Toward incremental dialogue act segmentation in fast-paced interactive dialogue systems

Ramesh Manuvinakurike[1], Maike Paetzel[2], Cheng Qu[1], David Schlangen[3] and David DeVault[1]
[1]USC Institute for Creative Technologies, Playa Vista, CA, USA
[2]Uppsala University, Department of Information Technology, Uppsala, Sweden
[3]Bielefeld University, Bielefeld, Germany

Abstract

In this paper, we present and evaluate an approach to incremental dialogue act (DA) segmentation and classification. Our approach utilizes prosodic, lexico-syntactic and contextual features, and achieves an encouraging level of performance in offline corpus-based evaluation as well as in simulated human-agent dialogues. Our approach uses a pipeline of sequential processing steps, and we investigate the contribution of different processing steps to DA segmentation errors. We present our results using both existing and new metrics for DA segmentation. The incremental DA segmentation capability described here may help future systems to allow more natural speech from users and enable more natural patterns of interaction.

1 Introduction

In this paper we explore the feasibility of incorporating an incremental dialogue act segmentation capability into an implemented, high-performance spoken dialogue agent that plays a time-constrained image-matching game with its users (Paetzel et al., 2015). This work is part of a longer-term research program that aims to use incremental (word-by-word) language processing techniques to enable dialogue agents to support efficient, fast-paced interactions with a natural conversational style (DeVault et al., 2011; Ward and DeVault, 2015; Paetzel et al., 2015).

It's important to allow users to speak naturally to spoken dialogue systems. It has been understood for some time that this ultimately requires a system to be able to automatically segment a user's speech into meaningful units in real-time while they speak (Nakano et al., 1999). Still, most current systems use relatively simple and limited approaches to this segmentation problem. For example, in many systems, it's assumed that pauses in the user's speech can be used to determine the segmentation, often by treating each detected pause as indicating a dialogue act (DA) boundary (Komatani et al., 2015).

While easily implemented, such a pause-based design has several problems. First, a substantial number of spoken DAs contain internal pauses (Bell et al., 2001; Komatani et al., 2015), as in *I need a car in... 10 minutes*. Using simple pause length thresholds to join certain speech segments together for interpretation is not a very effective remedy for this problem (Nakano et al., 1999; Ferrer et al., 2003). More sophisticated approaches train algorithms to join speech across pauses (Komatani et al., 2015) or decide which pauses constitute end-of-utterances that should trigger interpretation (e.g. (Raux and Eskenazi, 2008; Ferrer et al., 2003)). This addresses the problem of DA-internal pauses, but it does not address the second problem with pause-based designs, which is that it's also common for a continuous segment of user speech to include multiple DAs *without* intervening pauses, as in *Sure that's fine can you call when you get to the gate?* A third problem is that waiting for a pause to occur before interpreting earlier speech may increase latency and erode the user experience (Skantze and Schlangen, 2009; Paetzel et al., 2015). Together, these problems suggest the need for an incremental dialogue act segmentation capability in which a continuous stream of captured user speech, including the intermittent pauses therein, is incrementally segmented into appropriate DA units for interpretation.

In this paper, we present a case study of implementing an incremental DA segmentation capability for an image-matching game called RDG-Image, illustrated in Figure 1. In this game, two players converse freely in order to identify a spe-

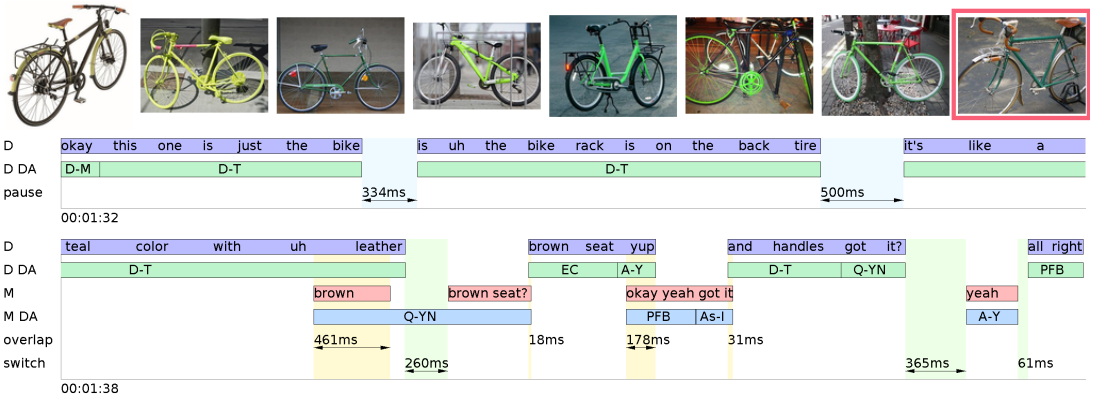

Figure 1: An example RDG-Image dialogue, where the director (D) tries to identify the target image, highlighted in red, to the matcher (M). The DAs of the director (D DA) and matcher (M DA) are indicated.

cific target image on the screen (outlined in red). When played by human players, as in Figure 1, the game creates a variety of fast-paced interaction patterns, such as question-answer exchanges. Our motivation is to eventually enable a future version of our automated RDG-Image agent (Paetzel et al., 2015) to participate in the most common interaction patterns in human-human gameplay. For example, in Figure 1, two fast-paced question-answer exchanges arise as the director D is describing the target image. In the first, the matcher M asks *brown...brown seat?* and receives an almost immediate answer *brown seat yup*. A moment later, the director continues the description with *and handles got it?*, both adding *and handles* and also asking *got it?* without an intervening pause. We believe that an important step toward automating such fast-paced exchanges is to create an ability for an automated agent to incrementally recognize the various DAs, such as yes-no questions (Q-YN), target descriptions (D-T), and yes answers (A-Y) in real-time as they are happening.

The contributions of this paper are as follows. First, we define a sequential approach to incremental DA segmentation and classification that is straightforward to implement and which achieves a useful level of performance when trained on a small annotated corpus of domain-specific DAs. Second, we explore the performance of our approach using both existing and new performance metrics for DA segmentation. Our new metrics emphasize the importance of precision and recall of specific DA types, independently of DA boundaries. These metrics are useful for evaluating DA segmenters that operate on noisy ASR output and which are intended for use in systems whose dialogue policies are defined in terms of the presence or absence of specific DA types, independently of their position in user speech. This is a broad class of systems. Third, while much of the prior work on DA segmentation has been corpus-based, we report here on an initial integration of our incremental DA segmenter into an implemented, high-performance agent for the RDG-Image game. Our case study suggests that incremental DA segmentation can be performed with sufficient accuracy for us to begin to extend our baseline agent's conversational abilities without significantly degrading its current performance.

2 Related Work

In this paper, we are concerned with the alignment between dialogue acts (DAs) and individual words as they are spoken within Inter-Pausal Units (IPUs) (Koiso et al., 1998) or *speech segments*. (We use the two terms interchangeably in this paper to refer to a period of continuous speech separated by pauses of a minimum duration before and after.) Beyond the work on this alignment problem mentioned in the introduction, a related line of work has looked specifically at DA segmentation and classification given an input string of words together with an audio recording to enable prosodic and timing analysis (Petukhova and Bunt, 2014; Zimmermann, 2009; Zimmermann et al., 2006; Lendvai and Geertzen, 2007; Ang et al., 2005; Nakano et al., 1999; Warnke et al., 1997). This work generally encompasses the problems of identifying DA-internal pauses as well as locating DA boundaries within speech segments. Prosody information has been shown to be helpful for accurate DA segmentation (Laskowski and Shriberg, 2010; Shriberg et al.,

2000; Warnke et al., 1997) as well as for DA classification (Stolcke et al., 2000; Fernandez and Picard, 2002). In general, DA segmentation has been found to benefit from a range of additional features such as pause durations at word boundaries, the user's dialogue tempo (Komatani et al., 2015), as well as lexical, syntactic, and semantic features. Work on system turn-taking decisions has used similar features to optimize a system's turn-taking policy during a user pause, often with classification approaches; e.g. (Sato et al., 2002; Takeuchi et al., 2004; Raux and Eskenazi, 2008). To our knowledge, very little research has looked in detail at the impact of adding incremental DA segmentation to an implemented incremental system (though see Nakano et al. (1999)).[1]

3 The RDG-Image Game and Data Set

Our work in this paper is based on the RDG-Image game (Paetzel et al., 2014), a collaborative, time constrained, fast-paced game with two players depicted in Figure 1. One player is assigned the role of director and the other the role of matcher. Both players see the same eight images on their screens (but arranged in a different order). The director's screen has a target image highlighted in red, and the director's goal is to describe the target image so that the matcher can identify it as quickly as possible. Once the matcher believes they have selected the right image, the director can request the next target. Both players score a point for each correct selection, and the game continues until a time limit is reached. The time limit is chosen to create time pressure.

3.1 Dialogue Act Annotations

We have previously collected data sets of human-human gameplay in RDG-Image both in a lab setting (Paetzel et al., 2014) and in an online, web-based version of the game (Manuvinakurike and DeVault, 2015; Paetzel et al., 2015). To support the experiments in this paper, a single annotator segmented and annotated the main game rounds from our lab-based RDG-Image corpus with a set of DA tags.[2] The corpus includes gameplay between 64 participants (32 pairs, age: $M = 35$, $SD = 12$, gender: 55% female). 11% of all participants reported they frequently played similar games before; the other 89% had no or very rare experience with similar games. All speech was previously recorded, manually segmented into speech segments (IPUs) at pauses of 300ms or greater, and manually transcribed. The new DA segmentation and annotation steps were carried out at the same time by adding boundaries and DA labels to the transcribed speech segments from the game. The annotator used both audio and video recordings to assist with the annotation task. The annotations were performed on transcripts which were seen as segmented into IPUs.

Table 1 provides several examples of this annotation. We designed the set of DA labels to include a range of communicative functions we observed in human-human gameplay, and to encode distinctions we expected to prove useful in an automated agent for RDG-Image. Our DA label set includes Positive Feedback (PFB), Describe Target (D-T), Self-Talk (ST), Yes-No Question (Q-YN), Echo Confirmation (EC), Assert Identified (As-I), and Assert Skip (As-S). We also include a filled-pause DA (P) used for 'uh' or 'um' separated from other speech by a pause. The complete list of 18 DA labels and their distribution are included in Tables 9 and 10 in the appendix. To assess the reliability of annotation, two annotators annotated one game (2 players, 372 speech segments); we measured kappa for the presence of boundary markers (||) at 0.92 and word-level kappa for DA labels at 0.83.

Summary statistics for the annotated corpus are as follows. The corpus contains 64 participants (32 pairs), 1,906 target images, 8,792 speech segments, 67,125 word tokens, 12,241 DA segments, and 4.27 hours of audio. The mean number of DAs per speech segment is 1.39. In Table 2, we summarize the distribution in number of DAs initiated per speech segment. 23% of speech segments contain the beginning of at least two DAs; this highlights the importance of being able to find the boundaries between multiple DAs inside a speech segment. Most DAs begin at the start of a speech segment (i.e. immediately after a pause), but 29% of DAs begin at the second word or later in a speech segment. 4% of DAs contain an internal pause and

[1] In Manuvinakurike et al. (2016), we describe a related application of incremental speech segmentation in a variant rapid dialogue game with a different corpus. In that paper, we focus on fine-grained segmentation of referential utterances that would all be labeled as D-T in this paper. The model presented here is shallower and more general, focusing on high-level DA labels.

[2] We excluded from annotation the training rounds in the corpus, where players practiced playing the game.

Example	# IPUs	# DAs	Annotation
1	1	5	PFB that's okay ‖ D-T um this castle has a ‖ ST oh gosh this is hard ‖ D-T this castle is tan ‖ D-T it's at a diagonal with a blue sky
2	1	2	D-T and it's got lemon in it ‖ Q-YN you got it
3	1	2	PFB okay ‖ D-T this is the christmas tree in front of a fireplace
4	1	2	EC fireplace ‖ As-I got it
5	2	2	D-M all right ‖ D-T this is ... this is this is the brown circle and it's not hollow
6	3	1	D-T this is a um ... tan or light brown ... box that is clear in the middle
7	3	2	D-M all right ‖ D-T he's got he's got that ... that ... first uh the first finger and the thumb pointing up
8	3	2	ST um golly ‖ DT this looks like a a a ... ginseng ... uh of some sort
9	2	4	ST oh wow ‖ D-M okay ‖ D-T this one ... looks it has gray ‖ D-T a lotta gray on this robot

Table 1: Examples of annotated DA types, DA boundaries (‖), and IPU boundaries (...). The number of IPUs and DAs in each example are indicated.

Number of DAs	0	1	2	≥ 3
% of speech segments	3	74	18	5

Table 2: The distribution in the number of DAs whose first word is within a speech segment.

thus span multiple speech segments.

4 Technical Approach

The goal for our incremental DA segmentation component is to segment the recognized speech for a speaker into individual DA segments and to assign these segments to the 18 DA classes in Table 9. We aim to do this in an incremental (word-by-word) manner, so that information about the DAs within a speech segment becomes available before the user stops or pauses their speech.

Figure 2 shows the incremental operation of our sequential pipeline for DA segmentation and classification. We use Kaldi for ASR, and we adapt the work of Plátek and Jurčíček (2014) for incremental ASR using Kaldi. The pipeline is invoked after each new partial ASR result becomes available (i.e., every 100ms), at which point all the recognized speech is resegmented and reclassified in a *restart incremental* (Schlangen and Skantze, 2011) design. The input to the pipeline includes all the recognized speech from one speaker (including multiple IPUs) for one target image subdialogue.

In our sequential pipeline, the first step is to use sequential tagging with a CRF (Conditional Random Field) (Lafferty et al., 2001) implemented in Mallet (McCallum, 2002) to perform the segmentation. The segmenter tags each word as either the beginning (B) of a new DA segment or as a continuation of the current DA segment (I).[3] Then, each

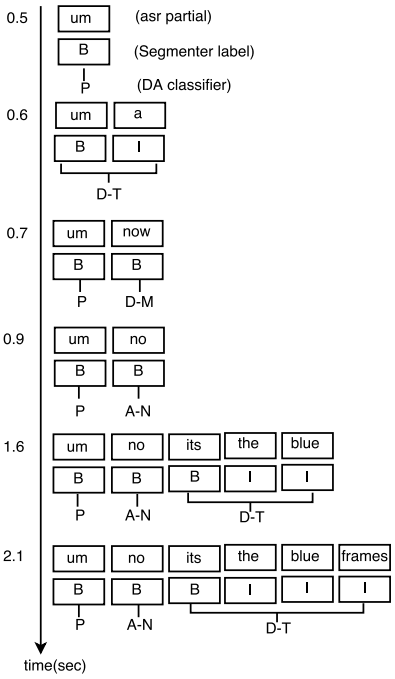

Figure 2: The operation of the pipeline on selected ASR partials (with time index in seconds).

resulting DA segment is classified into one of 18 DA labels using an SVM (Support Vector Machine) classifier implemented in Weka (Hall et al., 2009).

4.1 Features

Prosodic Features We use word-level prosodic features similar in nature to Litman et al. (2009). The alignment between words and computed prosodic features is achieved using a forced aligner (Baumann and Schlangen, 2012) to generate word-level timing information. For each word, we first

[3]Note that our annotation scheme completely partitions our data, with every word belonging to a segment and receiving a DA label. We have therefore elected not to adopt BIO (Begin-Inside-Outside) tagging.

obtain pitch and RMS values every 10ms using InproTK (Baumann and Schlangen, 2012). Because pitch and energy features can be highly variable across users, our pitch and energy features are represented as z-scores that are normalized for the current user up to the current word. For the pitch and RMS values, we obtain the max, min, mean, variance and the co-efficients of a second degree polynomial. Pause durations at word boundaries provide an additional useful feature (Kolář et al., 2006; Zimmermann, 2009). All numeric features are discretized into bins. We currently use prosody for segmentation but not classification.[4]

Lexico-syntactic & contextual features We use word unigrams along with the corresponding part-of-speech (POS) tags, obtained using Stanford CORENLP (Manning et al., 2014), as a feature for both the segmentation and the DA classifier. Words with a low frequency ($<$10) are substituted with a low frequency word symbol. The top level constituent category from a syntactic parse of the DA segment is also used.

Several contextual features are included. The role of the speaker (Director or Matcher) is included as a feature. Previously recognized DA labels from each speaker are included. Another feature is added to assist with the Echo Confirmation (EC) DA, which applies when a speaker repeats verbatim a phrase recently spoken by the other interlocutor. For this we use features to mark word-level unigrams that appeared in recent speech from the other interlocutor. Finally, a categorical feature indicates which of 18 possible image sets (e.g. bikes as in Figure 1) is under discussion; simpler images tend to have shorter segments.[5]

4.2 Discussion of Machine Learning Setup

A salient alternative to our sequential pipeline approach – also adopted for example by Ang et al. (2005) – is to use a joint classification model to solve the segmentation and classification problems simultaneously, potentially thereby improving performance on both problems (Petukhova and Bunt, 2014; Morbini and Sagae, 2011; Zimmermann, 2009; Warnke et al., 1997). We performed an initial test using a joint model and found, unlike the finding reported by Zimmermann (2009), that for our corpus a joint approach performed markedly worse than our sequential pipeline.[6] We speculate that this is due to the relative sparsity of data on rarer DA types in our relatively small corpus. For similar reasons, we have not yet tried to use RNN-based approaches such as LSTMs, which tend to require large amounts of training data.

[4]For the experiments reported in this paper, prosodic features were calculated offline, but they could in principle be calculated in real-time.

[5]The image set feature affects the performace of the segmenter only slightly.

Condition	Transcripts (T)	Segment Boundaries (S)	DA labels (D)
HT-HS-HD	Human	Human	Human
HT-HS-AD	Human	Human	Automated
HT-AS-AD	Human	Automated	Automated
AT-AS-AD	ASR	Automated	Automated

Table 3: Conditions for evaluating DA segmentation and classification.

5 Experiment and Results

We report on two experiments. In the first experiment, we train our DA segmentation pipeline using the annotated corpus of Section 3.1 and report results on the observed DA segment boundaries (Section 5.1) and DA class labels (Section 5.2). In the second experiment, presented in Section 5.3, we report on a policy simulation that investigates the effect of our incremental DA segmentation pipeline on a baseline automated agent's performance.

For the first experiment, we use a hold-one-pair-out cross-validation setup where, for each fold, the dialogue between one pair of players is held out for testing, while automated models are trained on the other pairs. To evaluate our pipeline, we use four data conditions, summarized in Table 3, that represent increasing amounts of automation in the pipeline. These conditions allow us to better understand the sources for observed errors in segment boundaries and/or DA labels. Our notation for these conditions is a compact encoding of the data sources used to create the transcripts of user speech, the segment boundaries, and the DA labels. Our reference annotation, described in Section 3.1, is notated HT-HS-HD (human transcript, human segment boundaries, human DA labels). Example segmentations for each condition are in Table 4.

5.1 Evaluation of DA Segment Boundaries

In this evaluation, we ignore DA labels and look only at the identification of DA boundaries (notated by $\|$ in Table 4, and encoded using B and I tags in our segmenter). For this evaluation, we use human

[6]We used a joint CRF model similar to the BI coding of Zimmermann (2009).

Condition	# IPUs	Example	
HT-HS-HD	1	(a)	A-N um no ‖ D-T it's the blue frame ‖ D-T but it's an orange seat and an orange handle
HT-HS-AD	1	(b)	A-N um no ‖ D-T it's the blue frame ‖ D-T but it's an orange seat and an orange handle
HT-AS-AD	1	(c)	P um ‖ A-N no ‖ D-T it's the blue frame ‖ D-T but it's an orange seat ‖ D-T and an orange handle
AT-AS-AD	1	(d)	A-N on no ‖ D-T it's the blue frame ‖ D-T but it's an orange seat ‖ D-T and orange ‖ A-N no

Table 4: Examples of DA boundaries (‖) and DA labels in each condition.

Condition	Features	**Accuracy**	**F-Score**		**DSER**
			B tag	I tag	
1-DA-per-IPU		0.78	0.23	0.87	0.26
HT-AS-AD	Prosody (I)	0.72	0.62	0.69	0.42
HT-AS-AD	Lexico-Syntactic & Contextual (II)	0.90	0.82	0.82	0.31
HT-AS-AD	I+II	0.91	0.83	0.84	0.30
Human annotator		0.95	0.91	0.94	0.15

Table 5: Observed DA segmentation performance. These results consider only DA boundaries.

transcripts and compare the boundaries in our reference annotations (HT-HS-HD) to the boundaries inferred by our automated pipeline (HT-AS-AD).[7]

In Table 5, we present results for versions of our pipeline that use three different feature sets: only prosody features (I), only lexico-syntactic and contextual features (II), and both (I+II). We include also a simple 1-DA-per-IPU baseline that assumes each IPU is a single complete DA; it assigns the first word in each IPU a B tag and subsequent words an I tag. Finally, we also include numbers based on an independent human annotator using the subset of our annotated corpus that was annotated by two human annotators. For this subset, we use our main annotator as the reference standard and evaluate the other annotator as if their annotation were a system's hypothesis.[8]

The reported numbers include word-level accuracy of the B and I tags, F-score for each of the B and I tags, and the DA segmentation error rate (DSER) metric of Zimmermann et al. (2006). DSER measures the fraction of reference DAs whose left and right boundaries are not exactly replicated in the hypothesis. For example, in Table 4, the reference (a) contains three DAs, but only the boundaries of the second DA (*it's the blue frame*) are exactly replicated in hypothesis (c). This yields a DSER of 2/3 for this example.

We find that our automated pipeline (HT-AS-AD) with all features performs the best among the pipeline methods, with word-level accuracy of 0.91 and DSER of 0.30. Its performance however is worse than an independent human annotator, with double the DSER. This suggests there remains room for improvement at boundary identification. The 1-DA-per-IPU baseline does well on the common case of single-IPU DAs, but it fails ever to segment an IPU into multiple DAs. We use the pipeline with all features in the following sections.

[7]We evaluate our DA segmentation performance using human transcripts, rather than ASR, as this allows a simple direct comparison of inferred DA boundaries.

[8]For comparison, the chance-corrected kappa value for word-level boundaries is 0.92; see Section 3.1.

Condition	Metrics used for human transcripts			Alignment-based metrics	
	DER	Strict	Lenient	Levenshtein-Lenient	CER
HT-HS-AD	0.39	0.09	0.09	0.07	0.27
HT-AS-AD	0.72	0.38	0.15	0.12	0.39
AT-AS-AD				0.39	0.52

Table 6: Observed DA classification and joint segmentation+classification performance.

5.2 Evaluation of DA Class Labels

In this evaluation, we consider DA labels assigned to recognized DA segments using several types of metrics. We summarize our results in Table 6.

Metrics used for human transcripts We first compare our reference annotations (HT-HS-HD) to the performance of our automated pipeline *when provided human transcripts as input*. For this comparison, we use three error rate metrics (Lenient, Strict, and DER) from the DA segmentation literature that are intuitively applied when the token sequence being segmented and labeled is identical (or at least isomorphic) to the annotated token sequence. Lower is better for these. The Lenient and Strict metrics (Ang et al., 2005) are based on the DA labels assigned to each individual word (by way of the label of the DA segment that contains that word). Lenient is a per-token DA label error

rate that ignores DA segment boundaries.[9] In Table 6, this error rate is 0.09 when human-annotated boundaries are fed into our DA classifier (HT-HS-AD) and 0.15 when automatically-identified boundaries are used (HT-AS-AD).

Strict and DER are boundary-sensitive metrics. Strict is a per-token error rate that requires each token to receive the correct DA label and also to be part of a DA segment whose exact boundaries appear in the reference annotation. This is a much higher standard.[10] Dialogue Act Error Rate (DER) (Zimmermann et al., 2006) is the fraction of reference DAs whose left and right boundaries and label are perfectly replicated in the hypothesis. While the reported boundary-sensitive error rate numbers (0.38 and 0.72) may appear to be high, many of these boundary errors may be relatively innocuous from a system standpoint. We return to this below.

Alignment-based metrics We also report two additional metrics that are intuitively applied even when the word sequence being segmented and classified is only a noisy approximation to the word sequence that was annotated, i.e. under an ASR condition such as AT-AS-AD. The Concept Error Rate (CER) is a word error rate (WER) calculation (Chotimongkol and Rudnicky, 2001) based on a minimum edit distance alignment of the DA tags (using one DA tag per DA segment). Our fully automated pipeline (AT-AS-AD) has a CER of 0.52.

We also report an analogous word-level metric which we call 'Levenshtein-Lenient'. To our knowledge this metric has not previously been used in the literature. It replaces each word in the reference and hypothesis with the DA tag that applies to it, and then computes a WER on the DA tag sequence. It is thus a Lenient-like metric that can be applied to DA segmentation based on ASR results. Our automated pipeline (AT-AS-AD) scores 0.39.

DA multiset precision and recall metrics When ASR is used, the CER and Levenshtein-Lenient metrics give an indication of how well you are doing at replicating the ordered sequence of DA tags. But in building a system, sometimes the sequence is less of a concern, and what is desired is a breakdown in terms of precision and recall per DA tag. Many dialogue systems use policies that are triggered when a certain DA type has occurred in the user's speech (such as an agent that processes yes (A-Y) or no (A-N) answers differently, or a director agent for the RDG-Image game that moves on when the matcher performs As-I ("got it")). For such systems, exact DA boundaries and even the order of DAs is not of paramount importance so long as a correct DA label is produced around the time the user performs the DA.

[9] E.g. in Table 4 (c), the only Lenient error is at word *um*.

[10] E.g. in Table 4 (c), only the four words *it's the blue frame* would count as non-errors on the Strict standard.

Condition	HT-HS-AD		HT-AS-AD		AT-AS-AD	
	P	**R**	**P**	**R**	**P**	**R**
D-T	0.98	0.98	0.85	0.95	0.79	0.88
As-I	0.97	0.97	0.74	0.96	0.73	0.68
NG	0.84	0.89	0.72	0.88	0.63	0.50
PFB	0.67	0.65	0.50	0.77	0.42	0.60
ST	0.92	0.92	0.71	0.63	0.41	0.31
Q-YN	0.94	0.85	0.86	0.85	0.55	0.52
AN	0.90	0.90	0.70	0.67	0.42	0.32
A-Y	0.79	0.79	0.65	0.75	0.59	0.58

Table 7: DA multiset precision and recall metrics for a sample of higher-frequency DA tags.

We therefore define a more permissive measure that looks only at precision and recall of DA labels within a sample of user speech. As an example, in (a) in Table 4, there is one A-N label and two D-T labels. In (d), there are two A-N labels and 3 D-T labels. Ignoring boundaries, we can represent as a multiset the collection of DA labels in a reference A or hypothesis H, and compute standard multiset versions of precision and recall for each DA type. For reference, a formal definition of multiset precision $P(\mathrm{DA}_i)$ and recall $R(\mathrm{DA}_i)$ for DA type DA_i is provided in the appendix.

We report these numbers for our most common DA types in Table 7. Here, we continue to use the speech of one speaker during a target image subdialogue as the unit of analysis. The data show that precision and recall generally decline for all DA types as automation increases in the conditions from left to right. We do relatively well with the most frequent DA types, which are D-T and As-I. A particular challenge, even in human transcript+segment condition HT-HS-AD, is the DA tag PFB. In a manual analysis of common error types, we found that the different DA labels used for very short utterances like 'okay' (D-M, PFB, As-I) and 'yeah' (A-Y, PFB, As-I) are often confused. We believe this type of error could be reduced through a combination of improved features, collapsed DA categories, and more detailed annotation guidelines. ASR errors also often cause DA errors; see e.g. Table 4 (d).

	image set	total time(sec)	total points p	p/sec	NLU accuracy	avg sec/image
All DAs	Pets	984.7	182	0.18	0.77	4.15
	Zoo	921.1	203	0.22	0.79	3.60
	Cocktails	1300.3	153	0.12	0.60	5.12
	Bikes	1630.9	126	0.08	0.47	6.12
Only D-T	Pets	992.0	184	0.19	0.78	4.19
	Zoo	932.8	198	0.21	0.77	3.64
	Cocktails	1326.7	155	0.12	0.61	5.22
	Bikes	1678.4	130	0.08	0.49	6.29

Table 8: Overall performance of the eavesdropper simulation on the unsegmented data (All DAs) and the automatically segmented data (Only D-T) identified with our pipeline (AT-AS-AD).

5.3 Evaluation of Simulated Agent Dialogues

Motivation. In prior work (Paetzel et al., 2015), we developed an automated agent called Eve which plays the matcher role in the RDG-Image game and has been evaluated in a live interactive study with 125 human users. Our prior work underscored the critical importance of pervasive incremental processing in order for Eve to achieve her highest performance in terms of points scored and also the best subjective user impressions. In this second experiment, we perform an offline investigation into the potential impact on our agent's image-matching performance if we integrate the incremental DA segmentation pipeline from this paper.

We take the "fully-incremental" version of Eve from Paetzel et al. (2015) as our baseline agent in this experiment. Briefly, this version of Eve includes the same incremental ASR used in our new DA segmentation pipeline (Plátek and Jurčíček, 2014), incremental language understanding to identify the target image (Naive Bayes classification), and an incremental dialogue policy that uses parameterized rules. See Paetzel et al. (2015) for full details.

The baseline agent's design focuses on the most common DA types in our RDG-Image corpora: D-T for the director (constituting 60% of director DAs), and As-I for the matcher (constituting 46% of matcher DAs). Effectively, the baseline agent assumes every word the user says is describing the target, and uses an optimized policy to decide the right moment to commit to a selection (As-I) or ask the user to skip the image (As-S). Eve's typical interaction pattern is illustrated in Figure 3.

This experiment is narrowly focused on the impact of using the pipeline to segment out only the D-T DAs and to use only the words from detected D-Ts in the target image classifier and the agent's policy decisions. Changing the agent pipeline from using the director's full utterance towards only taking the D-T tagged words into account could potentially have a negative impact on the baseline agent's performance. For example, for the fully automated condition AT-AS-AD in Table 7, D-T has precision 0.79 and recall 0.88. The 0.88 recall suggests that some D-T words will be lost (in false negative D-Ts) by integrating the new DA segmenter. Additionally, as shown in Figure 2, the recognized words and whether they are tagged as D-T can change dynamically as new incremental ASR results arrive, and this instability could undermine some of the advantage of segmentation. On the other hand, by excluding non-D-T text from consideration, there is a potential to decrease noise in the agent's understanding and improve the agent's accuracy or speed.

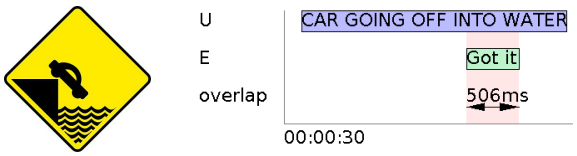

Figure 3: Eve (E) identifies a target image.

Experiment. As an initial investigation into the issues described above, we adopt the "Eavesdropper" framework for policy simulation and training detailed in Paetzel et al. (2015). In an Eavesdropper simulation, the director's speech from pre-recorded target image dialogues is provided to the agent, and the agent simulates alternative policy decisions as if it were in the matcher role. We have found that higher cross-validation performance in these offline simulations has translated to higher performance in live interactive human-agent studies (Paetzel et al., 2015).

We created a modified version of our agent that uses the fully automated pipeline (AT-AS-AD) to pass only word sequences tagged as D-T to the agent's language understanding component (a target image classifier), effectively ignoring other DA types. Tagging is performed every 100 ms on each new incremental output segment published by the

ASR. We then compare the performance of our baseline and modified agent in a cross-validation setup, using an Eavesdropper simulation to train and test the agents. We use a corpus of human-human gameplay that includes 18 image sets and game data from both the lab-based corpus of 32 games described in Section 3.1 and also the web-based corpus of an additional 98 human-human RDG-Image games described in Manuvinakurike and DeVault (2015). Each simulation yields a new trained NLU (target image classifier, based either on all text or only on D-T text) and a new optimized policy for when the agent should perform As-I vs. As-S. Within the simulations, for each target image, we compute whether the agent would score a point and how long it would spend on each image.

Table 8 summarizes the observed performance in these simulations for four sample image sets in the two agent conditions. All results are calculated based on leave-one-user-out training and a policy optimized on points per second. A Wilcoxon-Mann-Whitney Test on all 18 image sets indicated that, between the two conditions, there is no significant difference in the total time ($Z = -0.24$, $p = .822$), total points scored ($Z = -0.06$, $p = .956$), points per second ($Z = -0.06$, $p = .956$), average seconds per image ($Z = -0.36$, $p = .725$), or NLU accuracy ($Z = -0.13$, $p = .907$).

These encouraging results suggest that our incremental DA segmenter achieves a performance level that is sufficient for it to be integrated into our agent, enabling the incremental segmentation of other DA types without significantly compromising (or improving) the agent's current performance level. These results provide a complementary perspective on the various DA classification metrics reported in Section 5.2.

The current baseline agent (Paetzel et al., 2015) can only generate As-I and As-S dialogue acts. In future work, the fully automated pipeline presented here will enable us to expand the agent's dialogue policies to support additional patterns of interaction beyond its current skillset. For example, the agent would be better able to understand and react to a multi-DA user utterance like *and handles got it?* in Figure 1. By segmenting out and understanding the Q-YN *got it?*, the agent would be able to detect the question and answer with an A-Y like *yeah*. Overall, we believe the ability to understand the natural range of director's utterances will help the agent to create more natural interaction patterns, which might receive a better subjective rating by the human dialogue partner and in the end might even achieve a better overall game performance, as ambiguities can be resolved quicker and the flow of communication can be more efficient.

6 Conclusion & Future Work

In this paper, we have defined and evaluated a sequential approach to incremental DA segmentation and classification. Our approach utilizes prosodic, lexico-syntactic and contextual features, and achieves an encouraging level of performance in offline analysis and in policy simulations. We have presented our results in terms of existing metrics for DA segmentation and also introduced additional metrics that may be useful to other system builders. In future work, we will continue this line of work by incorporating dialogue policies for additional DA types into the interactive agent.

Acknowledgments

We thank our reviewers. This work was supported by the National Science Foundation under Grant No. IIS-1219253 and by the U.S. Army. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views, position, or policy of the National Science Foundation or the United States Government, and no official endorsement should be inferred. David Schlangen acknowledges support by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

Image credits. We number the images in Figure 1 from 1-8 moving left to right. Thanks to Hugger Industries (1)[11], Eric Sorenson (6)[12] and fixedgear (8)[13] for images published under CC BY-NC-SA 2.0. Thanks to Eric Parker (2)[14] and cosmo flash (4)[15] for images published under CC BY-NC 2.0, and to Richard Masonder / Cyclelicious (3)[16] (5)[17] and Florian (7)[18] for images published under CC BY-SA 2.0.

[11] https://www.flickr.com/photos/huggerindustries/3929138537/

[12] https://www.flickr.com/photos/ahpook/5134454805/

[13] http://www.flickr.com/photos/fixedgear/172825187/

[14] https://www.flickr.com/photos/ericparker/6050226145/

[15] http://www.flickr.com/photos/cosmoflash/9070780978/

[16] http://www.flickr.com/photos/bike/3221746720/

[17] https://www.flickr.com/photos/bike/3312575926/

[18] http://www.flickr.com/photos/fboyd/6042425285/

References

Jeremy Ang, Yang Liu, and Elizabeth Shriberg. 2005. Automatic dialog act segmentation and classification in multiparty meetings. In *ICASSP*, pages 1061–1064.

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.

Linda Bell, Johan Boye, and Joakim Gustafson. 2001. Real-time handling of fragmented utterances. In *The NAACL Workshop on Adaption in Dialogue Systems*, pages 2–8.

Ananlada Chotimongkol and Alexander I Rudnicky. 2001. N-best speech hypotheses reordering using linear regression.

David DeVault, Kenji Sagae, and David Traum. 2011. Incremental interpretation and prediction of utterance meaning for interactive dialogue. *Dialogue and Discourse*, 2(1):143–70.

Raul Fernandez and Rosalind W Picard. 2002. Dialog act classification from prosodic features using support vector machines. In *Speech Prosody 2002, International Conference*.

Luciana Ferrer, Elizabeth Shriberg, and Andreas Stolcke. 2003. A prosody-based approach to end-of-utterance detection that does not require speech. In *Proc. IEEE ICASSP*, pages 608–611.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Hanae Koiso, Yasuo Horiuchi, Syun Tutiya, Akira Ichikawa, and Yasuharu Den. 1998. An analysis of turn-taking and backchannels based on prosodic and syntactic features in japanese map task dialogs. *Language and Speech*, 41(3-4):295–321.

Jáchym Kolář, Elizabeth Shriberg, and Yang Liu. 2006. On speaker-specific prosodic models for automatic dialog act segmentation of multi-party meetings. In *Interspeech*, volume 1.

Kazunori Komatani, Naoki Hotta, Satoshi Sato, and Mikio Nakano. 2015. User adaptive restoration for incorrectly-segmented utterances in spoken dialogue systems. In *SIGDIAL*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289.

Kornel Laskowski and Elizabeth Shriberg. 2010. Comparing the contributions of context and prosody in text-independent dialog act recognition. In *ICASSP*, pages 5374–5377. IEEE.

Piroska Lendvai and Jeroen Geertzen. 2007. Token-based chunking of turn-internal dialogue act sequences. In *SIGDIAL*, pages 174–181.

Diane J Litman, Mihai Rotaru, and Greg Nicholas. 2009. Classifying turn-level uncertainty using word-level prosody. In *INTERSPEECH*, pages 2003–2006.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL: System Demonstrations*, pages 55–60.

Ramesh Manuvinakurike and David DeVault, 2015. *Natural Language Dialog Systems and Intelligent Assistants*, chapter Pair Me Up: A Web Framework for Crowd-Sourced Spoken Dialogue Collection, pages 189–201.

Ramesh Manuvinakurike, Casey Kennington, David DeVault, and David Schlangen. 2016. Real-time understanding of complex discriminative scene descriptions. In *SIGDIAL*.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Fabrizio Morbini and Kenji Sagae. 2011. Joint identification and segmentation of domain-specific dialogue acts for conversational dialogue systems. In *Proceedings of ACL: Human Language Technologies: short papers*, pages 95–100.

Mikio Nakano, Noboru Miyazaki, Jun-ichi Hirasawa, Kohji Dohsaka, and Takeshi Kawabata. 1999. Understanding unsegmented user utterances in real-time spoken dialogue systems. In *ACL*, pages 200–207.

Maike Paetzel, David Nicolas Racca, and David DeVault. 2014. A multimodal corpus of rapid dialogue games. In *LREC*, May.

Maike Paetzel, Ramesh Manuvinakurike, and David DeVault. 2015. "So, which one is it?" The effect of alternative incremental architectures in a high-performance game-playing agent. In *SIGDIAL*.

Volha Petukhova and Harry Bunt. 2014. Incremental recognition and prediction of dialogue acts. In *Computing Meaning*, pages 235–256. Springer.

Ondřej Plátek and Filip Jurčíček. 2014. Free on-line speech recogniser based on Kaldi ASR toolkit producing word posterior lattices. In *SIGDIAL*.

Antoine Raux and Maxine Eskenazi. 2008. Optimizing endpointing thresholds using dialogue features in a spoken dialogue system. In *SIGDIAL*, pages 1–10.

Ryo Sato, Ryuichiro Higashinaka, Masafumi Tamoto, Mikio Nakano, and Kiyoaki Aikawa. 2002. Learning decision trees to determine turntaking by spoken dialogue systems. In *Proceedings of ICSLP-02*, pages 861–864.

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

Elizabeth Shriberg, Andreas Stolcke, Dilek Hakkani-

Tür, and Gökhan Tür. 2000. Prosody-based automatic segmentation of speech into sentences and topics. *Speech communication*, 32(1):127–154.

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

Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Masashi Takeuchi, Norihide Kitaoka, and Seiichi Nakagawa. 2004. Timing detection for realtime dialog systems using prosodic and linguistic information. In *Speech Prosody 2004*.

Nigel G Ward and David DeVault. 2015. Ten challenges in highly interactive dialog systems. In *AAAI 2015 Spring Symposium*.

V. Warnke, R. Kompe, H. Niemann, and E. Nth. 1997. Integrated dialog act segmentation and classification using prosodic features and language models. In *Proc. 5th Europ. Conf. on Speech, Communication, and Technology*, volume 1.

Matthias Zimmermann, Yang Liu, Elizabeth Shriberg, and Andreas Stolcke, 2006. *Second International Workshop, MLMI 2005*, chapter Toward Joint Segmentation and Classification of Dialog Acts in Multiparty Meetings, pages 187–193.

Matthias Zimmermann. 2009. Joint segmentation and classification of dialog acts using conditional random fields. In *Interspeech*.

A Appendix

Definition of multiset precision and recall Let $\mathcal{D} = \{\mathrm{DA}_1, ..., \mathrm{DA}_n\}$ be the set of possible DAs. Let $A : \mathcal{D} \rightarrow \mathbb{Z}_{\geq 0}$ be an annotated reference DA multiset and $H : \mathcal{D} \rightarrow \mathbb{Z}_{\geq 0}$ be a hypothesized DA multiset. The multiset intersection for each DA type DA_i is:

$$(A \cap H)(\mathrm{DA}_i) = \min(A(\mathrm{DA}_i), H(\mathrm{DA}_i))$$

DA-level multiset precision $P(\mathrm{DA}_i)$ and recall $R(\mathrm{DA}_i)$ are then defined as:

$$P(\mathrm{DA}_i) = (A \cap H)(\mathrm{DA}_i) \ / \ H(\mathrm{DA}_i)$$

$$R(\mathrm{DA}_i) = (A \cap H)(\mathrm{DA}_i) \ / \ A(\mathrm{DA}_i)$$

DA	Description	Example
D-T	Describe target	this is the christmas tree in front of a fireplace
As-I	Assert Identified	got it
NG	Non-game utterances	okay there i saw the light go on
PFB	Positive feedback	okay
ST	Self-talk statements	ooh this is gonna be tricky
P	Filled pause	uh
D-M	Discourse marker	alright
Q-YN	Yes-No question	is it on something white
A-Y	Yes answer	yeah
EC	Echo confirmation	the blue
As-M	Matcher assertions	it didn't let me do it
Q-C	Clarification question	bright orange eyes?
A-D	Action directive	oh oh wait hold on
A-N	No answer	no, nah
H	Hedge	i don't know what it is
Q-D	Disjunctive question	are we talking dark brown or like caramel brown
Q-Wh	Wh-question	what color's the kitty
As-S	Assert skip	i'm gonna pass on that

Table 9: The complete list of DAs in the annotated RDG-Image corpus.

DA	All	Dir	Mat	DA	All	Dir	Mat
D-T	41	60	0	EC	2	.5	6
As-I	15	0	46	As-M	2	0	4
NG	11	9	11	Q-C	2	.5	4
PFB	8	10	7	A-D	1	.3	2
ST	4	4	4	A-N	.5	.7	.2
P	4	6	2	H	.5	.7	0
D-M	3	5	.2	Q-Wh	.3	0	.5
Q-YN	3	.6	7	As-S	.1	0	.1
A-Y	2	3	1	Q-D	.4	0	1.2

Table 10: DA distribution. We report the relative percentages for each DA out of all DAs, director DAs, and matcher DAs, respectively.