Stability and Accuracy in Incremental Speech Recognition

Ethan O. Selfridge[†], **Iker Arizmendi**[‡], **Peter A. Heeman**[†], **and Jason D. Williams**[‡]

[†] Center for Spoken Language Understanding, Oregon Health & Science University, Portland, OR
[‡]AT&T Labs – Research, Shannon Laboratory, Florham Park, NJ
{selfridg,heemanp}@ohsu.edu {iker,jdw}@research.att.com

Abstract

Conventional speech recognition approaches usually wait until the user has finished talking before returning a recognition hypothesis. This results in spoken dialogue systems that are unable to react while the user is still speaking. Incremental Speech Recognition (ISR), where partial phrase results are returned during user speech, has been used to create more reactive systems. However, ISR output is unstable and so prone to revision as more speech is decoded. This paper tackles the problem of stability in ISR. We first present a method that increases the *stability* and *accuracy* of ISR output, without adding delay. Given that some revisions are unavoidable, we next present a pair of methods for predicting the stability and accuracy of ISR results. Taken together, we believe these approaches give ISR more utility for real spoken dialogue systems.

1 Introduction

Incremental Speech Recognition (ISR) enables a spoken dialogue system (SDS) to react quicker than when using conventional speech recognition approaches. Where conventional methods only return a result after some indication of user completion (for example, a short period of silence), ISR returns partial phrase results while the user is still speaking. Having access to a real-time stream of user speech enables more natural behavior by a SDS, and is a foundation for creating systems which take a more active role in conversations.

Research by Fink et al.(1998) and Skantze & Schlangen (2009), among others, has demonstrated the efficacy of ISR but has also drawn attention to a significant obstacle to widespread use: partial phrase results are generally unstable and so, as more speech is decoded, are prone to revision. For example, the ISR component in a bus information SDS may return the partial "leaving from Hills", where "Hills" is a neighborhood name. It may then return the revision "leaving from Pittsburgh", which the system must handle gracefully. Given this propensity to revise, a Stability Measure (SM) — likelihood of a partial result remaining unchanged compared to the final result — is necessary for optimal incremental system behavior. Furthermore, since a stable partial may still be inaccurate, a Confidence Measure (CM) — likelihood of partial correctness — is also necessary.

Effective ISR enables systems to participate in more dynamic turn-taking. For instance, these two measures would enable an SDS to identify inaccurate recognition results while the user is still speaking. The SDS could then interrupt and prompt the user to start again. On the other hand, ISR allows systems to handle pauses gracefully. If the SDS recognizes that an utterance is incomplete (though stable and accurate), it could give the user more time to speak before reacting.

We present two contributions specific to the use of ISR. First, we characterize three approaches to ISR which make different trade-offs between stability and the number of partials generated. We then present a novel hybrid approach that combines their strengths to increase

Proceedings of the SIGDIAL 2011: the 12th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 110–119,
Portland, Oregon, June 17-18, 2011. ©2011 Association for Computational Linguistics

stability without adding latency. However, even with this method, some partial results are still later revised. The second contribution of the paper is to present a pair of methods which predict the stability and accuracy of each partial result. These two measures are designed for use in concert by dialogue systems, which must decide whether to act on each partial result in real time.

2 Background and Related Work

We now describe modern speech recognition methodology, the production of partial phrase results, and the advantages and deficiencies of ISR. In this we seek only to provide a topical foundation, and not a comprehensive review.

Most modern speech recognition engines use Hidden-Markov Models and the Viterbi algorithm to decode words from audio. Decoding employs three models: an acoustic model, which assigns probabilities to speech audio given a phone; a lexicon, which specifies phone sequences for a word; and a language model, which specifies the probability of a word sequence. The aim of the decoding process is to find the N most probable word sequences given the audio spoken and these three models.

Two useful but different forms of language models are commonly used in spoken dialogue systems. A *Rule-based Language Model* (RLM) specifies a list of valid sentences which may be recognized, usually via expansion rules. By contrast, a *Statistical Language Model* (SLM) specifies a vocabulary of words, allowing arbitrary sentences to be formed. Both models specify probabilities over their respective sets — RLMs via whole-sentence probabilities, and SLMs via probabilities of short word sequences called N-grams. In an SLM, special word symbols are used to represent the beginning and end of the phrase, so the probability of beginning or ending phrases with words can be modeled.

As speech frames are received, the recognizer builds up a *lattice* which compactly describes the probable sequences of words decoded from the audio. In conventional turn-based speech recognition, decoding continues until the user finishes speaking. Once the user has finished, the engine searches the lattice for the most probable word sequence and returns this to the dialogue manager. By contrast, in ISR the engine inspects the lattice *as it is being built*, and returns *partial* results to the dialogue manager as they become available. A key issue for ISR is that partial results may later be revised, because as more speech is received and the lattice is extended, a different path may become the most probable. In other words, partial results are *unstable* in the sense that they may later be revised. Note that stability is not the same as accuracy: a partial result may be accurate (correct so far) but unstable, because it is later revised. Similarly, a stable result may not be accurate.

In the literature, ISR has been proposed for dialogue systems to enable them to engage in more natural, human-like interactions. Studies have shown that incremental systems react faster than non-incremental ones, and are well-liked by users because of their naturalness (Aist et al., 2007; Skantze and Schlangen, 2009). Aist et al. (2007) found that incremental speech recognition yielded 20% faster task completion. Moreover, adding ISR improved users' satisfaction with the interaction; the authors attributed this improvement to "naturalness": "incremental systems are more like human-human conversation than their non-incremental counterparts." Skantze & Schlangen (2009) observed a similar trend, finding that an incremental system was "clearly preferred" since it "was experienced as more pleasant and human-like", though it did not actually outperform the non-incremental system in a number dictation task.

Some recent work has focused on incremental natural language understanding (NLU). DeVault et al. (2009) showed that when using a relatively small number of semantic possibilities the correct interpretation could be predicted by early incremental results. Schlangen et al. (2009) demonstrated that an incremental reference resolver could identify the correct reference out of 12 more than 50% of the time. This type of NLU can use context and other information to be somewhat resilient to errors, and word recognition inaccuracies may not yield a change in understanding. In this paper we focus on improving accuracy and stability at the word level; we belief that improvements at the word level are likely to improve performance at the understanding level, although we do not evaluate this here.

A number of researchers have described methods for evaluating and improving the stability of ISR results (Baumann et al., 2009; Fink et al., 1998). Baumann, Atterer, & Schlangen spoke directly to stability by comparing partial phrase results against the "final hypothesis produced by the ASR". They show that increasing the amount of "right context" — the amount of speech after the end of the putative partial result — increases the stability of the partials. Fink et al. (1998) also used a right context delay to decrease the word error rate of ISR results.

A key limitation of these past efforts to improve stability is that adding right context necessarily incurs *delay*, which degrades responsiveness and erodes the overall benefits of ISR. Furthermore, past work has not addressed the problem of identifying which partials are likely to be revised. In this paper, we tackle both of these problems. We first present a method for improving stability by considering features of the lattice itself, without incurring the delay associated with adding right context. Additionally, since some partials will still be revised, we then propose a method of scoring the stability of partial speech recognition results.

3 Three approaches to ISR

We now describe three approaches to ISR: Basic, Terminal, and Immortal. Basic ISR simply returns the most likely word sequence observed after some number of speech frames has been decoded (in our case every 3 frames or 30ms). This is the least restrictive approach, and we believe is the method used by recent ISR research.

Terminal ISR, a more restrictive approach, finds a partial result if the most likely path through the (partially-decoded) lattice ends at a *terminal* node in the language model. The intuition is that if a partial result finishes a complete phrase expected by the language model, it is more likely to be stable. The meaning of *terminal* is slightly different for rule-based language models (RLMs) and statistical language models (SLMs). For a rule-based grammar, the terminal node is simply one that ends a valid phrase ('Pittsburgh' in 'leaving from Pittsburgh'). For an SLM, a terminal node indicates that the most likely successor state is the special end-of-sentence symbol. In other words, in an SLM Terminal partial result, the language model assigns the highest probability to ending the phrase.

A third method, Immortal ISR, is the most restrictive method (Spohrer et al., 1980). If all paths of the lattice come together into a node — called an *immortal* node — then the lattice structure before that node will be unchanged by any subsequent decoding. This structure guarantees that the best word sequence prior to an immortal node is stable. Immortal ISR operates identically for both RLMs and SLMs.[1]

To compare these approaches we evaluate their performance. Utterances were extracted from real calls to the Carnegie Mellon "Lets Go!" bus information system for Pittsburgh, USA (Raux et al., 2005; Parent and Eskenazi, 2009). We chose this domain because this corpus is publicly available, and this domain has recently been used as a test bed for dialogue systems (Black et al. , 2010). The AT&T WATSON speech recognition engine was used, modified to output partials as described above (Goffin et al., 2005). We tested these three approaches to ISR on three different recognition tasks. The first two tasks used rule-based language models (RLM), and the third used a statistical language model (SLM).

The two rule-based language models were developed for AT&T "Let's Go" dialogue system, prior to its deployment (Williams et al. , 2010). The first RLM (RLM1) consisted

[1] The choice of search beam size affects both accuracy and the number of immortal nodes produced: a smaller beams yields a sparser lattice with more immortal nodes and lower accuracy; a larger beam yields a richer lattice with fewer immortal nodes and higher accuracy. In this work we used our recognizer's default beam size, which allows recognition to run in less than real time and yields near-asymptotic accuracy for all experiments.

of street and neighborhood names, built from the bus timetable database. The second RLM (RLM2) consisted of just neighborhood names. Utterances to test RLM1 and RLM2 were selected from the corpus provided by Carnegie Mellon to match the expected distribution of speech at the dialogue states where RLM1 and RLM2 would be used. RLM1 was evaluated on a set of 7722 utterances, and RLM2 on 5411 utterances. To simulate realistic use, both RLM test sets were built so that 80% of utterances are in-grammar, and 20% are out-of-grammar. The SLM was a 3-gram trained on a set of 140K utterances, and is tested on a set of 42620 utterances.

In past work, Raux et al. (2005) report word error rates (WERs) of 60-68% on data from the same dialogue system, though on a different set of utterances. By comparison, our SLM yields a WER of 35%, which gives us some confidence that our overall recognition accuracy is competitive, and that our results are relevant.

Table 1 provides a few statistics of the LMs and test sets, including *whole-utterance accuracy*, computed using an exact string match. Results are analyzed in two groups: *All*, where all of the utterances are analyzed, and *Multi-Word (MW)*, where only utterances whose transcribed speech (what was actually said) has more than one word. Intuitively, these utterances are where ISR would be most effective. That said, ISR is beneficial for both short and long utterances — for example, ISR systems can react faster to users regardless of utterance length.

ISR was run using each of the three approaches (Basic, Terminal, Immortal) in each of the three configurations (RLM1, RLM2, SLM). The mean number of partials per utterance is shown in Table 2. For all ISR methods, the more flexible SLM produces more partials than the RLMs. Also as expected, multi-word utterances produce substantially more partials per utterance than when looking at the entire utterance set. The Basic approach produces nearly double the number of partials than Terminal ISR does, and Immortal ISR production highlights its primary weakness: in many utterances, no immortal nodes are found. Given this however, immortal node occurrence is directly related to the number of words, as indicted by the greater number of immortal partials in multi-word utterances.

Table 1: Statistics for Recognition Tasks. In all tables, *All* refers to all utterances in a test set, and *MW* refers to the subset of multi-word utterances in a test set.

	RLM1	RLM2	SLM
Num. Utts All	7722	5411	42620
Num. Utts MW	3213	1748	20396
Words/Utt All	1.7	1.5	2.3
Words/Utt MW	2.8	2.6	3.8
Utt. Acc. All.	50 %	60 %	62 %
Utt. Acc. MW	53 %	56 %	44 %

Stability is assessed by comparing the partial to the final recognition result. For simplicity, we restrict our analysis to 1-Best hypotheses. If the *partial* 1-Best hypothesis is a prefix (or full exact match) of the *final* 1-Best hypothesis then it is considered stable. For instance, if the partial 1-Best hypothesis is "leaving from Forbes" then it would be stable if the final 1-Best is "leaving from Forbes" or "leaving from Forbes and Murray" but not if it is "from Forbes and Murray" or "leaving". Accuracy is assessed similarly except that the transcribed reference is used instead of the final recognition result.

We report stability and accuracy in Table 3. Immortal partials are excluded from stability since they are guaranteed to be stable. The first four rows report stability, and the second six report accuracy. The results show that Terminal Partials are relatively unstable, with 23%-

Table 2: Average Number of Partials per utterance

ISR	Group	RLM1	RLM2	SLM
Basic	All	12.0	9.9	11.6
	MW	14.6	12.3	29.7
Terminal	All	5.4	3.3	6.2
	MW	6.4	4.1	8.8
Immortal	All	0.22	0.32	0.55
	MW	0.42	0.67	0.63

Table 3: Stability and Accuracy Percentages

ISR	Group	RLM1	RLM2	SLM
Stability				
Basic	All	10 %	11 %	7 %
	MW	14 %	15 %	9 %
Terminal	All	23 %	31 %	37 %
	MW	20 %	28 %	36 %
Accuracy				
Basic	All	9 %	1 %	5 %
	MW	11 %	13 %	6 %
Terminal	All	13 %	21 %	24 %
	MW	12 %	17 %	21 %
Immortal	All	91 %	93 %	55 %
	MW	90 %	90 %	56 %

37% of partials being stable, and that their stability drops off when looking at multi-word utterances. SLM stability seems to be somewhat higher than that of the RLM. Basic partials are even more unstable (about 10% of partials are stable), with extremely low stability for the SLM. Unlike Terminal ISR, their stability grows when only multi-word utterances are analyzed, though the maximum is still quite low.

The results also show that partials are always less accurate than they are stable, indicating that not all stable partials are accurate. Immortal partials are rare, but when they are found, they are much more accurate than Terminal or Basic partials. The RLM accuracy is very high, and we suspect that immortal nodes are correlated with utterances which are easier to recognize. Terminal ISR is far more accurate than Basic ISR for all of the utterances, but its improvement declines for multi-word RLMs.

We have shown three types of ISR: Basic, Terminal and Immortal ISR. While Basic and Terminal ISR are both highly productive, Terminal ISR is far more stable and accurate than Basic. Furthermore, there are far more Basic partials than Terminal partials, implying that the dialogue manager would have to handle more unstable and inaccurate partials more often. Given this, Terminal ISR is a far better "productive ISR" than the Basic method. Taking production and stability together, there is a double dissociation between Terminal and Immortal ISR. Terminal partials are over produced and relatively unstable. Furthermore, they are even less stable when the transcribed reference is greater than one word. On the other hand, Immortal partials are stable and quite accurate, but too rare for use alone. By integrating the Immortal Partials with the Terminal ones, we may be able to increase the stability and accuracy overall.

Table 4: Lattice-Aware ISR (LAISR) Example

1-best	Partial Type
yew	Terminal
sarah	Terminal
baum	Terminal
dallas	Terminal
downtown	Terminal
downtown	Immortal
downtown pittsburgh	Terminal
downtown pittsburgh	Immortal

4 Lattice-Aware ISR (LAISR)

We introduce *Lattice-Aware ISR* (LAISR — pronounced "laser"), that integrates Terminal and Immortal ISR by allowing both types of partials to be found. The selection procedure works by first checking for an Immortal partial. If one is not found then it looks for a Terminal. Redundant partials are returned when the partial type changes. An example recognition is shown in Table 4. Notice how the first four partials are completely unstable. This is very common, and suppressing this noise is one of the primary benefits of using more right context. Basic ISR has even more of this type of noise.

LAISR was evaluated on the three recognition tasks described above (see Table 5). The first two rows show the average number of partials per utterance for each task and utterance group. Unsurprisingly, these numbers are quite similar to Terminal ISR. The stability percentage of LAISR is shown in the second two rows. For all the utterances, there appears to be a very slight improvement when compared to Terminal ISR in Table 3. The improvement increases for MW utterances, with LAISR improving over

Table 5: Lattice-Aware ISR Stats

Partials per Utterance			
	RLM1	RLM2	SLM
All	5.6	3.5	6.7
MW	6.7	4.5	9.6
Stability Percentage			
All	24 %	33 %	40 %
MW	24 %	35 %	41 %
Accuracy Percentage			
All	15 %	23 %	26 %
MW	16 %	22 %	24 %

Terminal ISR by 4–7 percentage points. This is primarily because there is a higher occurrence of Immortal partials as the utterance gets longer. Accuracy is reported in the final two rows. Like the previous ISR methods described, the accuracy percentage is lower than the stability percentage. When compared to Terminal ISR, LAISR accuracy is slightly higher, which confirms the benefit of incorporating immortal partials with their relatively high accuracy. To be useful in practice, it is important to examine *when* in the utterance ISR results are being produced. For example, if most of the partials are returned towards the end of utterances, than ISR is of little value over standard turn-based recognition. Figure 1 shows the percent of partials returned from the start of speech to the final partial for MW utterances using the SLM. This figure shows that partials are returned rather evenly over the duration of utterances. For example, in the first 10% of duration of each utterance, about 10% of all partial results are returned. Figure 1 also reports the stability and accuracy of the partials returned. These numbers grow as decoding progresses, but shows that mid-utterance results do yield reasonable accuracy: partials returned in the middle of utterances (50%-60% duration) have an accuracy of near 30%, compared to final partials 47% percent.

For use in a real-time dialogue system, it is also important to assess *latency*. Here we define latency as the difference in (real-world) time between (1) when the recognizer receives the last frame of audio for a segment of speech, and (2) when the partial that covers that segment of speech is returned from the recognizer. Measuring latencies of LAISR on each task, we find that RLM1 has a median of 0.26 seconds and a mean of 0.41s; RLM2 has a median of 0.60s and a mean of 1.48s; and SLM has a median of 1.04s and a mean of 2.10s. Since reducing latency was not the focus on this work, no speed optimizations have been made, and we believe that straightforward optimization can reduce these latencies. For example, on the SLM, simply turning off N-Best processing reduces the median latency to 0.55s and the mean to 0.79s. Human reaction time to speech is roughly 0.20 seconds (Fry, 1975), so even without optimization the RLM latencies are not far off human performance.

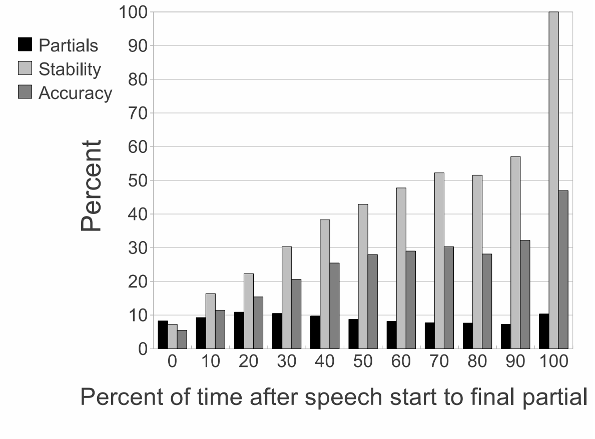


Figure 1: Percent of LAISR partials returned from the start of detected speech to the final partial using the SLM. The percentage of partials returned that are stable/accurate are also shown.

In sum, LAISR produces a steady stream of partials with relatively low latency over the course of recognition. LAISR has higher stability and accuracy than Terminal ISR, but its partials are still quite unstable and inaccurate. This means that in practice, dialogue systems will need to make important decisions about which partials to use, and which to discard. This need motivated us to devise techniques for predicting when a partial is stable, and when it is accurate, which we address next.

Table 6: Equal Error Rates: Significant improvements in bold. Basic at $p < 0.016$, Terminal at $p < 0.002$, and LAISR at $p < 0.00001$

		All			Multi-Word		
Stability Measure (SM) Equal Error Rate							
		RLM 1	RLM 2	SLM	RLM 1	RLM 2	SLM
Basic	WATSON Score	13.3	13.3	12.8	15.6	16.4	15.2
	Regression	**10.7**	**11.3**	**12.3**	**13.2**	**15.2**	15.1
Terminal	WATSON Score	24.3	29.1	34.4	26.6	26.0	34.1
	Regression	**19.7**	**26.5**	**26.5**	**23.0**	24.3	**24.7**
LAISR	WATSON Score	24.7	29.3	35.0	24.0	27.0	35.3
	Regression	**19.2**	**25.6**	**25.0**	**18.4**	**23.3**	**22.7**
Confidence Measure (CM) Equal Error Rate							
Basic	WATSON Score	11.3	11.7	9.9	14.1	14.0	11.6
	Regression	**9.8**	**9.8**	9.7	**12.3**	**12.9**	**11.0**
Terminal	WATSON Score	15.1	21.1	30.6	15.7	17.4	29.3
	Regression	**11.7**	**16.8**	**20.8**	**12.1**	**14.5**	**18.4**
LAISR	WATSON Score	15.8	21.8	32.3	18.4	19.5	31.8
	Regression	**11.6**	**16.6**	**21.0**	**11.6**	**14.2**	**18.7**

5 Stability and Confidence Measures

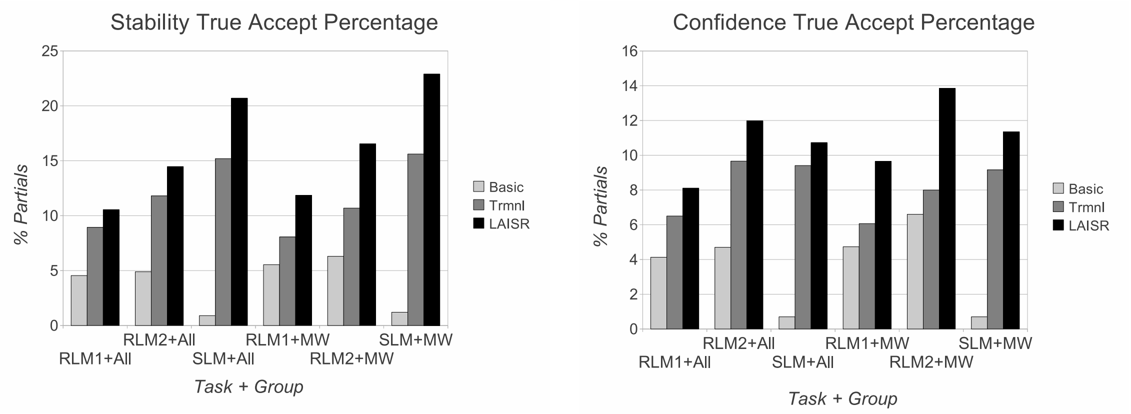
As seen in the previous section, partial speech recognition results are often revised and inaccurate. In order for a dialogue system to make use of partial results, measures of both stability and confidence are crucial. A Stability Measure (SM) predicts whether the current partial is a prefix or complete match of the final recognition result (regardless of whether the final result is accurate). A Confidence Measure (CM) predicts whether the current partial is a prefix or complete match of what the user actually said. Both are useful in real systems: for example, if a partial is likely stable but unlikely correct, the system might interrupt the user and ask them to start again.

We use logistic regression to learn separate classifiers for SM and CM. Logistic regression is appealing because it is well-calibrated, and has shown good performance for whole-utterance confidence measures (Williams and Balakrishnan, 2009). For this, we use the BXR package with default settings (Genkin et al., 2011). For Terminal and Basic ISR we use 11 features: the raw WATSON confidence score, the individual features which affect the confidence score, the normalized cost, the normalized speech likelihood, the likelihoods of competing models, the best path score of word confusion network (WCN), the length of WCN, the worst probability in the WCN, and the length of N-best list. For LAISR, four additional features are used: three binary indicators of whether the partial is Terminal, Immortal or a Terminal following an Immortal, and one which gives the percentage of words in the hypothesis that are immortal.

We built stability and confidence measures for Basic ISR, Terminal ISR, and LAISR. Each of the three corpora (RLM1, RLM2, SLM) was divided in half to form a train set and test set. Regression models were trained on *all* utterances in the train set. The resulting models were then evaluated on both All and MW utterances. As a baseline for both measures, we compare to AT&T WATSON's existing confidence score. This score is used in numerous deployed commercial applications, so we believe it is a fair baseline. Although the existing confidence score is designed to predict accuracy (*not stability*), there is no other existing mechanism for predicting stability.

We first report "equal error rate" for the measures (Table 6). Equal error rate (EER) is the sum of false accepts and false rejects at the rejec-

Figure 2: True accept percentages for stability measure (a) and confidence measure (b), using a fixed false accept rate of 5%. LAISR yields highest true accept rates, with $p < 0.0001$ in all cases.



(a) Stability measure

(b) Confidence measure

tion threshold for which false accepts and false rejects are equal. Equal error rate is a widely used metric to evaluate the quality of scoring models used for accept/reject decisions. A perfect scoring model would yield an EER of 0. For statistical significance we use χ^2 contingency tables with 1 degree of freedom. It is inappropriate to compare EER across ISR methods, since the total percentage of stable or accurate partials significantly effects the EER. For example, Basic ISR has relatively low EER, but this is because it also has a relatively low number of stable or accurate partials.

The top six rows of Table 6 show EER for the Stability Measure (SM). The left three columns show results on the entire test set (all utterances, of any length). On the whole, the SM outperforms the WATSON confidence scores, and the greatest improvement is a 10.0 point reduction in EER for LAISR on the SLM task. The right three columns show results on only multi-word (MW) utterances. Performance is similar to the entire test set, with a maximum EER reduction of 12.6 percent. The SLM MW performance is interesting, suggesting that it is easier to predict stability after at least one word has been decoded, possibly due to higher probability of immortal nodes occurring. This suggests there would be benefit in combining our method with past work that adds right-context, perhaps using more context early in the utterance. This idea is left for future work.

The bottom six rows show results for the Confidence Measure (CM). We see that that even when comparing our CM against the WATSON confidence scores, there is significant improvement, with a maximum of 13.1 for LAISR in the MW SLM task.

The consistent improvement shows that logistic regression is an effective technique for learning confidence and stability measures. It is most powerful when combined with LAISR, and only slightly less so with Terminal. Furthermore, though the gains are slight, it is also useful with Basic ISR, which speaks to the generality of the approach.

While equal error rate is useful for evaluating discriminative ability, when building an actual system a designer would be interested to know how often the correct partial is accepted. To evaluate this, we assumed a fixed false-accept rate of 5%, and report the resulting percentage of partials which are correctly accepted (true-accepts). Results are shown in Figure 1. LAISR accepts substantially more correct partials than other methods, indicating that LAISR would be more useful in practice. This result also shows a synergy between LAISR and our regression-based stability and confidence measures: not only does LAISR improve the fraction of stable and correct partials, but the regression is able to identify them better than for Terminal ISR. We believe this shows the usefulness of the additional lattice features used by the regression model built on LAISR results.

6 Discussion and Conclusion

The adoption of ISR is hindered by the number of revisions that most partials undergo. A number of researchers have proposed the use of right context to increase the stability of partials. While this does increase stability, it mitigates the primary gain of ISR: getting a relatively real-time stream of the user's utterance. We offer two methods to improve ISR functionality: the integration of low-occurring Immortal partials with higher occurring Terminal partials (LAISR), and the use of logistic regression to learn stability and confidence measures.

We find that the integrative approach, LAISR, outperforms Terminal ISR on three recognition tasks for a bus timetable spoken dialogue system. When looking at utterances with more than one word this difference becomes even greater, and this performance increase is due to the addition of immortal partials, which have a higher occurrence in longer utterances. This suggests that as dialogue systems are used to process multi-phrasal utterances and have more dynamic turn-taking interactions, immortal partials will play an even larger roll in ISR and partial stability will further improve.

The Stability and Confidence measures both have lower Equal Error Rates than raw recognition scores when classifying partials. The improvement is greatest for LAISR, which benefits from additional features describing lattice structure. It also suggests that other incremental features such as the length of right context could be useful for predicting stability. The higher number of True Accept partials by LAISR indicates that this method is more useful to a dialogue manager than Basic or Terminal ISR. Even so, for all ISR methods there are still more useful stable partials than there are accurate ones. This suggests that both of these measures are important to the downstream dialogue manager. For example, if the partial is predicted to be stable but not correct, than the agent could possibly interrupt the user and ask them to begin again.

There are a number of avenues for future work. First, this paper has examined the *word* level; however dialogue systems generally operate at the *intention* level. Not all changes at the word level yield a change in the resulting intention, so it would be interesting to apply the confidence measure and stability measures developed here to the (partial) intention level. These measures could also be applied to later stages of the pipeline – for example, tracking stability and confidence in the dialogue state resulting from the current partial intention. Features from the intention level and dialogue state could be useful for these measures – for instance, indicating whether the current partial intention is incompatible with the current dialogue state.

Another avenue for future work would be to apply these techniques to non-dialogue real-time ASR tasks, such as transcription of broadcast news. Confidence and stability measures could be used to determine whether/when/how to display recognized text to a viewer, or to inform down-stream processes such as named entity extraction or machine translation.

Of course, an important objective is to evaluate our Stability and Confidence Measures with LAISR in an actual spoken dialogue system. ISR completely restructures the conventional turn-based dialogue manager, giving the agent the opportunity to speak at any moment. The use of reinforcement learning to make these turn-taking decisions has been shown in a small simulated domain by Selfridge and Heeman (2010), and we believe this paper builds a foundation for pursuing these ideas in a real system.

Acknowledgments

Thanks to Vincent Goffin for help with this work, and the anonymous reviewers for their thoughtful suggestions and critique. We acknowledge funding from the NSF under grant IIS-0713698.

References

G. Aist, J. Allen, E. Campana, C. Gallo, S. Stoness, Mary Swift, and Michael K. Tanenhaus. 2007. Incremental understanding in human-computer dialogue and experimental evidence for advantages over nonincremental methods. In *Proc. DECALOG*, pages 149–154.

T. Baumann, M. Atterer, and D. Schlangen. 2009. Assessing and improving the performance of speech recognition for incremental systems. In *Proc. NAACL: HLT*, pages 380–388.

A. Black, S. Burger, B. Langner, G. Parent, and M. Eskenazi, 2010. Spoken dialog challenge 2010, In *Proc. Workshop on Spoken Language Technologies (SLT), Spoken Dialog Challenge 2010 Special Session.*

David DeVault, Kenji Sagae, and David Traum. 2009. Can i finish? learning when to respond to incremental interpretation results in interactive dialogue. In *Proc. SIGdial 2009 Conference*, pages 11–20,

G.A. Fink, C. Schillo, F. Kummert, and G. Sagerer. 1998. Incremental speech recognition for multimodal interfaces. In *Industrial Electronics Society, 1998. IECON'98* volume 4, pages 2012–2017.

D.B. Fry. 1975. Simple reaction-times to speech and non-speech stimuli.. *Cortex* volume 11, number 4, page 355.

A. Genkin, L. Shenzhi, D. Madigan, and DD. Lewis. 2011. Bayesian logistic regression. *http://www.bayesianregression.org.*

V. Goffin, C. Allauzen, E. Bocchieri, D. Hakkani-Tur, A. Ljolje, S. Parthasarathy, M. Rahim, G. Riccardi, and M. Saraclar. 2005. The AT&T WATSON speech recognizer. In *Proc. of ICASSP*, pages 1033–1036.

G. Parent and M. Eskenazi. 2009. Toward Better Crowdsourced Transcription: Transcription of a year of the Let's Go Bus Information System Data. *Proc. of Interspeech 2005, Lisbon, Portugal.*

A. Raux, B. Langner, D. Bohus, A.W. Black, and M. Eskenazi. 2005. Lets go public! taking a spoken dialog system to the real world. In *Proc. of Interspeech 2005.*

D. Schlangen, T. Baumann, and M. Atterer. 2009. Incremental reference resolution: The task, metrics for evaluation, and a Bayesian filtering model that is sensitive to disfluencies. In *Proc. SIGdial*, pages 30–37.

E.O. Selfridge and P.A. Heeman. 2010. Importance-Driven Turn-Bidding for spoken dialogue systems. In *Proc. of ACL 2010*, pages 177–185.

G. Skantze and D. Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proc. EACL 2009*, pages 745–753

J.C. Spohrer, PF Brown, PH Hochschild, and JK Baker. 1980. Partial traceback in continuous speech recognition. In *Proc. of the IEEE International Conference on Cybernetics and Society.*

J.D. Williams, I. Arizmendi and A. Conkie. 2010. Demonstration of AT&T "Let's Go": A production-grade statistical spoken dialog system. In *Proc Demonstration Session at IEEE Workshop on Spoken Language Technology*

J.D. Williams and S. Balakrishnan. 2009. Estimating probability of correctness for ASR N-Best lists. In *Proc. of SIGdial 2009*, pages 132–135.