Recognising Conversational Speech: What an Incremental ASR Should Do for a Dialogue System and How to Get There

Timo Baumann, Casey Kennington, Julian Hough and David Schlangen

Abstract Automatic speech recognition (ASR) is not only becoming increasingly accurate, but also increasingly adapted for producing timely, incremental output. However, overall accuracy and timeliness alone are insufficient when it comes to interactive dialogue systems which require stability in the output and responsivity to the utterance as it is unfolding. Furthermore, for a dialogue system to deal with phenomena such as disfluencies, to achieve deep understanding of user utterances these should be preserved or marked up for use by downstream components, such as language understanding, rather than be filtered out. Similarly, word timing can be informative for analyzing deictic expressions in a situated environment and should be available for analysis. Here we investigate the overall accuracy and incremental performance of three widely used systems and discuss their suitability for the aforementioned perspectives. From the differing performance along these measures we provide a picture of the requirements for incremental ASR in dialogue systems and describe freely available tools for using and evaluating incremental ASR.

1 Introduction

Incremental ASR is becoming increasingly popular and is available both commercially and as open-source. Given this recent development of systems, the question arises as

Timo Baumann
Natural Language Systems Group, Department of Informatics, Universität Hamburg, Germany, e-mail: baumann@informatik.uni-hamburg.de.

Casey Kennington, Julian Hough, and David Schlangen
Dialogue Systems Group, Faculty of Linguistics and Literature and CITEC, Bielefeld University, Germany, e-mail: ckennington@citec.uni-bielefeld.de, julian.hough@uni-bielefeld.de, david.schlangen@uni-bielefeld.de.

This work is supported by a Daimler and Benz Foundation PostDoc Grant to the first author, by the BMBF KogniHome project, DFG DUEL project (grant SCHL 845/5-1) and the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University.

to how they perform and compare to each other, not just in terms of utterance-final accuracy but also in terms of their *incremental* performance.

For a spoken dialogue system (SDS) consuming ASR output, incrementally receiving partial results for an on-going utterance means the system can start processing words before the utterance is complete, leading to advantages such as quicker responses, better interactive behaviour and dialogue management, more efficient database queries, and compensation for inefficient downstream processors such as slow robot actuators – see Schlangen and Skantze (2011) for an overview. SDSs that process incrementally produce behaviour that is perceived to be more natural than systems that use the traditional turn-based approach (Aist et al. 2006; Skantze and Schlangen 2009; Skantze and Hjalmarsson 2010; Asri et al. 2014), offer a more human-like experience for users (Edlund et al. 2008), and are more satisfying to interact with than non-incremental systems (Aist et al. 2007).

Metrics have been proposed to evaluate incremental performance for ASR (Baumann, Atterer, and Schlangen 2009; Selfridge et al. 2011; McGraw and Gruenstein 2012), which we build on in this paper. We also deal with evaluating an incremental ASR's performance on difficult phenomena from *conversational speech* such as disfluency. In this paper we investigate these challenges, firstly by outlining suitable evaluation criteria for incremental ASRs for dialogue systems, then investigating how off-the-shelf ASRs deal with speech from participants in a task-oriented dialogue domain, both with and without training on in-domain data. We present findings using our criteria to help SDS builders in their decision as to which ASR is suitable for their domain. The alternative ASR engines that are evaluated in this paper are all accessible in a uniform way with the freely available InproTK[1] (Baumann and Schlangen 2012), as is the evaluation toolbox InTELiDa[2] that we use.

2 The challenge of interactive, conversational speech

While many current SDSs claim to deal with spontaneous speech, this is often in the form of voice commands that do not require a fast verbal response, with some exceptions (Skantze and Schlangen 2009; Skantze and Hjalmarsson 2010). When using voice commands, it has been established that people use more controlled, fluent and restricted speech than when in a human-only dialogue (Shriberg 1996), with users often defaulting to what Fischer (2006) calls 'Computer Talk'.

We argue ASR evaluation currently does not focus on the challenge of interactive speech as required for a highly interactive SDS. While popular dictation evaluation domains such as the spoken Wall Street Journal (Paul and Baker 1992) are clearly unsuitable, even the more SLU (Spoken Language Understanding)-based benchmarks such as the ATIS (Airline Travel Information Systems) corpus and other genres

[1] `http://bitbucket.org/inpro/inprotk`

[2] `http://bitbucket.org/inpro/intelida`

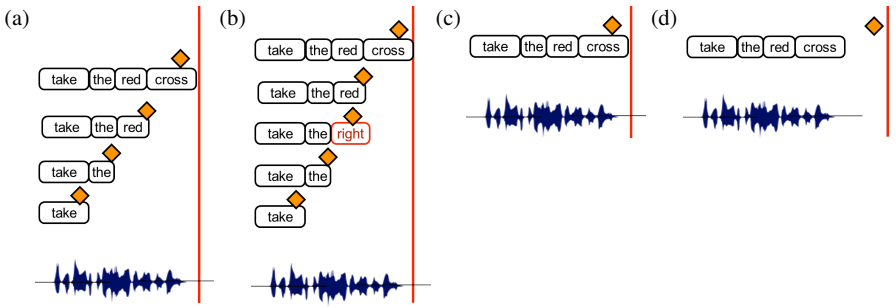

Fig. 1 Incrementality in ASR: vertical line indicates current time, diamond the time of update. (a) perfect output, (b) unstable output, (c) non-incremental but timely, (d) non-incremental and latent.

mentioned in Morbini et al. (2013)'s ASR analysis do not meet the demands of ASR for high levels of interactivity and responsiveness.

3 Desiderata for incremental ASR for interactive SDSs

To address the challenge of interactive, conversational speech, here we briefly set out requirements for ASR for its suitability for interactive SDSs.

3.1 Incrementality and timing information

In addition to being timely and accurate in terms of the final output at the end of an utterance, we would like timeliness and accuracy on the word level from an ASR. In Figure 1 we demonstrate the qualities needed by representing the evolution of hypotheses made by a system over time, going from bottom to top, for the reference transcription 'take the red cross': (a) is the ideal behaviour as it produces fully incremental output which is completely accurate, occasionally predicting the word before it is over, whilst the failings in (b), (c) and (d) give us the incremental desiderata of *stability of output*, *word-by-word incremental output* and *timeliness of output*. Metrics and tools for measuring these incremental qualities will be described in Section 4.

Another factor of situated conversational speech are deictic references that, in a fast-moving environment, can only be interpreted correctly if the timing of deictic references (and possibly co-occurring pointing gestures) is available for analysis. It is thus crucial that an ASR provide, in a timely manner, *timing of the recognized words*.

3.2 Suitability for disfluency

One principal feature strikingly absent from Computer Talk but abundant in human conversational speech is disfluency. Within the larger goal of incorporating understanding of disfluent behaviour to dialogue systems (Ginzburg et al. 2014), we require an ASR to detect all words in speech repairs, preserving the elements of the well-established structure in (1) from Meteer et al. (1995)'s mark-up.

(1) John $\underbrace{[\text{ likes}}_{\text{reparandum}}$ + $\underbrace{\{\text{F uh}\}}_{\text{interregnum}}$ $\underbrace{\text{loves }]}_{\text{repair}}$ Mary

There is evidence that repairs are reasoned with on an incredibly time-critical level in terms of understanding (Brennan and Schober 2001) and there are clear examples of the reparandum being needed to calculate meaning – such as in (2) and (3) where semantic processing access to "the interview" is required to resolve the anaphoric "it" and "the oranges" is required to resolve "them". If an incremental disfluency detector such as Hough and Purver (2014) is to work in a live system, all words within a disfluency must become available in the ASR output, and not be filtered out.

(2) " [the interview, was + {. . . } it was] all right." (Clark 1996)

(3) "have the engine [take the oranges to Elmira, + { um, I mean, } take them to Corning] " (Core and Schubert 1999)

Filled pauses'um' and 'uh' can be considered English words in terms of their meaning in conversation (Clark and Fox Tree 2002) and transcribers can reliably transcribe them. While they can form interregna as in (1), isolated, non-repair filled pauses can indicate forward-looking trouble from conversation participants (Ginzburg, Fernández, and Schlangen 2014). These should therefore not be filtered out during speech recognition if we are to build truly interactive systems.

Given this motivation, in additional to good incremental properties, we would also like an ASR to exhibit *preservation of disfluent material*, that is, we would prefer word hypotheses that are useful for disfluency detection and processing, with no filtering out of reparanda and filled pauses.

4 Evaluation Metrics

To address the desiderata we split our evaluation methods into accuracy, timing and evolution of hypotheses over time. Incremental metrics are provided by the InTELiDa toolbox (Malsburg, Baumann, and Schlangen 2009).

4.1 Utterance-level Accuracy and Disfluency Suitability

We use standard Word Error Rate (WER) of the final (non-incremental) hypothesis. Incremental ASR cannot reliably outperform the accuracy of non-incremental systems, hence its utterance-final quality is what matters most. To measure accuracy on disfluencies, we filter all filled pauses and all reparanda from the transcripts (leaving only the repair phases), so the standard reference 'John likes uh loves Mary' becomes 'John loves Mary' and compare WER before and after filtering. This is in order to find how much disfluent material is recovered (which would result in worse performance on the filtered reference) or whether the ASR itself filters disfluencies accurately (in which case the performance would improve on the filtered reference). **WER disfluency gain** is simply: WER *on disfluency filtered original transcript* – WER *on original transcript*. For preservation of disfluent material, the higher this gain the better. However for accuracy of filtering out disfluency, the lower the better.

4.2 Timing

Following Baumann, Buß, and Schlangen (2011) we use the First Occurrence (FO) and Final Decision (FD) measures to investigate timeliness, where:

FO is the time between the (true) beginning of a word and the *first* time it *occurs* in the output (regardless if it is afterwards changed). In Figure 1, (c) and (d) would perform poorly using this metric, in particular for 'take' which is reported only long after it has been spoken.

FD is the time between the (true) end of a word and the time when the recognizer *decides* on the word, without later revising it anymore. If an ASR correctly guesses a word before it is over, the value will be negative. Often, FD occurs simultaneously with FO. If not, a word is revised and later returned to, which can be a frequent occurrence at word boundaries.

Timeliness can only be measured for words that are correctly recognized or at least appear in the final output of the recognizer and timing distributions are reported below. FO and FD measure *when words are recognized*, but not how well-aligned these are to the actual timing of the word in the audio. However, our impression is that recognizers which report such timing information are very accurate (on the order of centiseconds). Thus, the availability of timing is mostly a binary decision and depends on the recognizer's interface.

4.3 Diachronic Evolution

The diachronic evolution of hypotheses is relevant to capture *how often* consuming processors have to re-consider their output and for *how long* hypotheses are likely to still change. We have previously used *Edit Overhead* the proportion of unnecessary

edits during hypothesis building, to account for the former. However, we disregard this aspect in the present work, as EO is mostly measuring computational overhead and there are effective measures to reduce EO (Baumann, Atterer, and Schlangen 2009).

We instead focus on the stability of hypotheses (Selfridge et al. 2011), which measures the 'temporal extent' of edits. For words that are added and later revoked or substituted we measure the "survival time" and report aggregated plots of **word survival rate** (**WSR**) after a certain age. These statistics can be used to estimate the likelihood of the recognizer being committed to a word during recognition.

5 Evaluation domain: Pentomino puzzle playing dialogue

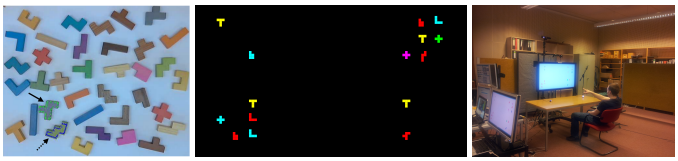

Fig. 2 Example game scenes and collection setup used in collecting Pentomino interaction data.

The evaluations below make use of recorded human-human dialogue, and also interactions between humans and (wizard controlled) SDSs, where participants were instructed to play simple games with the "systems". In all cases, the games made use of geometric *Pentomino* puzzle tiles where participants referred to and instructed the systems or human interlocutors to manipulate the orientation and placement of those tiles. The interactions were all collected and utterances were segmented and transcribed. The corpora were originally described, respectively, in Fernández, Lucht, and Schlangen (2007), Kousidis, Kennington, and Schlangen (2013), and Kennington and Schlangen (2015). We make use of two sets of data in German and English. The German data yields 13,063 utterances (average length of 5 words; std 6.27) with a vocabulary size of 1,988. Example game scenes are shown in Figure 2 and example utterances (with English glosses) are given in Examples (4), and (5) below. We use the German data for training and evaluating ASR models explained in Section 6.2. We also use English data (both UK and US) from this domain yielding 686 telephone-mediated utterances (6,157 words) for evaluating existing English models, as explained in Section 6.1.

(4) a. *drehe die Schlange nach rechts*
b. rotate the snake to the right

(5) a. *dann nehmen wir noch das zw- also das zweite t das oben rechts ist ... aus dieser gruppe da da möchte ich gern das gelbe t haben ... ja*
b. then we take now the se- so the second t that is on the top right ... out of this group there I would like to have the yellow t ... yes

6 Evaluation of three ASR systems: Google, Sphinx-4, and Kaldi

6.1 Experiment 1: Off-the-shelf models for a dialogue domain

In our first experiment we do not train or adapt any of our ASR systems but evaluate their off-the-shelf performance (as in Morbini et al. 2013, but including incremental performance). We evaluate on 686 utterances from the English data explained above.

6.1.1 Systems

We evaluate Sphinx-4 (Walker et al. 2004) with most recent general AM and LM (version 5.2 PTM) for (US-)English, Google's web-based ASR API (Schalkwyk et al. 2010) (in the US-English setting) and Kaldi (Povey et al. 2011), for which we use the English Voxforge recipe (57,474 training utterances, avg 9.35 words per utterance, presumably dominated by US-English). We choose *Google* as the state-of-the-art ASR available via a Web-interface. We use *Sphinx-4* because it has previously been adapted for incremental output processing (Baumann, Atterer, and Schlangen 2009) and Kaldi as an open-source speech recognition system that is growing in popularity and has incremental capabilities (Plátek and Jurčíček 2014).

Google partial results can consist of multiple segments, each of which is given a stability estimate (McGraw and Gruenstein 2012). In practice, Google only returns stabilities of 1 % or 90 % (for both German and English). While incremental results are 1-best, the final (non-incremental) result contains multiple alternatives, with a confidence measure for the first (presumably most likely) alternative. This final hypothesis appears to make use of post-incremental re-scoring or re-ranking. While this is obviously intended to optimize the result quality (SER or WER), it means that incremental results are just a 'good guess' as to what the final result will be, with implications for the timing metrics as reported and discussed below.

We implemented multiple options for interpreting the Google output:

stable use only those segments which have a high stability (we use a threshold of $> 50\%$, but estimates as reported by Google are essentially binary),

quick use all segments, including the material with low stability,

sticky ignore the re-ranking from Google and choose the final hypothesis that best matches the previous 1-best incremental result (as generated by the **quick** setting). This setting is expected to result in lower non-incremental performance.

6.1.2 Non-incremental quality and disfluency suitability

WER results across the reference variants are shown in Table 1. *Google-API* clearly outperforms the other systems. However, its WER does not degrade on disfluency-filtered transcripts as much as *Sphinx-4*, which has the largest WER disfluency gain of 4.70, showing it is preserving the disfluent material the most. Manual inspection

System	US English speakers		All English speakers	
	WER (all)	disfluency filtered	WER (all)	disfluency filtered
Google-API-stable/quick	25.46	28.16 (+2.70)	40.62	41.60 (+0.98)
Google-API-sticky	26.08	29.29 (+3.21)	41.23	42.82 (+1.59)
Sphinx-4	57.61	62.31 (+4.70)	72.08	75.34 (+3.26)
Kaldi	71.31	73.38 (+2.07)	77.57	79.05 (+1.48)

Table 1 Word Error Rate (WER) results on English Pentomino data for the off-the-shelf systems under different transcript conditions with the WER disfluency gain in brackets.

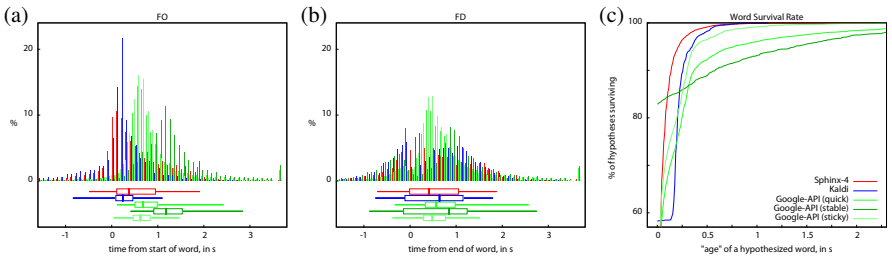

Fig. 3 (a and b): Histograms showing the distribution of *first occurrence* of words (a) and *final decision* for words (b) for the three recognizers (and Google's three settings). Box plots show the median, quartiles (box) and 5/95% quantiles (whiskers). Some extreme (negative) values may be caused by alignment errors. (c): Stability of hypotheses expressed as *word survival rate* over time. A higher curve implies a higher stability.

shows Google filtering out many speech repairs and performing badly around them – see (6-a) vs. (6-b). An improved model for filled pauses would also prevent errors like (7-b).

(6) a. **Reference:** and the and his front uh his le- the the the back
b. **Google-API-fast:** and the and the front of theater

(7) a. **Reference:** uh another L shape except it's um symmetrically
b. **Google-API-fast:** another L shape septic sam symmetrically

Also, we notice that performance varies substantially between UK and US speakers, which is a problem for a corpus that contains mixed speakers. Finally, the post-hoc re-scoring that is performed by Google-API in the stable and quick conditions only marginally improves WER over sticking with the strategy used for incremental processing (presumably SER-optimizing Viterbi decoding).

Finally, we note that the Google-API only provides a transcript of words, both Sphinx and Kaldi generate detailed word timings that can be used for analysis by downstream modules.

6.1.3 Incremental quality

Figure 3 plots timing and stability for three recognizers (and Google's three settings). Timing metrics are shown for all hypothesized words (rather than just for words that match the transcript). As can be seen in Figure 3 (a, b), both Kaldi and Sphinx often have a first impression (FO, Subfigure a) of the word right after it is being spoken, while Google is lagging a little. Google and Sphinx are a little quicker in deciding for a word (FD, Subfigure b) than Kaldi, but Google in particular is hurt by words being revised long after they have been hypothesized. This is clearly observable in Figure 3 (c), which shows that a word still has a 5 %-chance of revision even after it has been hypothesized for 1 second (and Google is already slower in hypothesizing words in the first place). This ratio is even worse when limiting hypotheses to just the 'stable' part, but can be radically improved when ignoring the final, non-incremental changes of Google ASR (the 'sticky' setting), albeit at the cost of about 2 % points WER relative. As Figure 3 (c) also shows, Kaldi most likely performs some variation of hypothesis smoothing (Baumann, Atterer, and Schlangen 2009) for 150 ms.

6.2 Experiment 2: Training models on in-domain data

We found rather poor performance (in terms of WER) for the off-the-shelf open-source systems in our interaction-driven domain, presumably because this speaking style does not conform to the material used when training models for open-source systems. In this experiment, we trained models with in-domain data, under the hypothesis that these result in better performance.

6.2.1 Systems and data

We train acoustic and language models for German using 10.7 hours of transcribed interactions (partly human-human, human-system, and human-wizard) from the Pentomino domain described above.[3] Our Kaldi model is based on an adaptation of the Voxforge recipe, while our Sphinx-4 model uses the standard settings of Sphinxtrain. Both used the same data for training.

We evaluate our trained systems (and the Google systems) on 465 utterances (3,818 words) from randomly chosen speakers from the German data explained above (the rest was used for training). Given the human-Wizard interaction domain, compared to the English corpus above, it contains slower, more dictation-like speech with few disfluencies, so we would expect the accuracy results to be better, all things being equal in this domain. However, we find how the large gap to big data driven ASRs such as Google can be closed somewhat with in-domain trained models.

[3] In our effort, we tried reasonably hard to build well-performing models, but we did not strive for best performance, using as much material (whether in-domain or not) as we could get; e.g., blending our LMs with Wikipedia, or the like.

System	German	
	WER (all)	disfluency filtered
Google-API-stable/quick	22.00	21.86 (-0.14)
Google-API-sticky	20.51	20.44 (-0.07)
Sphinx-4	30.28	30.25 (-0.03)
Kaldi	38.95	38.91 (-0.04)

Table 2 Word Error Rate (WER) results on German Pentomino data for the trained systems under different transcript conditions with the WER disfluency gain in brackets.

6.3 Results

WER results across the reference variants are shown in Table 2. Google-API's systems have comparable performance to the English data above, however the post-hoc rescoring actually hurts on this data, with a relative performance hit of 7 %. Sphinx-4 and Kaldi greatly improve through the in-domain training.

The disfluency results in this setting are not as interesting, given the lack of disfluency in the training files, and we take the analysis on the English data above to be indicative of the relative performance of the ASRs.

Incremental metrics are generally unchanged, with a tendency for Sphinx and Kaldi to perform even better which may be related to their better non-incremental performance.

7 Conclusions

We claim that for suitability for incremental, interactive dialogue systems, ASR, in addition to having good utterance-final accuracy, must also exhibit good incremental properties, and offer a broad interface that either keeps or marks up disfluencies, and provides timing information for downstream processing.

In our evaluation, we find that Google-API offers the best non-incremental performance and almost as good incremental performance as Sphinx and Kaldi. However, Google tends to filter out disfluencies, does not provide word timing information, and limits access to 500 calls per API key a day. We also find that Google's post-hoc rescoring does not improve WER while considerably hurting incremental performance. Finally, Sphinx and Kaldi seem to be on par performance-wise, and at least when trained on in-domain data, these perform similarly well to the Google-API.

We have not, in the present paper, factored out the difference between in-domain acoustic models and language models. LMs may already be enough to boost performance for open-source recognizers and are much easier to train. Finally, we want to look into how to incrementally combine recognizers (e. g. Google-API for lowest-possible WERs with Sphinx or Kaldi for timely and time-stamped responses).

References

Aist, Gregory, James Allen, Ellen Campana, Lucian Galescu, Carlos A. Gomez Gallo, Scott Stoness, Mary Swift, and Michael Tanenhaus (2006). "Software architectures for incremental understanding of human speech". In: *Proceedings of Interspeech*, pp. 1922–1925.

Aist, Gregory, James Allen, Ellen Campana, Carlos Gomez Gallo, Scott Stoness, and Mary Swift (2007). "Incremental understanding in human-computer dialogue and experimental evidence for advantages over nonincremental methods". In: *Proceedings of SemDial*, pp. 149–154.

Asri, Layla El, Romain Laroche, Olivier Pietquin, and Hatim Khouzaimi (2014). "NASTIA: Negotiating Appointment Setting Interface". In: *Proceedings of LREC*, pp. 266–271.

Baumann, Timo, Michaela Atterer, and David Schlangen (2009). "Assessing and improving the performance of speech recognition for incremental systems". In: *Proceedings of NAACL-HTL 2009*. ACL, pp. 380–388.

Baumann, Timo, Okko Buß, and David Schlangen (2011). "Evaluation and Optimisation of Incremental Processors". In: *Dialogue & Discourse* 2.1, pp. 113–141.

Baumann, Timo and David Schlangen (2012). "The InproTK 2012 Release". In: *Proceedings of SDCTD*. ACL.

Brennan, S.E. and M.F. Schober (2001). "How Listeners Compensate for Disfluencies in Spontaneous Speech". In: *Journal of Memory and Language* 44.2, pp. 274–296.

Clark, Herbert H. (1996). *Using Language*. Cambridge University Press.

Clark, Herbert H. and Jean E. Fox Tree (2002). "Using uh and um in spontaneous speaking". In: *Cognition* 84.1, pp. 73–111.

Core, Mark G. and Lenhart K. Schubert (1999). "A syntactic framework for speech repairs and other disruptions". In: *Proceedings of ACL*, pp. 413–420.

Edlund, Jens, Joakim Gustafson, Mattias Heldner, and Anna Hjalmarsson (2008). "Towards human-like spoken dialogue systems". In: *Speech Communication* 50.8-9, pp. 630–645.

Fernández, Raquel, Tatjana Lucht, and David Schlangen (2007). "Referring under restricted interactivity conditions". In: *Proceedings of SIGdial*. ACL, pp. 136–139.

Fischer, Kerstin (2006). *What computer talk is and isn't: Human-computer conversation as intercultural communication*. Vol. 17. Linguistics - Computational Linguistics. AQ-Verlag.

Ginzburg, Jonathan, Raquel Fernández, and David Schlangen (2014). "Disfluencies as intra-utterance dialogue moves". In: *Semantics and Pragmatics* 7.9, pp. 1–64.

Ginzburg, Jonathan, Ye Tian, Pascal Amsili, Claire Beyssade, Barbera Hemforth, Yannick Mathieu, Claire Saillard, Julian Hough, Spyros Kousidis, and David Schlangen (2014). "The Disfluency, Exclamation and Laughter in Dialogue (DUEL) Project". In: *Proceedings of SemDial*, pp. 176–178.

Hough, Julian and Matthew Purver (2014). "Strongly Incremental Repair Detection". In: *Proceedings of EMNLP*. ACL, pp. 78–89.

Kennington, Casey and David Schlangen (2015). "Simple Learning and Compositional Application of Perceptually Grounded Word Meanings for Incremental Reference Resolution". In: *Proceedings of ACL*.

Kousidis, Spyros, Casey Kennington, and David Schlangen (2013). "Investigating speaker gaze and pointing behaviour in human-computer interaction with the mint.tools collection". In: *Proceedings of SIGdial*.

Malsburg, Titus von der, Timo Baumann, and David Schlangen (2009). "TELIDA: A Package for Manipulation and Visualisation of Timed Linguistic Data". In: *Proceedings of SigDial*.

McGraw, Ian and Alexander Gruenstein (2012). "Estimating Word-Stability During Incremental Speech Recognition". In: *Proceedings of Interspeech*.

Meteer, M., A. Taylor, R. MacIntyre, and R. Iyer (1995). *Disfluency annotation stylebook for the Switchboard Corpus. Ms.* Tech. rep. Department of Computer and Information Science, University of Pennsylvania.

Morbini, Fabrizio, Kartik Audhkhasi, Kenji Sagae, Ron Artstein, Dogan Can, Panayiotis Georgiou, Shri Narayanan, Anton Leuski, and David Traum (2013). "Which ASR should I choose for my dialogue system?" In: *Proceedings of SigDial*, pp. 394–403.

Paul, Douglas B and Janet M Baker (1992). "The design for the Wall Street Journal-based CSR corpus". In: *Proceedings of the Workshop on Speech and Natural Language*. ACL, pp. 357–362.

Plátek, Ondřej and Filip Jurčíček (2014). "Free on-line speech recogniser based on Kaldi ASR toolkit producing word posterior lattices". In: *Proceedings of SIGdial*. ACL, pp. 108–112.

Povey, Daniel, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely (2011). "The Kaldi Speech Recognition Toolkit". In: *Proceedings of ASRU*.

Schalkwyk, Johan, Doug Beeferman, Françoise Beaufays, Bill Byrne, Ciprian Chelba, Mike Cohen, Maryam Kamvar, and Brian Strope (2010). "Your Word is my Command: Google Search by Voice: A Case Study". In: *Advances in Speech Recognition*. Springer, pp. 61–90.

Schlangen, David and Gabriel Skantze (2011). "A General, Abstract Model of Incremental Dialogue Processing". In: *Dialogue & Discourse* 2.1, pp. 83–111.

Selfridge, Ethan O., Iker Arizmendi, Peter A. Heeman, and Jason D. Williams (2011). "Stability and accuracy in incremental speech recognition". In: *Proceedings of SigDial*. ACL, pp. 110–119.

Shriberg, Elizabeth (1996). "Disfluencies in Switchboard". In: *Proceedings of ICSLP*.

Skantze, Gabriel and Anna Hjalmarsson (2010). "Towards Incremental Speech Generation in Dialogue Systems". In: *Proceedings of SIGdial*.

Skantze, Gabriel and David Schlangen (2009). "Incremental dialogue processing in a micro-domain". In: *Proceedings of EACL*, pp. 745–753.

Walker, Willie, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel (2004). *Sphinx-4: A Flexible Open Source Framework for Speech Recognition*. Tech. rep. SMLI TR2004-0811. Sun Microsystems Inc.