learning techniques. In *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications*.

[4] Jonggi Hong and Leah Findlater. 2018. Identifying Speech Input Errors Through Audio-Only Interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM.

[5] Anurag Kumar and Bhiksha Raj. 2016. Audio Event Detection Using Weakly Labeled Data. In *Proceedings of the 24th ACM International Conference on Multimedia*.

[6] Amazon Web Services. 2020. Amazon SageMaker Ground Truth. https://aws.amazon.com/sagemaker/groundtruth/

[7] Raphael Tang, Ferhan Ture, and Jimmy Lin. 2019. Yelling at Your TV: An Analysis of Speech Recognition Errors and Subsequent User Behavior on Entertainment Systems. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[8] Tong Wei, Lan-Zhe Guo, Yu-Feng Li, and Wei Gao. 2018. Learning safe multi-label prediction for weakly labeled data. *Machine Learning* (2018).