Improving Perceived Accuracy for In-Car Media Search

Yun-Cheng Ju , Michael Seltzer , Ivan Tashev

Microsoft Research, Redmond, Washington, USA
{yuncj|mseltzer|ivantash}@microsoft.com

Abstract

Speech recognition technology is prone to mistakes, but this is not the only source of errors that cause speech recognition systems to fail; sometimes the user simply does not utter the command correctly. Usually, user mistakes are not considered when a system is designed and evaluated. This creates a gap between the claimed accuracy of the system and the actual accuracy perceived by the users. We address this issue quantitatively in our in-car infotainment media search task and propose expanding the capability of voice command to accommodate user mistakes while retaining a high percentage of the performance for queries with correct syntax. As a result, failures caused by user mistakes were reduced by an absolute 70% at the cost of a drop in accuracy of only 0.28%.

Index Terms: speech recognition accuracy, music search, voice UI, voice command, CFG

1. Introduction

Users, especially new users, of a voice command dialog system often don't know or don't remember exactly what the system expects them to say. The reasons include: users do not have time or are reluctant to read the manuals; the syntax or keywords for some of the voice commands are confusing or difficult to remember. Additionally, users may be occupied or distracted and thus make mistakes unintentionally. Unfortunately, they all lead to a failed voice request and unhappy user experience.

As an example, there are four media keywords used in the media search task in our in-car infotainment system [1], namely the *track*, *album*, *artist*, and *genre*. As straightforward as it may seem, even the authors themselves make mistakes during demos. Sometimes we are focusing on reciting the exact title of a song and forget to include the keyword *track*. Other times we want to listen to an album by an artist but say "play *album* U2" instead of the correct command "play *artist* U2".

Command keywords are there to constrain the search space of speech recognition and to improve accuracy, but sometimes they get in the way and cause the system to fumble on user errors. How often do user mistakes occur in a deployed voice command system? In *Project54* [2], a Command and Control (C&C) application used in patrol cars for retrieving police information, the authors report roughly 63% of the failed voice commands were due to user error. Of that number, roughly 54% were from *ill-formed queries*, where users fail to use the correct keywords in their commands.

User mistakes are often not considered in system evaluation. This creates a gap between the *claimed accuracy* of the system and the actual accuracy perceived by the end users, or *perceived accuracy*. In this paper, we present quantitative studies and discuss design choices and tradeoffs related to recognizing ill-formed queries. Is it feasible to make our system more flexible and robust in handling ill-formed queries without sacrificing accuracy for *well-formed queries*, where users use the correct keywords in their commands? Is it necessary to modify the context free grammars (CFG) to achieve this goal? If so, what are the possible choices in the CFG topology and which one performs the best? And if possible, can we evaluate the tradeoffs without acquiring new test corpora? Besides ill-formed queries, users certainly make other types of mistakes such as reciting the title or the artist name wrong. We addressed how to alleviate such mistakes in a separate research [3].

The rest of this paper is organized as follows: In Section 2, we describe the current voice command grammar topology used in our system and the test set we use for evaluation. We confirm that current grammar topology, while very efficient for well-formed queries, cannot adequately accommodate ill-formed queries. We also share a novel workaround that allows us to evaluate the performance without acquiring new acoustic data, which is both expensive and time consuming to do so. In Section 3, we propose a few simple grammar topologies and compare their performance against both well-formed and ill-formed queries. In particular, we explain how developers can adjust the weights on our suggested topology to achieve satisfactory performance trade-offs. In Section 4, we compare our approach with a typical two-pass recognition strategy [4] and demonstrate the superiority of our proposed approach. Finally, we conclude this paper with a discussion.

2. Preliminary Study

Figure 1 illustrates both the command structure and the CFG topology (C) used for media search in our current infotainment system. Grammar developers usually faithfully craft the CFG topology to only accept the utterances specified by the command structure, expecting this practice to give their systems the highest performance.

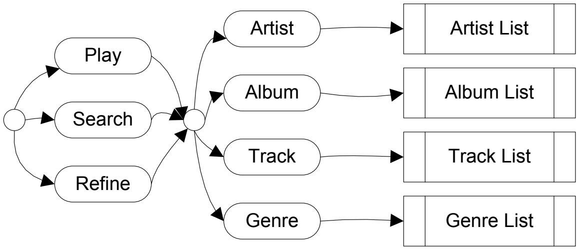

Figure 1: Current Media Search CFG Topology (C)

2.1. Evaluation corpus

We use a set of 2,131 waveforms provided by a car manufacture to evaluate our system performance. They are read utterances recorded in a quiet room and then mixed with different road noises to create a balance of the three driving conditions: "parked car", "local street", and "high way" (Table 1). The grammar has 4,950 entries in combination of entries from all four command keywords.

# of	Album	Artist	Track	Genre
Utterances	272	288	1256	315
Distinct Entries	247	245	649	79
Grammar Entries	1642	1284	1944	80

Table 1: Evaluation Corpus

In order to investigate whether the current grammar can accommodate queries with incorrect keywords, one might suggest we use the current grammar and re-record each utterance, replacing the keywords with incorrect ones (say from "track" to "album"). It is, however, quite a challenge in practice to create a completely matched corpus. Instead, we decided to change the metadata (or in practice the grammar) to simulate ill-formed queries by swapping the entries in the track list with the ones in the albums list. Table 2 shows the number of well-formed and ill-formed queries misrecognized. The Sentence Error Rate (SER) rises from 7.74% to 97.84%. In fact, if we remove the special cases where the same titles are both track names and album names, the system fails on every single query. The result strongly suggests the need to change the grammar topology in order to achieve flexibility toward ill-formed queries.

SER (%)	Overall	Album	Artist	Track	Genre
Well-formed	7.74	5.88	11.46	5.81	13.65
Incorrect Keyword	97.84	97.79	100.00	96.82	100.00

Table 2: SER for Well-formed and Ill-formed Queries

3. New Grammar Topologies

We propose a few simple topology changes to handle ill-formed queries. We further divide ill-formed queries into "*incorrect keyword"* queries and "*missing keyword"* queries and evaluate the performance of each proposed topology in handling these two types of ill-formed queries as well as well-formed queries. While we are interested in improving the accuracy for ill-formed queries, it is vital that we do not degrade performance on well-formed queries. As a result, we always evaluate the performance impact on well-formed queries first, and discard any approaches that cause significant performance degradation.

3.1. Possible grammar topologies

One obvious solution to address the keyword issues is to ignore them completely in the grammar topology (S0), as shown in Figure 2. Another approach is to simply merge the four lists into one big list (S1). Both are symmetric topologies and should accommodate both well-formed queries and "incorrect keyword" queries equally well. The only potential issue is whether they sacrifice the performance for well-formed queries too much. We also came up with a simple asymmetric topology (A0), as depicted in Figure 3 where we allow transitions to the incorrect keywords at 1/10 of the weights of the correct keywords.

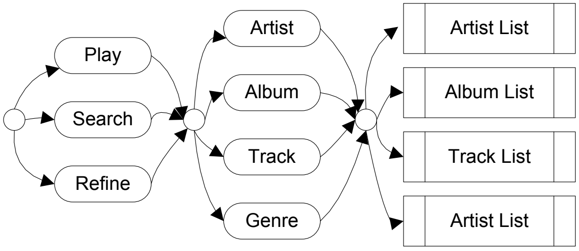

Figure 2: Symmetric Grammar Topology (S0)

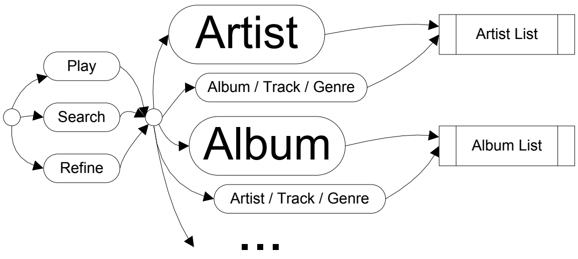

Figure 3: Proposed Asymmetric Topology (A0)

The performances of the three grammar topologies for both well-formed and ill-formed queries are summarized in Table 3. Clearly, both symmetric topologies (S0, S1) accommodate ill-formed queries very well, only 31% worse than well-formed queries. However, the same 31% relative error rate increase over well-formed queries is too costly for these two topologies to be adopted. One number worth noticing is the single list topology (S1) performs much worse on the Genre test cases because the weights of the genres were diluted due to the small size of the entries.

On the other hand, asymmetric A0 topology appears promising in providing good performance on both types of queries. Even though it doesn't accommodate the ill-formed queries (12.39%) as well as S0 topology does (10.14%), it only degrades the accuracy for well-formed queries for a relative 5.6%. We adopted this topology and made an intuitive modification to further accommodate the ill-formed queries with missing keywords.

SER (%)		Overall	Album	Artist	Track	Genre
Well-formed	C	7.74	5.88	11.46	5.81	13.65
	S0	10.14	7.72	13.89	8.44	15.56
	S1	10.70	7.72	12.85	7.17	25.40
	A0	8.17	6.25	12.15	6.05	14.60
Incorrect Keyword	C	97.84	97.79	100.00	96.82	100.00
	S0	10.14	7.72	13.89	8.44	15.56
	S1	10.70	7.72	12.85	7.17	25.40
	A0	12.39	10.66	15.63	10.99	16.51

Table 3: SER for Well-formed and "Incorrect keyword" Queries for All Topologies

3.2. Queries with missing keywords

In the previous section, we explained how we simulate a test corpus for the "*incorrect keyword*" queries in the previous section. However, we couldn't find any easier way to evaluate the "*missing keyword*" scenario without physically producing new waveforms with the keywords removed. The authors manually examined 40% of the original corpus waveforms and removed the portions associated with the keywords. In order not to affect the study by the artifact of hand editing, we processed the modified waveforms with a sanity speech recognition run. As an example, for waveforms in the track corpus, we use the grammar "Play Track <track name>" on the original waveforms and another grammar "Play <track name>" on the new edited waveforms. Any waveforms that receive different SR results were discarded. We selected a new test set of 791 utterances for our performance studies. Table 4 shows the number of utterances of the original corpus and newly generated corpus.

# of	Album	Artist	Track	Genre
Correct Utterances	272	288	1256	315
Missing Keyword Utterances	104	105	457	115

Table 4: Manually Generated "Missing Keyword" Corpus

We further augmented our asymmetric grammar topology (A0) to allow the skipping of command keywords. The weight for the skipped keywords was empirically set to five times the weights for incorrect keywords. Experimental results show this configuration to be very reasonable and can produce similar performance on both "incorrect keyword" queries and "missing keyword" queries. This final proposed grammar topology (A1) is depicted in Figure 4.

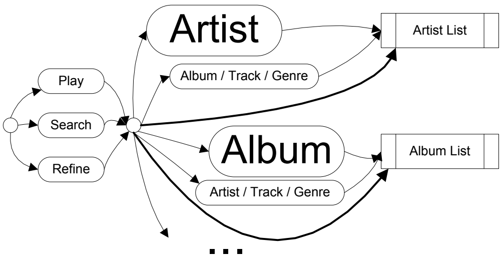

Figure 4: Final Proposed Asymmetric Topology (A1)

We measured the additional SER increase on both well-formed and "incorrect keyword" queries for switching from grammar topology (A0) to (A1) in Table 5. Accommodating the "missing keyword" queries is slightly more costly as the accuracy for the "Genre" queries degraded since the queries "Genre <genre>" sound like the names of some artists.

SER (%)	Overall	Album	Artist	Track	Genre
A0 Well-formed	8.17	6.25	12.15	6.05	14.60
A1 Well-formed	8.37	6.62	12.15	6.29	17.14
A0 Ill-formed	12.38	10.66	15.63	10.99	16.51
A1 Ill-formed	13.32	11.03	15.63	11.39	20.96

Table 5: Additional Cost for Accommodating "Missing Keyword" Queries

3.3. Accuracy Trade-off

We designed the asymmetric grammar topologies (A0 & A1) to have adjustable branch weights assigned to ill-formed commands. This section describes the experiment we performed to demonstrate that our proposed topology conveniently provides a continuous region of operating points. Developers can pick the desired performance for ill-formed commands at the cost they feel comfortable.

As we mentioned, we are very keen on maintaining the claimed accuracy, therefore we first studied the possibility of reducing the SER degradation on correct queries if we are willing to accommodate fewer ill-formed queries.

Figure 5 plots the SER trade-offs for both well-formed queries (as the X-axis) and "Incorrect keyword" queries (as the Y-axis) under different weight configurations (from 0 to 0.2) for the two grammar topologies (A0) and (A1). Notice both the range and the scale of the two axes are different. As expected, Topology (A1) performs slightly worse than (A0) for these two tasks because of its greater flexibility. However, we can significantly reduce the accuracy penalty as we don't have to accommodate that many ill-formed queries. By operating at the 25.5% SER (weight=0.005) instead of the 13.3% SER level, we maintain the 8.02% SER for well-formed queries which is even lower than the original 8.17% we reported for (A0) in Table 3.

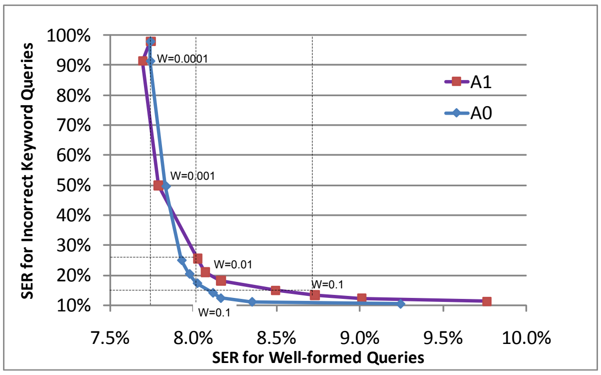

Figure 5: Accuracy Trade-offs

In Figure 6, we plot the SER trade-offs for both well-formed queries and the two types of ill-formed queries (as separate curves) using topology (A1). Since the two experiments shares the same grammar, the operating points on both curves align with each other on the X-axis.

We believe it is reasonable to sacrifice 0.28% absolute SER increase for well-formed queries (equivalent to one more failure out of every 357 queries) while accommodating 70% of both types of ill-formed queries. Accuracy degradation at this level is imperceptible, but our system is now 70% more robust in handling user mistakes.

To make our study complete, we created one more grammar topology (A2), which removes the incorrect keyword branches from topology (A1) and only accommodates the missing keyword queries. If the users only occasionally drop the media keyword, topology (A2) can be used to reduce the performance impact on the SER of well-formed queries even further, as illustrated in Figure 7.

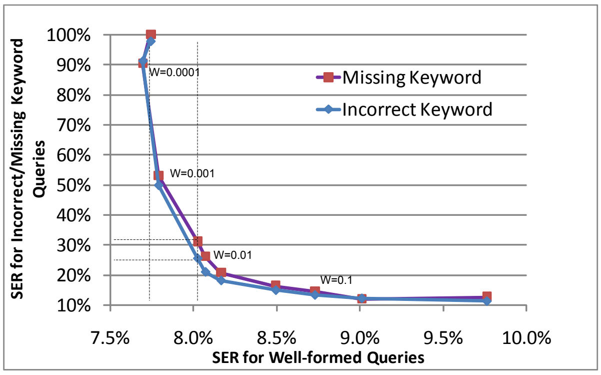

Figure 6: 0.28% Increase in SER on Well-formed Queries Accommodates 70% of Ill-formed Queries

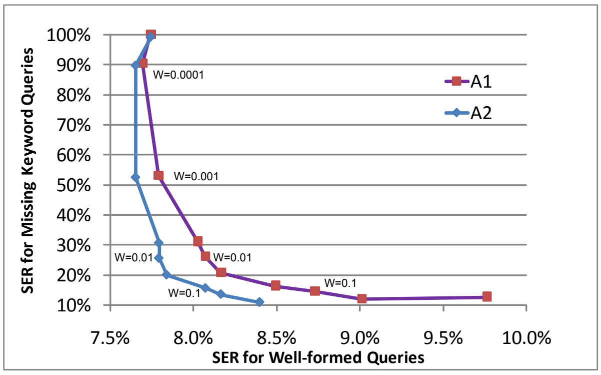

Figure 7: SER Trade-offs For Accommodating "Missing Keyword" Queries Only

4. Two Pass Recognitions

Multi-pass recognition has been used by many speech recognition systems to try capturing different scopes of utterances using different grammars, one at a time to improve SR accuracy. For example, in a speech enabled business search application on mobile phones [5], the users can say the name of the business together with the city and state in the same utterance. The system first tries to recognize the utterance using a national business grammar and a city state name grammar in series, and then backs off to the local business grammar from the specific region once the city/state part is identified.

Since the symmetric topology (S0) achieves higher accuracy than our proposed final topology (A1) for ill-formed queries as shown in Table 3, we investigate the possibility of applying a two pass recognition approach to accommodate well-formed queries in the first recognition pass using the more rigid topology (C), and use a second recognition pass with the more relaxed grammar to accommodate the ill-formed queries, only if topology C fails (or observes a low confidence).

4.1. Feasibility Study

We first examined the confidence scores [6] from the experiments summarized in Table 3. Since we'll need to find a confidence threshold to reject the results from the first recognition pass, we look at the confidence distributions for well-formed queries (especially those which were recognized correctly) and "incorrect keyword" queries, both using the original topology (C).

The accumulated distributions of the utterance level confidence are plotted in Figure 8. Since the SER for well-formed queries is only at 7.74%, the curves for the well-formed query and the correctly recognized well-formed query are very close to each other. We are particularly interested at the point where the curve for ill-formed queries passes the 74.5% coverage because we know we have to reject at least that many ill-formed queries from the first recognition in order to perform as well or better than topology A1 since its SER against "incorrect keyword" queries is only 25.5% (See Figure 5). At first glance, we reject around 9.5% of the correctly recognized well-formed queries. However, the estimation was too pessimistic because many of the well-formed queries with a confidence score lower than that threshold might still get recognized correctly using the relaxed grammar.

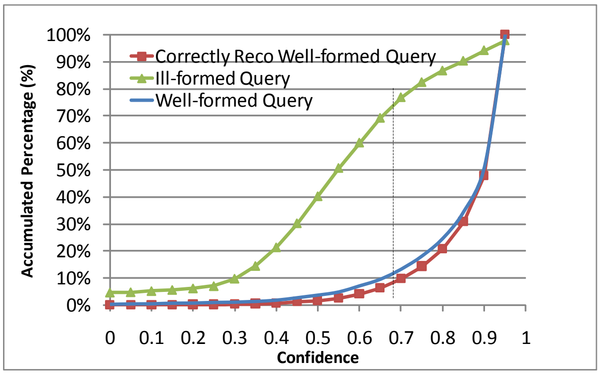

Figure 8: Accumulated Confidence Distribution

Unfortunately, many ill-formed queries which might have been accommodated by the second recognition pass received higher confidence scores than the threshold and were falsely accepted during the first recognition pass. As shown in Figure 9, for incorrect keyword queries the overall performance of the two-pass approach was worse than our single pass approach with grammar topology A1. At the cost of 0.28% absolute SER increase, the two pass approach accommodates only 20% of ill-formed queries, compared with 74.5% by topology A1. In order to accommodate 74.5% of ill-formed queries, the two-pass approach takes an absolute 9.12% SER, which is 4.5 times relatively worse than the 8.02% of topology A1.

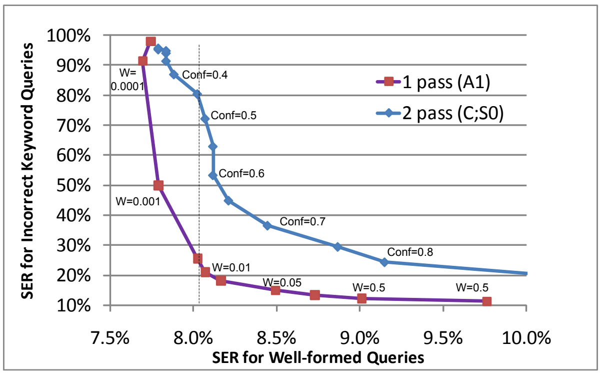

Figure 9: Two Pass Approach Acconmodates Fewer Ill-formed Queries

4.2. Troubles with confidence scores

We attribute most of the performance discrepancy to the quality of the confidence score. The utterance level confidence we were using might be inflated due to the high confidence scores from the command and keyword portion.

As illustrated in the first example in Figure 10, even though the misrecognized word "Digimon" aligns poorly with the partial phrase "18 Miles" with a mediocre word level confidence of 0.5931, the SR engine still reports an utterance level confidence of 0.7860.

Therefore, we examined the feasibility of using the minimum word level confidence to amend the problem. However, we also found many issues of the word level confidences especially for short and function words where the word level confidence is very unreliable. As illustrated in the second example in Figure 10, the SR engine seems to be very uncertain about the confidence of the word "a" (confidence 0.1457), but using the minimum word level confidence is certainly too pessimistic in this case for our application.

```
<EXPECTED> Play Track 18 Miles to Memphis </EXPECTED>
<RECO> Play Track Digimon </RECO>
<CONFIDENCE>
        <PHRASE> 0.7860 </PHRASE>
        <WORDS> 0.9930, 0.9880, 0.5931 </WORDS>
</CONFIDENCE>

<EXPECTED> Play Track A Nightingale Sang In Berkeley Square </EXPECTED>
<RECO> Play Track A Nightingale Sang In Berkeley Square </RECO>
<CONFIDENCE>
        <PHRASE> 0.9322 </PHRASE>
        <WORDS> 0.9831, 0.9585, 0.1457, 0.8668, 0.9730, 0.9850, 0.9894, 0.9916
</WORDS>
</CONFIDENCE>
```

Figure 10: Utterance and Word Level Confidence Scores

5. Conclusions

In this paper, we investigated the potential gap between the claimed accuracy of a voice command system and the perceived accuracy in the field by the actual users because of potential user mistakes. We quantitatively addressed user mistakes and proposed a few new grammar topologies to accommodate ill-formed queries without noticeably degrading the claimed accuracy for users that make little or no mistakes.

By placing a small but non-zero weight on grammar paths that support ill-formed queries, the new grammar topologies balance the accuracy between well-formed and ill-formed queries. We demonstrated the superiority of our approach against the common two pass approach which uses confidence scores.

We also shared a novel workaround on evaluating ill-formed queries without acquiring a new acoustic corpus by swapping metadata and grammar, and a data scrubbing practice of using SR to eliminate the potential artifacts from hand edited waveforms.

We urge system developers to focus more on the overall perceived accuracy to provide a better user experience, even though it is the users that make the mistakes. We don't have concrete statistics from the real users on how bad the situation is, or whether the users can quickly realize their mistakes. However, we have demonstrated it only takes an SER increase of 0.28% to enable our system to handle 70% of user mistakes on our media search application.

6. Acknowledgements

We thank Stefanie Tomko in our automobile research and development group, Jian Wu and Taro Miwa in our speech component group, and Dong Yu in our group for many useful discussions.

7. References

[1]http://www.autoweek.com/apps/pbcs.dll/article?AID=/20071003/FREE/71002006/1528/newsletter01

[2] A. Kun & L. Turner. "Evaluating the project54 speech user interface". In Proc. Pervasive. 2005

[3] Young-In Song, Ye-Yi Wang, Yun-Cheng Ju, Michael Seltzer, Ivan Tashev, Alex Acero, "Voice Search of Structured Media Data", Proceedings of ICASSP. 2009

[4] Tim Paek, Sudeep Gandhe, David Chickering, Yun-Cheng Ju, "Handling Out-of-Grammar Commands in Mobile Speech Interaction Using Backoff Filler Models", Proceedings of SPEECHGRAM. 2007

[5] A. Acero, N. Bernstein, R. Chambers, Y. C. Ju, X. Li, J. Odell, P. Nguyen, O. Scholz and G. Zweig. "Live Search for Mobile: Web Services by Voice on the Cellphone", Proceedings of ICASSP. 2008.

[6] Christopher White, Jasha Droppo, Alex Acero, and Julian Odell, "Maximumentropy confidence estimation for speech recognition," in Proc. ICASSP, 2007.