A Comprehensive Evaluation of Incremental Speech Recognition and Diarization for Conversational AI

Angus Addlesee
The Interaction Lab
Heriot-Watt University
Edinburgh
ja204@hw.ac.uk

Yanchao Yu
The Interaction Lab
Heriot-Watt University
Edinburgh
y.yu@hw.ac.uk

Arash Eshghi
The Interaction Lab
Heriot-Watt University
Edinburgh
a.eshghi@hw.ac.uk

Abstract

Automatic Speech Recognition (ASR) systems are increasingly powerful and more accurate, but also more numerous with several options existing currently as a service (e.g. Google, IBM, and Microsoft). Currently the most stringent standards for such systems are set within the context of their use in, and for, Conversational AI technology. These systems are expected to operate *incrementally in real-time*, be *responsive*, *stable*, and *robust* to the pervasive yet peculiar characteristics of conversational speech such as *disfluencies* and *overlaps*. In this paper we evaluate the most popular of such systems with metrics and experiments designed with these standards in mind. We also evaluate the *speaker diarization (SD)* capabilities of the same systems which will be particularly important for dialogue systems designed to handle *multi-party interaction*. We found that Microsoft has the leading incremental ASR system which preserves disfluent materials and IBM has the leading incremental SD system in addition to the ASR that is most robust to speech overlaps. Google strikes a balance between the two but none of these systems are yet suitable to reliably handle natural spontaneous conversations in real-time.

1 Introduction

Automatic Speech Recognition (ASR) is one of the first important milestones of deep learning (Seide et al., 2011) and the standard evaluation metric for ASR systems, the Word Error Rate (WER), has been consistently improving to impressive levels (Saon et al., 2016; Saon et al., 2017; Xiong et al., 2017). But with a few notable exceptions discussed below, the suitability of these state of the art ASR systems for Conversational AI (henceforth SDS for Spoken Dialogue System[1].) is rarely evaluated.

To be truly natural and responsive, SDS architectures have had to be redesigned (Schlangen and Skantze, 2009; Schlangen et al., 2010; Schlangen and Skantze, 2011; Baumann and Schlangen, 2012) to take into account the inherently time-linear and incremental nature of human language processing (see Purver et al. (2009); Healey et al. (2011) & Kempson et al. (2016) among many others). These *incremental* architectures impose additional evaluation criteria and further constraints on the output required of ASR systems:

First, ASR hypotheses should be produced *incrementally*, word by word, with minimal latency. This means that downstream processing (e.g. for Natural Language Understanding (NLU)) can begin before a turn is complete. This has been shown to lead to dialogue systems which feel more natural (Aist et al., 2006; Skantze and Hjalmarsson, 2010) and more satisfying to interact with (Aist et al., 2007). ASR hypotheses should furthermore be *stable* through time (minimising 'jitter' in the output) (Selfridge et al., 2011; Baumann et al., 2011; Baumann et al., 2016).

Second, incremental detection and processing of disfluent speech is crucial for robust Natural Language Understanding (Hough, 2015; Hough and Schlangen, 2015; Shalyminov et al., 2018). And people are not only capable of processing disfluencies, but they exploit them in the understanding that results (Brennan and Schober, 2001). But Baumann et al. (2016) observe that disfluency structure and its constituent tokens such as edit terms (e.g. "sorry", "I mean") are often removed or rewritten by the ASR system.

[1]We do not use CA here to avoid confusion with the common acronym for Conversation Analysis.

Proceedings of the 28th International Conference on Computational Linguistics, pages 3492–3503
Barcelona, Spain (Online), December 8-13, 2020

Furthermore, in multi-party settings involving an SDS, people often interact directly with each other without using the simpler and more fluent language they tend to use when interacting with a computer (Shriberg, 1996; Fischer, 2006). Another crucial challenge in multi-party settings is that conversational overlaps impact the ASRs performance as speech inputs combine. Speaker Diarization (SD), partitioning an input audio stream into speaker-labeled segments, can be combined with ASR (Shafey et al., 2019) but must also work *incrementally* if we are to produce naturally interactive SDSs.

In this paper, we evaluate three of the most well-known ASR services, namely IBM Watson Speech to Text[2] (Saon et al., 2017), Google Speech to Text[3], and Microsoft Azure Speech to Text[4] (Xiong et al., 2017). The incremental performance criteria and metrics we use were first introduced in Selfridge et al. (2011) & McGraw and Gruenstein (2012), used by Baumann et al. (2011), and detailed below in Sec. 2. For this evaluation, we have used the Switchboard corpus of dyadic telephone conversations (Godfrey et al., 1992b) because it is open-domain (with speech overlaps) and contains high quality annotations of disfluency structure (Meteer et al., 1995).

Both IBM Watson and Google's platforms include SD components that we also evaluate on the Switchboard corpus in this paper. As the Switchboard corpus is strictly dyadic however, we additionally evaluate these SD systems on a second corpus, called AVDIAR (Gebru et al., 2017), containing multi-party conversations with up to four speakers.

Results show that Microsoft has the most *responsive*, *stable*, and *accurate* incremental ASR system which also preserves disfluent materials. IBM and Google both filter filled pauses but even though all systems are impacted by speech overlaps, IBM is the *most* robust to them. IBM also has the leading incremental SD system which suggests it's fitting for multi-party settings. Google is a neat balance between the two others but all three of the systems are not yet adequate to reliably handle natural spontaneous conversations in real-time.

To aid experiment reproducibility and future research in this area, we will release our incremental ASR code for audio processing[5] and framework for evaluation[6]. Alongside this framework, we will also release the Switchboard transcripts we have generated as part of the speech disfluencies experiment detailed in Sec. 4.2.

2 Incremental ASR Performance Criteria

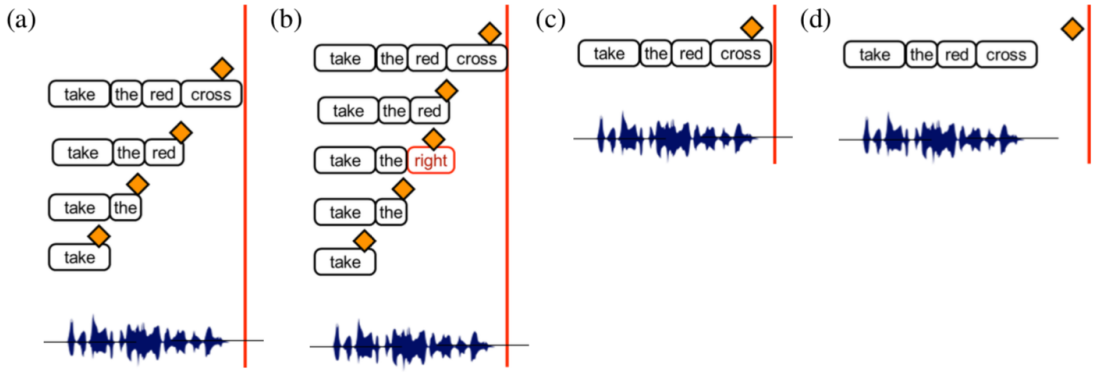

Figure 1: Incremental ASR output: (a) ideal; (b) unstable; (c) not incremental but timely; (d) not incremental and late - from Baumann et al. (2016)

As outlined in the introduction, incremental SDS architectures such as those in Schlangen and Skantze (2011); Eshghi (2015) & Eshghi et al. (2017) require ASR components to be word by word incremental,

[2] https://www.ibm.com/uk-en/cloud/watson-speech-to-text
[3] https://cloud.google.com/speech-to-text
[4] https://azure.microsoft.com/en-gb/services/cognitive-services/speech-to-text/
[5] https://github.com/wallscope-research/incremental-asr-processing
[6] https://github.com/wallscope-research/incremental-asr-evaluation

i.e. that they should produce *word hypotheses* in a time-linear fashion (see Fig. 1). This incrementality in ASR output imposes a set of performance metrics, in addition to the standard turn or utterance-final WER, which Baumann et al. (2016) nicely present. We describe these below.

2.1 Latency

Word hypotheses should be timely with minimal *latency*. There are two kinds of latency:

First Occurance (FO) latency (not to be confused with F0, zero, the fundamental frequency of a speech signal) is the time it takes from the appearance of a word in the gold input signal, to the first time the word hypothesis appears in the output. This is the first hypothesis time *minus* the time that the word *started* to be uttered. As this is timed from the beginning of the word, FO latency cannot be negative.

Final Decision (FD) latency is how long it takes for the system to make a final decision on a word hypothesis such that thereafter the hypothesis does not change in the output. This is the final hypothesis time *minus* the time that the word *finished* being uttered. As this is timed from the end of the word, FD latency can be negative if the final hypothesis is output before the word has finished being spoken.

Referring to example (b) in Fig. 1, the third word was first hypothesised as "right" before the system replaced that hypothesis with "red". The FO latency for the third word in this example is the time that "right" was hypothesised *minus* the *start* of "red" being uttered. The FD latency for the third word however is the time that "red" was hypothesised *minus* the *end* of "red" being uttered. For clarity the FO latency is positive in this case, whereas the FD latency is roughly zero.

2.2 Stability

Stability (Selfridge et al., 2011) is a measure of how stable word hypotheses are, i.e. how likely it is that the ASR system changes its mind or revokes a word later by edit steps (substitution or deletion) in the output. In other words, stability measures how committed the system is to a hypothesis.

For this measure, following Baumann et al. (2016), we report on the **Word Survival Rate (WSR)** of word hypotheses after a certain 'age' (time passed from when the hypothesis was created). At a given time, a higher WSR shows that a system is more stable as a larger percentage of the hypotheses will survive (remain unchanged).

2.3 Fidelity to disfluent material

Conversation is rarely a clean sequence of sentences stringed together (Purver et al., 2009; Hough, 2015; Howes and Eshghi, 2017). It is instead full of pauses, filled pauses (e.g. 'uhm'), stops and starts, restarts, repetitions, and mid-constituent self-corrections, all the while the conversation is going smoothly. As noted in the introduction, capturing these phenomena in an SDS is essential for robust incremental Natural Language Understanding (Hough and Schlangen, 2015; Eshghi et al., 2017; Shalyminov et al., 2018).

Here, in an experiment that is the same in spirit as that in Baumann et al. (2016), we measure: (1) how well ASR systems preserve different types of disfluency token - specifically we look at the preservation of *edit terms* and *filled pauses*; and (2) how well the structure of *self-corrections* (aka self-repair (Schegloff et al., 1977)) is retained in the output, instead of rewritten to a clean, sentential form by the ASR component. We discuss examples of (1) and (2) below in Sec. 3.

We report on these measures as the *gain or loss* in WERs by the different ASR systems in different experimental conditions, created using the disfluency annotations in the Switchboard Corpus (Meteer et al. (1995); see Sec. 3) to automatically rewrite (or 'clean') the corpus into different gold versions[7]. Some with and some without the presence of the annotated disfluency phenomena, such as *edit terms* and *filled pauses*. For *self-corrections*, the same annotations can be used to rewrite the original gold corpus to a version in which self-corrections have been rewritten to a complete constituent form - see (1) below; and see Sec. 4.2 for a more detailed description and subsequent discussion.

[7] https://github.com/wallscope-research/incremental-asr-processing/tree/master/swbcorpustools

Disfluency WER gain Relative to ASR performance against the original gold corpus with all disfluent material intact, WER *gain* in a condition where some disfluent material is removed or rewritten reveals that the system is preserving disfluent material, i.e. that the 'cleaned' ground truth is now less like the system output. A *loss* in WER shows the opposite - the 'cleaned' ground truth is more like the system output and reveals that the system is itself 'cleaning' disfluent material. The magnitude of the loss indicates its accuracy in doing so. In order to utilise natural speech in downstream components (with the aim to tackle challenges that SDSs face in healthcare (Addlesee et al., 2019) for example) we promote the preservation of disfluent material and consider *gains* in WER a positive. For other use cases, such as live captioning in video calls, cleaning disfluent material may be of value.

2.4 Robustness to speech overlaps

In natural everyday conversations, especially those in multi-party settings, speakers often interrupt each other which leads to frequent speech overlaps.

Overlap WER gain To measure how robust an ASR system is to speech overlaps, we measure the difference between the system's overall WER on a single, combined audio stream consisting of speech from all speakers on the one hand; and the WER of the same system on individual channels each carrying speech from a single speaker. A smaller *Overlap WER gain* would suggest that the system is more robust to overlaps.

For reasons which we discuss in Sec. 3, we have in this instance used *Maximum Overlap WER gain* for Switchboard. We define this modified metric in Sec. 4.3.

3 The Switchboard Corpus (SWB)

The Switchboard Corpus (SWB; originally: Godfrey et al. (1992a)) is a large corpus of human-human dyadic conversations in English over the phone on a wide range of topics. The corpus can therefore be described as open-domain. For evaluation of ASR systems, SWB has become the de facto standard through its NIST 2000 evaluation test set/subcorpus[8] - see Saon et al. (2016) & Xiong et al. (2017) among many others.

Here we use Switchboard-1[9], together with its more recent dialogue act and disfluency annotations (Meteer et al., 1995)[10] to evaluate both ASR and SD systems' *incremental performance* for use in Spoken Dialogue Systems (SDSs). We choose SWB for two main reasons: (1) more recent versions of the corpus contain accurate and fine-grained annotations of disfluency material (Meteer et al., 1995) enabling our disfluency fidelity experiments below in Sec. 4.2; and (2) existence of separate channels for each speaker allowing us to study the robustness of ASR systems to speech overlaps.

The disfluency annotations in SWB contain tags for our phenomena of interest in this paper: *edit terms* $\langle e/\rangle$, *filled pauses* $\langle f/\rangle$, as well as the detailed structure of self-corrections. (1) shows an example of how self-corrections are annotated in SWB:

(1) $\underbrace{\text{[with}}_{reparandum} + \underbrace{\text{uh}}_{\langle f/\rangle} \underbrace{\text{I mean}}_{\langle e/\rangle} \underbrace{\text{without]}}_{repair}$ any strings attached

Here, 'uh' is a filled pause, 'I mean' is an edit term; 'without' is the *repair* signalling substitution of 'with', which is the *reparandum*. In self-corrections in general, the reparandum can be deleted, substituted by the repairing phrase, or simply repeated as in (2):

(2) $\underbrace{\text{[it's a}}_{reparandum} + \underbrace{\text{it's a]}}_{repair}$ fairly large community

As noted above in Sec. 2.3, these annotations allow us to create controlled experimental gold test sets where edit terms or filled paused are filtered out; or where self-corrections are rewritten to their 'clean', complete constituent form; e.g. (1) would be rewritten simply to "without any strings attached".

[8] https://mig.nist.gov/MIG_Website/tests/ctr/2000/h5_2000_v1.3.html
[9] https://catalog.ldc.upenn.edu/LDC97S62
[10] The Switchboard Dialogue Act corpus: http://compprag.christopherpotts.net/swda.html

Switchboard's audio files are stereo with each channel representing one of the two speakers. For our system comparison in Sec. 4.1, we report overall incremental WER on the audio with the channels merged together. This more closely resembles the common dyadic dialogue setting as the speech inputs are not separate in a dyadic conversation or a multi-party setting in which one interactant is an SDS.

In order to run the experiments in Sec. 4.2 and 4.3, we separated Switchboards stereo files into their individual channels that each represent one speaker and therefore do not contain speech overlaps. These channels do have audio channel bleeding however so both speakers can sometimes be heard in a single channel. The isolated WERs are increased due to a mismatch between the single speaker gold transcript and the system's output containing both speakers. This is crucial to note but does not impact the findings of our experiments as explained within each experiment description.

4 ASR System Evaluations

For this paper we experimented with three of the most well-known, production-level ASR systems with the ability to transcribe audio incrementally. These were mentioned in Sec. 1 and from now on we will refer to them by their company names - namely: Microsoft, IBM, and Google. We used the most up to date systems as of **May 2020**. The underlying system architectures are undisclosed but, if access was granted, it would be useful to tie them to our metric findings.

Microsoft Azure Speech to Text: Microsoft supports continuous speech recognition which, as the name suggests, is their incremental ASR. It processes an audio stream continuously and can output its current hypothesis upon request. We requested hypotheses every 0.05s and there were often differences in hypotheses at that frequency.

IBM Watson Speech to Text: IBM has a HTTP interface but using their websocket interface is required in order to receive interim results. IBM notes that this feature returns intermediate hypotheses that are for interactive applications and real-time transcription. These intermediate results are returned by the system as soon as they are generated and they are notably less frequent than the other systems, usually returning increments spanning more than one word. This is an important characteristic of the IBM system in explaining our latency and stability results to which we return below.

Google Speech to Text: Google returns incremental speech recognition requests via their 'streaming_recognize' cloud API method when interim results are requested in the config. As Google provides an option of four standard ASR models, we selected and used the 'phone_call' model as the Switchboard corpus is a collection of dyadic conversations over the phone. Similarly to IBM, interim results are returned as they are generated. Unlike IBM however, these are returned much more frequently.

4.1 Overall Incremental ASR Performance

For each of the production-level incremental ASR systems detailed in Sec. 4, we transcribed the Switchboard corpus (Sec. 3) using the performance criteria described in Sec. 2.

We first present each systems incremental WER in table 1 and as explained in Sec. 3, the two audio channels were merged to imitate the setting of an SDS. The overall WER is the *mean incremental* WER across the Switchboard corpus per system. We then present each system's FO latency, FD latency, and stability in Fig. 2.

Service	**Microsoft**	**IBM**	**Google**
Incremental WER (%)	**32.89**	35.55	33.62
Non-Incremental WER (%)	**5.1**	5.5	6.8

Table 1: Overall WER of Incremental ASR Systems on Switchboard

Overall Incremental WER: The overall system WERs are all roughly the same (range of 2.66%) but Microsoft does perform better than the others with Google following in close second.

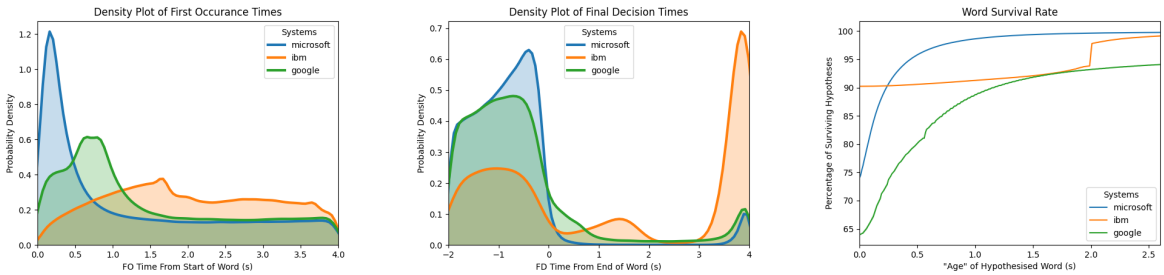

Figure 2: From left to right we have charts (a), (b), and (c). Charts (a) and (b) show the distribution of (a) the first occurrence of words and (b) the final decision of words. (c) shows the survival rate of a word hypothesis, i.e. what percentage of hypotheses are stable after a duration of time.

Microsoft, IBM and Google have all reported significantly lower non-incremental WERs on the Switchboard corpus than we report here. Microsoft has achieved a WER of 5.1% (Xiong et al., 2018), IBM has achieved a WER of 5.5% (Saon et al., 2017), and Google has achieved a WER of 6.8% (Johnson, 2019). The results are incredibly impressive but these bidirectional models are not designed to be used in real time[11] - there is no way for a system to use future words to predict the current one. This comparative lack of information is likely the main reason that the *incremental* WERs we report are much larger. This difference between incremental and non-incremental performance is not limited to ASR. For example, incremental disfluency tagging is also a more challenging task than its non-incremental counterpart for the same reason that the model is unable to exploit future tokens (Shalyminov et al., 2018).

If we inspect charts (a), (b) and (c) in Fig. 2, we start to see some major system differences.

FO Latency: Firstly in chart (a), 0s on the x-axis represents the start of the word being uttered and the first occurrence (FO) time is the time that the word was first hypothesised.

We can see that Microsoft almost always hypothesises a word a mere fraction of a second after it begins to be uttered. This is extremely responsive and much faster than the other two systems. Google usually hypothesises a word less than a second after it begins to be spoken (which is of course still adequately quick) and is often as responsive as Microsoft. IBM however has a more even spread of FO latency times, often hypothesising a word over 1.5s after it begins to be spoken. It does sometimes hypothesise words sooner but generally IBM is not very responsive.

As mentioned in Sec. 4, Micorosft allows you to request the current hypothesis as often as desired. Google and IBM however only send you hypotheses as they are generated. IBM sends hypotheses significantly less frequently which is illustrated in this result.

FD Latency: A low FO latency is desirable for an incremental and responsive system but if the system continues to change its hypothesis, downstream components would be constantly changing their selected actions too. We want a system to update their hypothesis as a word unfolds but how long after a word has finished being uttered does each system finally decide on that word? In chart (b) we begin to inspect this.

All the systems frequently decide on a final hypothesis before the word has even finished being spoken. Both Microsoft and Google are both very responsive, chart (a), *and* usually decide on a final word hypothesis before it has finished being spoken. This is ideal behaviour for an incremental ASR system. IBM on the other hand is less responsive and frequently changes its hypothesis *seconds* after it has finished being spoken. This is again a result of its less-frequent hypothesis outputs as it doesn't output them word per word. If the system changes its hypothesis on a single word, it is only output as part of a much larger hypothesis - seconds later.

Stability: Finally, we investigated how stable a system's hypothesis is. As previously mentioned, incremental ASR systems can change their hypotheses on the fly and it is inconvenient for downstream components if this happens over a long duration of time. Chart (c) displays the percentage of hypotheses

[11] In response to a anonymous reviewer, bidirectional models could potentially be used incrementally if the model is applied to progressively longer segments of the signal as it appears. However, the system would still have significantly less information than a non-incremental system, so the improvement in WER would likely not make up for the increase in latency and instability.

that are finalised at a given 'age' (FD hypothesis time minus FO hypothesis time). It is important to note that the y-axis does not start at 0 and the point at which the line cuts this axis is the percentage of hypotheses that do not ever change.

Beginning with Microsoft, we can see that roughly three quarters of the system's predictions do not ever change. Hypotheses stabilise fast however with about 95% of them becoming stable after just half a second. From one second onwards, almost all of Microsoft's predictions are stable.

IBM on the other hand rarely changes its hypotheses with over 90% of its predictions remaining unchanged. As illustrated in chart (a), IBM takes longer to output its hypotheses and this is likely a design choice to achieve this high initial stability. As with the FD latencies in chart (b), the step in survival rate is again due to IBM's less-frequent prediction rate. This focus on initial stability doesn't just impact its latencies, Microsoft with its lower initial stability has more stable hypotheses after just a fraction of a second.

Google changes the largest percentage of its hypotheses but nevertheless, almost 65% of its initial predictions remained unchanged. Even with more frequent prediction rates than IBM, Google has a smaller percentage of surviving hypotheses at any 'age'. After 2.5s, over 5% of Google's hypotheses were still not finalised.

As just touched upon, the step in IBMs word survival rate depicts the system's slow hypothesis rate. Conversely, even though the *number* of hypotheses is not explicitly represented in this chart, Microsoft's extremely smooth word survival rate confirms that it does indeed change its hypothesis more frequently than Google. It is necessary to point out that the *number* of word hypotheses does not indicate that a system is unstable, the duration of this instability is the concern.

4.2 Experiment 1: Fidelity to Disfluencies and Self-corrections

As discussed in Sec. 2.3, to determine whether a system is preserving disfluent material (or not) we will report the gain (or loss) in WERs. An increase in WER reveals that the 'cleaned' ground truth is now less like the system output, therefore the system must be preserving disfluent materials. Inversely, a decrease in WER shows that a system is also 'cleaning' its output.

Condition	Microsoft	IBM	Google
Orig	67.48	73.42	68.17
SC	72.32 (+4.84)	78.44 (+5.02)	71.69 (+3.52)
ET	67.79 (+0.31)	73.77 (+0.35)	68.40 (+0.23)
FP	69.65 (+2.17)	73.58 (+0.16)	67.96 (-0.21)

Table 2: Disfluency WERs on Split Switchboard Dataset. We report results over three conditions: SC - rewritten/cleaned self-corrections; ET - filtered edit terms; FP - filtered filled pauses

Explained above in Sec. 3, we run this experiment on the Switchboard corpus but with the speakers isolated into their individual channels. Audio channel bleeding increases the original WER from the WERs reported in Sec. 4.1 but this does not impact this experiment. The system outputs are the same throughout so loss or gain in WER can only come from the 'cleaning' of the gold transcripts.

Reading table 2, all of the systems have a similar increase in WER after self-corrections are rewritten (row SC). Rewriting self corrections would be a sophisticated operation to perform incrementally so it is unsurprising to see that no systems appear to do this.

Similarly, all of the systems have a slight increase in WER when edit terms are filtered (row ET). As the increase is small, it is possible that the systems are all filtering these but not exceptionally well. As the three systems perform so similarly however, it is more likely that none of the systems do this at all.

Finally, after filtering filled pauses (row FP), Microsoft has an increase in WER of over 2%. IBM however has a tiny increase in WER and Google even manages to decrease its WER on the filtered ground truth. This indicates that Microsoft preserves filled pauses whereas IBM and Google both filter these on the fly. This behaviour by Google and IBM is presumably by design, as discussed in Sec. 2.3, but we promote the preservation of disfluent materials to aid downstream components.

4.3 Experiment 2: Robustness to Speech Overlaps

In this experiment, we wish to measure and compare the ASR systems' robustness to speech overlaps. In general, the most natural way of doing this is to compare system WERs on a single combined channel containing speech from both parties to WERs on the mean of the split individual channels each containing speech **only** from one party. However, as detailed in Sec. 3, there is often a lot of audio bleeding from one channel to the other in Switchboard. Our initial investigation showed that the systems were in fact performing *worse* for some of the dialogues on the individual channels compared to the combined channel. Further investigation of these individual dialogues showed that they indeed did contain significant bleeding.

Thankfully, audio channel bleeding is not present in every case, so to tackle this issue we will report the *maximum* WER improvement (MWERI) for each system *across all dialogues*. That is, the difference between the merged audio WER and the mean WER of both speakers on the split channels *on a single dialogue*. For complete clarity, the dialogues leading to maximum improvement can be different for different ASR systems but for a single system the WER improvement is calculated *per dialogue*. Max WER Improvement is defined in Eq. (1):

$$\text{MWERI} = max(WER^{combined} - \frac{WER^{speakerA} + WER^{speakerB}}{2}) \quad (1)$$

Inspecting Table 3 it is evident that IBM is the most robust to overlaps with the smallest MWERI. That is, when overlaps were removed from the audio input, the improvement was the smallest and therefore the system was initially accurate with the overlaps present. Microsoft and Google perform similarly on this experiment but are considerably less robust to speech overlaps than IBM.

Service	**Microsoft**	**IBM**	**Google**
Max WER Improvement (%)	19.82	**14.76**	20.09

Table 3: Robustness to Speech Overlaps

Handling speech overlaps is necessary in multi-party settings in which one interactant is an SDS. These scenarios are not uncommon now as more and more people have SDSs in their home.

5 Speaker Diarization (SD) System Evaluations

Here we present our evaluation of the Google and IBM SD systems - Microsoft does not currently include a SD service. We evaluate both systems within a *real-time (i.e. incremental) multi-party setting*. For both systems, we therefore only take into account the interim results via a streaming method instead of the final results, although we acknowledge that the final result usually performs better than the interim ones.

Google Cloud SD: In the past three years or so, Google have presented several SD models. Of these, Zhang et al. (2019) achieves the lowest Diarization Error Rate (DER) of 7.6% on the NIST Speaker Recognition Evaluation benchmark (LDC2001S97) compared to their previous models (Wang et al., 2018; Garcia-Romero et al., 2017; Zhang et al., 2018)[12]. For our experiments below, we cannot ascertain which model is currently deployed on the cloud service. Of the different models within the Google Cloud Services to support different types of audio stream, we evaluate the 'video' model only below.

IBM Watson Speaker Diariztion: Dimitriadis and Fousek (2017) present the first online, real-time, production level SD system within the IBM cloud Speech-To-Text (STT). The deployment proceeds in two stages: (1) an unsupervised audio segmentation to the homogeneous segments per speaker; and (2) speaker labelling. Since extracting acoustic representations from audio within non-speech sessions may affect SD performance, Dimitriadis and Fousek (2017) move the ASR module forward in their architecture, where ASR results are used as input to the segmenter to separate different speaker chunks, and to improve the overall SD performance without dropping the ASR quality. It is this SD system which we evaluate below, noting that it operates together with ASR.

[12] In a recent Google paper, Shafey et al. (2019) introduce a joint ASR+SD model with acoustic and linguistic cues that learns to jointly transcribe speech and predict speaker labels; but they compare their performance only to their own baseline

5.1 Experiment 3: Speaker Diarization

Data Unlike ASR, SD systems are affected by the number of speakers involved in a multi-party conversation in a real-time, streaming setting as we do here. Therefore, in addition to the Switchboard data (see Sec.3), we also use AVDIAR (Gebru et al., 2017): a multi-party audio-visual diarization data set consisting of 23, up-to-four speaker conversations in both audio and video form; though in this paper we only use the audio data.

Metric We use the Diarization Error Rate (DER) as a performance metric for this experiment. DER was first introduced by NIST in their Rich Transcription (RT) benchmarks (Fiscus et al., 2006), which consists of three factors: (a) Speaker Error rate (SER) measuring the percentage of time with incorrectly predicted speaker labels; (b) False Alarm Speech rate (FASR) indicating the proportion of time during which non-speech regions are incorrectly identified as containing speech; and (c) the Missed Speech rate (MSR) (the opposite of FASR). We, here, employ the d-score toolkit[13] for with a forgiveness collar of 0.25 seconds (Ortega et al., 2016).

Switchboard Results Table 4 shows the performance of Google and IBM SD services on the Switchboard data set: with 15.33% DER, IBM Watson performs considerably better than Google with a 43.93% DER; in large part due to incorrect speaker labels (SER) from Google.

Service	MSR (%)	FASR (%)	SER (%)	DER (%)
Google	17.35	**0.39**	26.15	43.93
IBM	**12.96**	0.73	**1.64**	**15.33**

Table 4: Performance of SD Systems on Switchboard

AVDIAR Results Since the data set contains conversations involving up to four speakers, we compare performance of the two services across conversations with different number of speakers (see Table 5). In this table, we only present Google's score from 18 out of 24 audio streams, since Google can't recognise any speakers from the rest of the audio files (including all 4-person conversation streams). In general, both Google (68.79% DER) and IBM (48.94% DER) show considerably worse performance on AVDIAR than than on Switchboard, even when we only consider the dyadic conversations (Google: 66.43% DER; IBM: 41.89% DER). Performance on dyadic conversations is also not considerably better than the overall performance. This might be because the audio in AVDIAR is of a lower quality/amplitude because speakers stand further from the microphone. And it is likely this lower audio quality that leads Google to misidentify speech session very often (57.99% MSR overall), while at the same time showing a more stable performance on speaker labelling (10.07% SER overall). In contrast, IBM Watson on AVDIAR shows much higher Speaker Error rate overall (21.24%) than in Switchboard.

Num of Speakers	Service	MSR (%)	FASR (%)	SER (%)	DER (%)
1	Google	49.62	0.00	6.45	56.06
	IBM	25.47	0.51	13.14	39.12
2	Google	57.48	0.98	7.97	66.43
	IBM	23.95	2.05	15.88	41.89
3	Google	60.74	0.57	13.91	75.22
	IBM	28.14	1.07	20.41	49.62
4	Google	-	-	-	-
	IBM	29.65	0.85	36.82	67.32
Overall	Google	57.99	**0.73**	**10.07**	68.79
	IBM	**26.31**	1.39	21.24	**48.94**

Table 5: Performance of Speaker Diarization Systems on AVDIAR (Gebru et al., 2017)

Overall, IBM shows considerably better performance than Google across both data sets. But we note that the AVDIAR setting is a lot more similar than Switchboard to a real-life human-robot interaction

[13] https://github.com/nryant/dscore

scenario, in which we expect audio quality and intensity to vary as speakers move away or closer to the robot, and as ambient noise changes. Overall it is evident that neither Google nor IBM SD systems are ready to cope with natural, spontaneous conversation with real humans, especially in multi-party settings.

6 Conclusion

Looking at only each system's incremental WERs, which are all very similar, it is impossible to distinguish which system is best for a certain use-case. Using the additional metrics and further experiments detailed in this paper however, we have carried out a more sophisticated analysis.

Microsoft has the most *responsive*, *stable* and *accurate* incremental ASR system, while preserving disfluent material but does not have a real-time SD system. IBM and Google both filter filled pauses but even though all systems are impacted by speech overlaps, IBM is the *most* robust to them. IBM also has the leading real-time SD system which suggests it's a better fit for multi-party settings. Google is a neat balance between the two others. All of the systems report impressive WERs on the Switchboard corpus using non-incremental (bidirectional) models but this does not mean that this challenge is complete. None of these systems are yet adequate to reliably handle natural spontaneous conversations *in real-time*.

We hope that by releasing our evaluation framework, we will encourage future assessment of such systems and promote the design of speech recognition and diarization systems with the above metrics in mind.

7 Acknowledgements

The first author is funded by Wallscope [14] & The Data Lab[15]. The second author is funded by the Horizon2020 SPRING project[16] (Grant Agreement Number: #871245). We thank them for their support.

References

Angus Addlesee, Arash Eshghi, and Ioannis Konstas. 2019. Current challenges in spoken dialogue systems and why they are critical for those living with dementia. In *Proceedings of the Dialogue for Good (DiGo) workshop on Speech and Language Technology Serving Society*.

G.S. Aist, J. Allen, E. Campana, L. Galescu, C.A. Gomez Gallo, S. Stoness, M. Swift, and M. Tanenhaus. 2006. Software architectures for incremental understanding of human speech. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Pittsburgh.

G. Aist, J. Allen, E. Campana, C.A. Gomez Gallo, S. Stoness, M. Swift, and M.K. Tanenhaus. 2007. Incremental dialogue system faster than and preferred to its nonincremental counterpart. In *Proceedings of the 29th Annual Conference of the Cognitive Science Society*.

Timo Baumann and David Schlangen. 2012. The inprotk 2012 release. In *NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data*, pages 29–32. Association for Computational Linguistics.

T. Baumann, O. Buß, and D. Schlangen. 2011. Evaluation and optimisation of incremental processors. *Dialogue & Discourse*, 2(1):113–141.

Timo Baumann, Casey Kennington, Julian Hough, and David Schlangen. 2016. Recognising conversational speech: What an incremental asr should do for a dialogue system and how to get there. In *International Workshop on Dialogue Systems Technology (IWSDS) 2016*. Universität Hamburg.

S.E. Brennan and M.F. Schober. 2001. How listeners compensate for disfluencies in spontaneous speech. *Journal of Memory and Language*, 44(2):274–296.

Dimitrios Dimitriadis and Petr Fousek. 2017. Developing on-line speaker diarization system. In *INTERSPEECH*.

Arash Eshghi, Igor Shalyminov, and Oliver Lemon. 2017. Bootstrapping incremental dialogue systems from minimal data: the generalisation power of dialogue grammars. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[14] https://wallscope.co.uk/
[15] https://www.thedatalab.com/
[16] https://spring-h2020.eu/

Arash Eshghi. 2015. DS-TTR: An incremental, semantic, contextual parser for dialogue. In *Proceedings of Semdial 2015 (goDial), the 19th workshop on the semantics and pragmatics of dialogue*.

Kerstin Fischer. 2006. What computer talk is and isn't: Human-computer con- versation as intercultural communication. *Linguistics - Computational Linguistics*, 17:53–66.

Jonathan G. Fiscus, Jerome Ajot, Martial Michel, and John S. Garofolo. 2006. The rich transcription 2006 spring meeting recognition evaluation. In Steve Renals, Samy Bengio, and Jonathan G. Fiscus, editors, *Machine Learning for Multimodal Interaction, Third International Workshop, MLMI 2006, Bethesda, MD, USA, May 1-4, 2006, Revised Selected Papers*, volume 4299 of *Lecture Notes in Computer Science*, pages 309–322. Springer.

Daniel Garcia-Romero, David Snyder, Gregory Sell, Daniel Povey, and Alan McCree. 2017. Speaker diarization using deep neural network embeddings. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, pages 4930–4934. IEEE.

Israel D. Gebru, Silèye Ba, Xiaofei Li, and Radu Horaud. 2017. Audio-visual speaker diarization based on spatiotemporal bayesian fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39.

J. J. Godfrey, E. C. Holliman, and J. McDaniel. 1992a. Switchboard: Telephone speech corpus for research and development. In *Proceedings of ICASSP*, pages 517–520, Paris, France, September.

John J. Godfrey, Edward Holliman, and J. McDaniel. 1992b. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of IEEE ICASSP-92*, pages 517–520, San Francisco, CA.

P. G. T. Healey, Arash Eshghi, Christine Howes, and Matthew Purver. 2011. Making a contribution: Processing clarification requests in dialogue. In *Proceedings of the 21st Annual Meeting of the Society for Text and Discourse*, Poitiers, July.

Julian Hough and David Schlangen. 2015. Recurrent neural networks for incremental disfluency detection. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, pages 849–853.

Julian Hough. 2015. *Modelling Incremental Self-Repair Processing in Dialogue*. Ph.D. thesis, Queen Mary University of London.

Christine Howes and Arash Eshghi. 2017. Feedback relevance spaces: The organisation of increments in conversation. In *IWCS 2017 — 12th International Conference on Computational Semantics*.

Khari Johnson. 2019. Google's specaugment achieves state-of-the-art speech recognition without a language model. *VentureBeat*.

Ruth Kempson, Ronnie Cann, Eleni Gregoromichelaki, and Stergios Chatzikiriakidis. 2016. Language as mechanisms for interaction. *Theoretical Linguistics*, 42(3-4):203–275.

Ian McGraw and Alexander Gruenstein. 2012. Estimating word-stability during incremental speech recognition. In *Proceedings of INTERSPEECH'12*.

M. Meteer, A. Taylor, R. MacIntyre, and R. Iyer. 1995. Disfluency annotation stylebook for the switchboard corpus. ms. Technical report, Department of Computer and Information Science, University of Pennsylvania.

Alfonso Ortega, Ignacio Vinals, Antonio Miguel, and Eduardo Lleida. 2016. The albayzin 2016 speaker diarization evaluation. In *Proc. IberSPEECH 2016, Lisbon, Portugal*.

Matthew Purver, Christine Howes, Eleni Gregoromichelaki, and Patrick G. T. Healey. 2009. Split utterances in dialogue: A corpus study. In *Proceedings of the 10th Annual SIGDIAL Meeting on Discourse and Dialogue (SIGDIAL 2009 Conference)*, pages 262–271, London, UK, September. Association for Computational Linguistics.

George Saon, Tom Sercu, Steven J. Rennie, and Hong-Kwang Jeff Kuo. 2016. The ibm 2016 english conversational telephone speech recognition system. *ArXiv*, abs/1604.08242.

George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, Bergul Roomi, and Phil Hall. 2017. English conversational telephone speech recognition by humans and machines. In *Proc. Interspeech 2017*, pages 132–136.

E.A. Schegloff, Gail Jefferson, and Harvey Sacks. 1977. The preference for self-correction in the organization of repair in conversation. *Language*, 53(2):361–382.

David Schlangen and Gabriel Skantze. 2009. A general, abstract model of incremental dialogue processing. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 710–718, Athens, Greece, March. Association for Computational Linguistics.

David Schlangen and Gabriel Skantze. 2011. A general, abstract model of incremental dialogue processing. *Dialogue and Discourse*, 2(1):83–111.

David Schlangen, Timo Baumann, Hendrik Buschmeier, Okko Buß, Stefan Kopp, Gabriel Skantze, and Ramin Yaghoubzadeh. 2010. Middleware for incremental processing in conversational agents. In *Proceedings of the SIGDIAL 2010 Conference*, pages 51–54, Tokyo, Japan, September. Association for Computational Linguistics.

F. Seide, G. Li, X. Chen, and D. Yu. 2011. Feature engineering in context-dependent deep neural networks for conversational speech transcription. In *2011 IEEE Workshop on Automatic Speech Recognition Understanding*, pages 24–29.

Ethan Selfridge, Iker Arizmendi, Peter Heeman, and Jason Williams. 2011. Stability and accuracy in incremental speech recognition. In *Proceedings of the SIGDIAL 2011 Conference*, pages 110–119, Portland, Oregon, June. Association for Computational Linguistics.

Laurent El Shafey, Hagen Soltau, and Izhak Shafran. 2019. Joint Speech Recognition and Speaker Diarization via Sequence Transduction. In *Proc. Interspeech 2019*, pages 396–400.

Igor Shalyminov, Arash Eshghi, and Oliver Lemon. 2018. Multi-task learning for domain-general spoken disfluency detection in dialogue systems. In *Proceedings of the 22nd Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, Aix-en-Provence, France, November. SEMDIAL.

Elizabeth Shriberg. 1996. Disfluencies in switchboard. In *In Proceedings of the International Conference on Spoken Language Processing*, volume 96, pages 3–6. Citeseer.

Gabriel Skantze and Anna Hjalmarsson. 2010. Towards incremental speech generation in dialogue systems. In *Proceedings of the SIGDIAL 2010 Conference*, pages 1–8, Tokyo, Japan, September. Association for Computational Linguistics.

Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopez-Moreno. 2018. Speaker diarization with LSTM. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 5239–5243. IEEE.

W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig. 2017. The microsoft 2016 conversational speech recognition system. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5255–5259.

Wayne Xiong, Lingfeng Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke. 2018. The microsoft 2017 conversational speech recognition system. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5934–5938. IEEE.

Aonan Zhang, Quan Wang, Zhenyao Zhu, John W. Paisley, and Chong Wang. 2018. Fully supervised speaker diarization. *CoRR*, abs/1810.04719.

Aonan Zhang, Quan Wang, Zhenyao Zhu, John W. Paisley, and Chong Wang. 2019. Fully supervised speaker diarization. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 6301–6305. IEEE.