Dialogue and Discourse 2(1) (2011) 143-170 doi: 10.5087/dad.2011.107

Incremental Interpretation and Prediction of Utterance Meaning for Interactive Dialogue

David DeVault DEVAULT@ICT.USC.EDU
USC Institute for Creative Technologies
12015 Waterfront Drive
Playa Vista, CA 90094 USA

Kenji Sagae SAGAE@ICT.USC.EDU
USC Institute for Creative Technologies
12015 Waterfront Drive
Playa Vista, CA 90094 USA

David Traum TRAUM@ICT.USC.EDU
USC Institute for Creative Technologies
12015 Waterfront Drive
Playa Vista, CA 90094 USA

Editor: David Schlangen, Hannes Rieser

Abstract

We present techniques for the incremental interpretation and prediction of utterance meaning in dialogue systems. These techniques open possibilities for systems to initiate responsive overlap behaviors during user speech, such as interrupting, acknowledging, or completing a user's utterance while it is still in progress. In an implemented system, we show that relatively high accuracy can be achieved in understanding of spontaneous utterances before utterances are completed. Further, we present a method for determining when a system has reached a point of maximal understanding of an ongoing user utterance, and show that this determination can be made with high precision. Finally, we discuss a prototype implementation that shows how systems can use these abilities to strategically initiate system completions of user utterances. More broadly, this framework facilitates the implementation of a range of overlap behaviors that are common in human dialogue, but have been largely absent in dialogue systems.

1. Introduction

Human spoken dialogue is highly interactive, including feedback on the speech of others while the speech is progressing (so-called "backchannels" (Yngve 1970)), monitoring of addressees and other listener feedback (Nakano et al. 2003), fluent turn-taking with little or no delays (Sacks et al. 1974), and overlaps of various sorts, including collaborative completions (Goodwin 1979), repetitions and other grounding moves (Clark and Schaefer 1987), and interruptions. Interruptions can be either to advance the new speaker's goals (which may not be related to interpreting the other's speech) or in order to prevent the speaker from finishing. Few of these behaviors can be replicated by current spoken dialogue systems. Most of these behaviors require first an ability to perform incremental interpretation, and second, an ability to predict the final meaning and timing of the utterance.

Submitted 1/2010; Accepted 12/2010; Published online 5/2011

Most spoken dialogue systems wait until the user stops speaking before trying to understand and react to what the user is saying. In particular, in a typical dialogue system pipeline, it is only once the user's spoken utterance is complete that the results of automatic speech recognition (ASR) are sent on to natural language understanding (NLU) and dialogue management, which then triggers generation and synthesis of the next system utterance. While this style of interaction is adequate and perhaps even preferred for some applications (Funakoshi et al. 2010), it enforces a rigid pacing that can be unnatural and inefficient for mixed-initiative dialogue. To achieve more flexible turn-taking with human users, for whom turn-taking and feedback at the sub-utterance level is natural and common, the system needs to engage in incremental processing, in which interpretation components are activated, and in some cases decisions are made, before the user utterance is complete.

Incremental interpretation enables more rapid response, since most of the utterance can be interpreted before utterance completion (Skantze and Schlangen 2009). It also enables giving early feedback (e.g., head nods and shakes, facial expressions, gaze shifts, and verbal backchannels) to signal how well things are being perceived, understood, and evaluated (Allwood et al. 1992).

For some responsive behaviors, one must go beyond incremental interpretation and predict some aspects of the full utterance before it has been completed. For behaviors such as complying with the evocative function (Allwood 1995) or intended perlocutionary effect (Sadek 1991), grounding by demonstrating (Clark and Schaefer 1987), or interrupting to avoid having the utterance be completed, one must predict the semantic content of the full utterance from a partial prefix fragment. For other behaviors, such as timing a reply to have little or no gap, grounding by saying the same thing at the same time (called "chanting" by Hansen et al. (1996)), performing collaborative completions (Clark and Wilkes-Gibbs 1986), or some corrections, it is important not only to predict the meaning, but also the form and timing of the remaining part of the utterance.

We have begun to explore these issues in the context of the dialogue behavior of virtual humans (Rickel and Johnson 1999) (also called embodied conversational agents (Cassell et al. 2000)) for multiparty negotiation role-playing (Traum et al. 2008). In these kinds of systems, human-like behavior is a goal, since the purpose is to allow a user to practice this kind of dialogue with the virtual humans in training for real negotiation dialogues. The more realistic the characters' dialogue behavior is, the more kinds of negotiation situations can be adequately trained for. We discuss these systems further in Section 2.

In the remainder of the paper, we present the progress we have made to date in implementing responsive behaviors in our systems. This paper summarizes and expands on several previously reported results. In Sagae et al. (2009), we presented our first results at prediction of semantic content from partial speech recognition hypotheses, looking at length of the speech hypothesis as a general indicator of semantic accuracy in understanding. Sections 3 and 4 summarize this previous work, and also provide additional details and examples for our predictive model. Section 3 presents our basic approach to understanding *complete* user utterances, and Section 4 shows how we have extended this approach to enable predictive, incremental understanding of partial utterances.

In DeVault et al. (2009), we extended our previous work by incorporating additional features of real-time incremental interpretation to develop a more nuanced prediction model that can accurately identify moments of maximal understanding within individual spoken utterances. We summarize this work in Section 5. In Section 6, we explore the potential value of this new ability using a prototype implementation that collaboratively completes user utterances when the system becomes confident about how the utterance will end. This section provides a more detailed evaluation and discussion of the characteristics of this prototype utterance completion capability than was present in

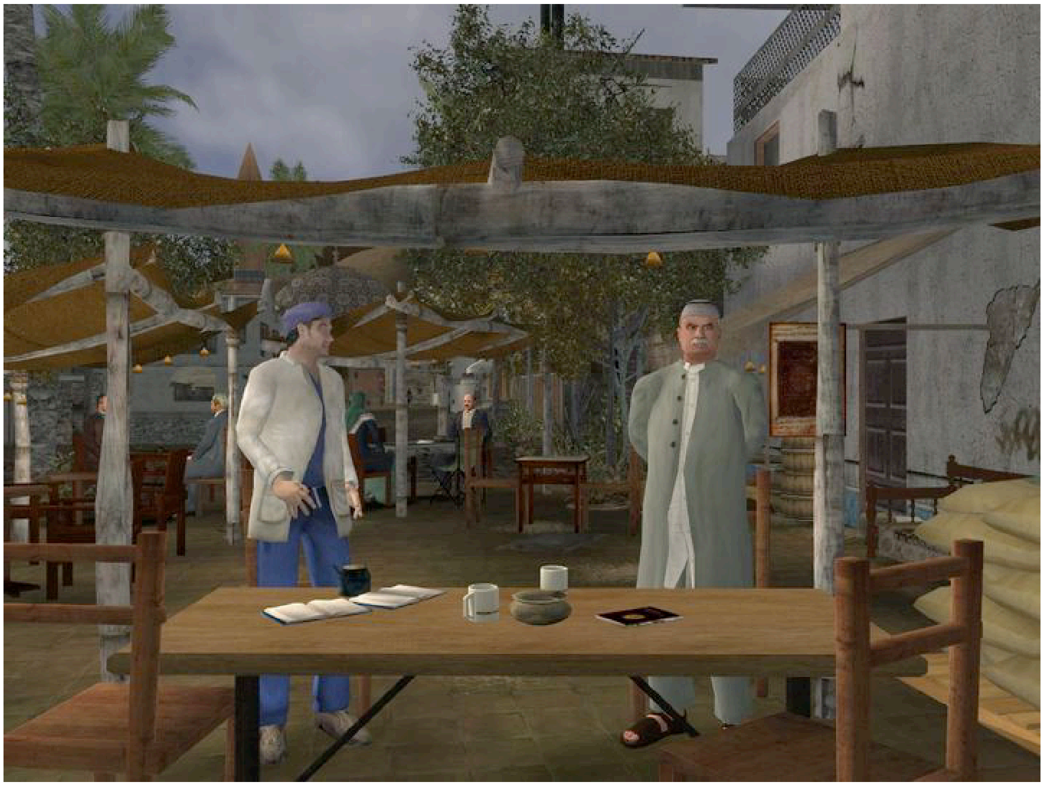


Figure 1: SASO-EN negotiation in the cafe: Dr. Perez (left) looking at Elder al-Hassan.

DeVault et al. (2009). Finally, this paper adds a new, expanded discussion of all of these techniques in Section 7, including discussion of their limitations, potential role in system building, and future research directions. We conclude in Section 8.

We believe the predictive models presented in this paper will be more broadly useful in implementing responsive overlap behaviors such as rapid grounding using completions, confirmation requests, or paraphrasing, as well as other kinds of interruptions and multi-modal displays.

2. Domain setting

The case study we present in this paper is taken from the SASO-EN virtual human system (Hartholt et al. 2008, Traum et al. 2008). This system is designed to allow a trainee to practice multi-party negotiation skills by engaging in face to face negotiation with virtual humans. The scenario involves a negotiation about the possible re-location of a medical clinic in an Iraqi village. A human trainee plays the role of a US Army captain, and there are two virtual humans that he negotiates with: Doctor Perez, the head of the NGO clinic, and a local village elder, al-Hassan. The doctor's main objective is to treat patients. The elder's main objective is to support his village. The captain's main objective is to move the clinic out of the marketplace, ideally to the US army base. Figure 1 shows the doctor and elder in the midst of a negotiation, from the perspective of the trainee. Figure 2 presents a sample dialogue from this domain.

1 C Hello Doctor Perez.

2 D Hello captain.

3 E Hello captain.

4 C Thank you for meeting me.

5 E How may I help you?

6 C I have orders to move this clinic to a camp near the US base.

7 E We have many matters to attend to.

8 C I understand, but it is imperative that we move the clinic out of this area.

9 E This town needs a clinic.

10 D We can't take sides.

11 C Would you be willing to move downtown?

12 E We would need to improve water access in the downtown area, captain.

13 C We can dig a well for you.

14 D Captain, we need medical supplies in order to run the clinic downtown.

15 C We can deliver medical supplies downtown, Doctor.

16 E We need to address the lack of power downtown.

17 C We can provide you with power generators.

18 E Very well captain, I agree to have the clinic downtown.

19 E Doctor, I think you should run the clinic downtown.

20 D Elder, the clinic downtown should be in an acceptable condition before we move.

21 E I can renovate the downtown clinic, Doctor.

22 D OK, I agree to run the clinic downtown, captain.

23 C Excellent.

24 D I must go now.

25 E I must attend to other matters.

26 C Goodbye.

26 D Goodbye.

26 E Farewell, sir.

Figure 2: Successful negotiation dialogue between C, a captain (human trainee), D, a doctor (virtual human), and E, a village elder (virtual human).

The system has a fairly typical set of processing components for virtual humans or dialogue systems, including ASR (mapping speech to words), NLU (mapping from words to semantic frames), dialogue interpretation and management (handling context, dialogue acts, reference and deciding what content to express), NLG (mapping frames to words), non-verbal generation, and synthesis and realization. The doctor and elder use the same ASR and NLU components, but have different modules for the other processing, including different models of context and goals, and different output generators.

3. Understanding complete user utterances in the SASO-EN system

We begin by reviewing the technical approach we have used to understand complete user utterances in SASO-EN. We first define the NLU task in Section 3.1, and then describe the data used in our experiments in Section 3.2. We present our NLU model in Section 3.3, and summarize our results in Section 3.4. We will turn to the understanding of partial user utterances in Section 4.

3.1 The natural language understanding task

The NLU module used in the SASO-EN virtual human system takes the output of ASR as input, and produces domain-specific semantic frames as output. For all the results presented in this paper, we have used Sonic (Pellom 2001) as the ASR component. Alternative ASR components are compatible with our techniques, however. Our techniques require only for the ASR to provide the top-ranking text hypothesis for the utterance, and for the ASR to be able to provide its top hypothesis incrementally, as user speech progresses and additional audio is captured. Indeed, in more recent work, we have begun to use the PocketSphinx ASR (Huggins-Daines et al. 2006) as a substitute for Sonic.

Given the top-ranking text hypothesis from the ASR component, the NLU module analyzes the utterance into a domain-specific semantic frame representation. These frames are intended to capture much of the meaning of the utterance, although a dialogue manager further enriches the frame representations with pragmatic information (Traum 2003). The NLU output frame representation is an attribute-value matrix (AVM), where the attributes and values represent semantic information that is linked to a domain-specific ontology and task model (Hartholt et al. 2008). Complicating the NLU task is the relatively high word error rate (0.54) in ASR of user speech input, given conversational speech in a complex domain and an untrained broad user population.

Figure 3 shows an example of NLU input and output for an utterance where the user attempts to address complaints about lack of power in the proposed location for the clinic. In the figure, we show both the ideal, hypothetical condition in which the ASR output perfectly matches the user utterance (*we are prepared to give you guys generators for electricity downtown*) as well as the actual condition in which the ASR output includes several errors. Given either of these ASR results as input, the NLU output is the same in our system, which illustrates the desired robustness to ASR errors. The figure shows the NLU frame, which is an AVM, linearized using a path-value notation. The linearized semantic frame in this example corresponds to the AVM shown in Figure 4.

3.2 Data

The work described in the following sections is based on a corpus of user data consisting of 4,500 spoken user utterances, which were spread across a number of different dialogue sessions in the SASO-EN system. Each user utterance was transcribed, and each of the complete utterance transcripts was manually annotated with a “gold standard” semantic frame. This annotated frame is viewed as the desired output from the NLU module for that complete utterance. Utterances that were judged to be out-of-domain (13.7% of the corpus) were assigned to a “garbage” frame, with no semantic content.

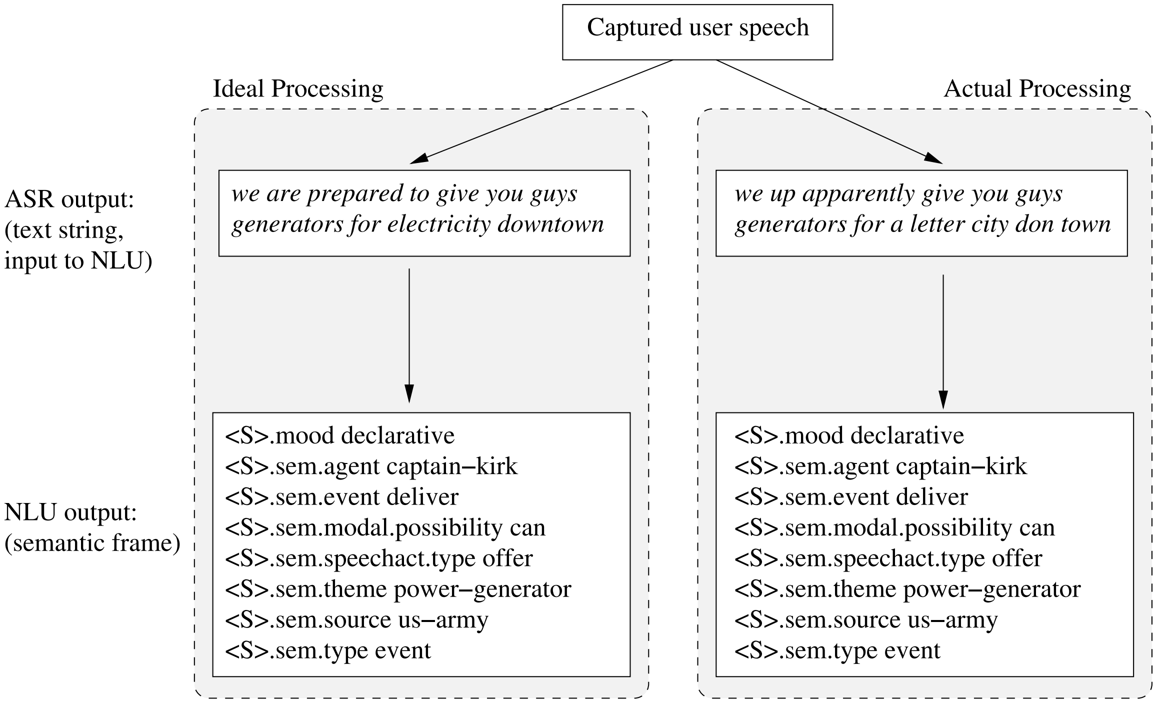

Captured user speech
Ideal Processing
Actual Processing
ASR output:
(text string,
input to NLU)
we are prepared to give you guys generators for electricity downtown
we up apparently give you guys generators for a letter city don town
NLU output:
(semantic frame)
<S>.mood declarative
<S>.sem.agent captain−kirk
<S>.sem.event deliver
<S>.sem.modal.possibility can
<S>.sem.speechact.type offer
<S>.sem.theme power−generator
<S>.sem.source us−army
<S>.sem.type event
<S>.mood declarative
<S>.sem.agent captain−kirk
<S>.sem.event deliver
<S>.sem.modal.possibility can
<S>.sem.speechact.type offer
<S>.sem.theme power−generator
<S>.sem.source us−army
<S>.sem.type event

Figure 3: Example of NLU input and output for a specific user utterance.

$$\begin{bmatrix} mood : declarative & \\ sem : & \begin{bmatrix} type : event \\ agent : captain - kirk \\ event : deliver \\ theme : power - generator \\ source : us - army \\ modal : \begin{bmatrix} possibility : can \end{bmatrix} \\ speech - act : \begin{bmatrix} type : offer \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 4: AVM utterance representation.

For training, development and evaluation of data-driven NLU models, the corpus was divided as follows. Approximately 10% of the utterances were set aside for evaluation, and another 10% were designated as the development test corpus for the NLU module. The remaining 80% was used for training the NLU module. The training set included 136 distinct frames. Note that the development and test sets were chosen so that all the utterances from the same dialogue session were kept in the same set, but sessions were chosen at random for inclusion in the development and test sets.

3.3 NLU as multiclass classification with mxNLU

The SASO-EN NLU module, mxNLU (Sagae et al. 2009), is based on a data-driven approach to language understanding that treats the task as multiclass classification. We use a supervised machine learning technique to learn a mapping from user utterances to semantic representations. More specifically, mxNLU uses Maximum Entropy (ME) models (Berger et al. 1996), where entire semantic frames are treated as classes, and features used for utterance classification are derived from the text string produced by ASR. Our task is then to estimate a model that determines the conditional probability that the user has expressed the meaning represented by a specific frame y, given that ASR output x was observed, which we denote by $p(y|x)$. The model is estimated using the Maximum Entropy framework (see Berger et al. (1996) for details) according to a set of training examples $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, and has the following parametric form:

$$p(y|x) = \frac{1}{Z(x)} \exp\left(\sum_i \lambda_i f_i(x, y)\right) \tag{1}$$

where $Z(x) = \sum_y \exp(\sum_i \lambda_i f_i(x, y))$ is a normalizing constant determined by the requirement that $\sum_y p(y|x) = 1$ for all x, each f_i is a feature function, and the parameters λ_i can be thought of as “weights” that reflect the importance of the corresponding feature f_i. The features used by mxNLU come from a fixed set of templates used to generate feature functions similar to the one below:

$$f_j(x, y) = \begin{cases} 1 & \text{if } y = y' \text{ and } x \text{ includes the word “generators”} \\ 0 & otherwise \end{cases}$$

Although it is possible to define different types of features that are specific to each individual classes (frames) y', in practice mxNLU considers the same types of features for every frame. This means that the ME model includes features such as the one shown above for every y' corresponding to each specific frame observed in the training examples. Each feature function f_i is then intended to capture a specific event observed in the output of ASR (such as the occurrence of the word “generators”, or the occurrence of the word sequence “we can”), and the corresponding parameter λ_i is intended to capture the relationship between that event and a specific output frame. The events captured by the templates used to generate features for mxNLU are: each word in the input string (bag-of-words representation of the input), each bigram (consecutive words), each pair of any two words in the input, and the number of words in the input string. Table 1 illustrates what information about a specific text string is captured by the feature functions used in mxNLU. Note that the features generated for the bigram “we can” and the pair of words “we can” are distinct, and each is treated separately by the classifier, according to their respective λ_i parameters.

The training examples used to create the ME model used by mxNLU come from the corpus of user data described in Section 3.2, which includes ASR output from audio files containing recorded

ASR output (NLU input)	*we can help you*
Words	we, can, help, you
Bigrams	(<S> we), (we can), (can help), (help you), (you </S>)
Pairs of words	(we can), (we help), (we you), (can help), (can you), (help you)
Length	4

Table 1: Information about a specific utterance, represented as a text string produced by ASR, captured by the feature functions used in mxNLU. The special tokens <S> and </S> mark the beginning and end of the utterance, respectively.

utterances directed at the system and corresponding semantic representations. Each pair of user utterance (represented as the output of ASR) and semantic frame (represented as a linearized AVM) is used to create an individual training example. Each training example is composed of a feature vector (generated using the feature templates listed above) and a corresponding class (i.e. an entire semantic frame, which is treated as an atomic entity, rather than compositionally).

Given the set of training examples, the Maximum Entropy classifier should learn, for example, that when the word "generators" appears in the output of ASR, the correct output frame is likely to be one that includes the value *power-generator*. Although such a mapping between a specific feature of the NLU input and individual keys or values in NLU output frames cannot be made explicitly in our NLU model, since frames are never decomposed into individual keys and values, it can still hold implicitly, under the assumption that input instances that include the word "generators" are associated with frames that include the value *power-generator* in the training data.

Another consequence of our choice not to decompose frames is that the process of mapping utterances to semantic representations is divorced from the complexity and other details of the frame representation. For example, the frame language could allow for co-indexation of values and reentrant structures (although these are not used in SASO-EN frames), which mxNLU would have no problems dealing with. However, mxNLU would not treat these phenomena productively, but rather would simply memorize each instance as a part of a specific frame, which it could reproduce at run time. In other words, mxNLU can only produce a finite number of output frames, which correspond exactly to the set of frames present in the training data. Although this is a clear theoretical limitation of the approach of classifying utterances into entire frames at once, the precise impact of this limitation on the efficacy of this NLU approach depends on specific characteristics of the dialogue system and the semantic representation it uses. In our corpus of user data, only 136 distinct frames were observed for the 3,500 user utterances in the training set, and the development set contained no frames that had not been observed in the training set. This indicates that the lack of productive ability in the NLU approach should have little negative impact, if any, in SASO-EN, which is a system that supports relatively rich natural language interaction in a limited domain.

3.4 NLU performance on complete ASR output

Although mxNLU produces entire frames as output, we evaluate NLU performance by looking at precision and recall of the attribute-value pairs (or *frame elements*) that compose frames. Precision represents the portion of frame elements produced by mxNLU that were correct, and recall represents the portion of frame elements in the gold-standard annotations that were proposed by mxNLU. By using precision and recall of frame elements, we take into account that certain pairs of frames are more similar than others and also allow more meaningful comparative evaluation with NLU modules that construct a frame from sub-elements or for cases when the actual frame is not in the training set.

When mxNLU is trained on complete ASR utterances in the training set, and tested on complete ASR utterances in the development test set, the F-score of frame elements is 0.76, with precision at 0.78 and recall at 0.74. To gain insight on what the upperbound on the accuracy of the NLU module might be, we also trained the classifier using features extracted from gold-standard manual transcription (instead of ASR output), and tested the accuracy of analyses of gold-standard transcriptions (which would not be available at run-time in the dialogue system). Under these ideal conditions, the NLU F-score is 0.87. Training on gold-standard transcriptions and testing on ASR output produces results with a lower F-score, 0.74.

4. Understanding partial user utterances in SASO-EN

In this section, we refine the non-incremental NLU module described in Section 3 in a way that enables the incremental understanding and prediction of utterance meaning during user speech.

There is a growing body of work on incremental processing in dialogue systems. Some of this work has demonstrated overall improvements in system responsiveness and user satisfaction; e.g. Aist et al. (2007), Skantze and Schlangen (2009). Several research groups, inspired by psycholinguistic models of human processing, have also been exploring technical frameworks that allow diverse contextual information to be brought to bear during incremental processing; e.g. Kruijff et al. (2007), Aist et al. (2007).

While this work often assumes or suggests it is possible for systems to understand partial user utterances even before those utterances are complete, this premise has generally not been given detailed quantitative study (though see Schlangen et al. (2009), Heintze et al. (2010)). In this section, we demonstrate and explore quantitatively the extent to which our SASO-EN dialogue system can anticipate what an utterance means, on the basis of partial ASR results, before the utterance is complete.

A useful starting point is to observe the length, measured in words, of complete spoken user utterances in our corpus. Figure 5 shows the utterance length distribution in the development set. Roughly half of the utterances in our data contain six words or more, and the average utterance length is 5.9 words. Since the ASR module is capable of sending partial results to the NLU module even before the user has finished an utterance, in principle the dialogue system can start understanding and even responding to user input as soon as enough words have been uttered to give the system some indication of what the user means, or even what the user will have said once the utterance

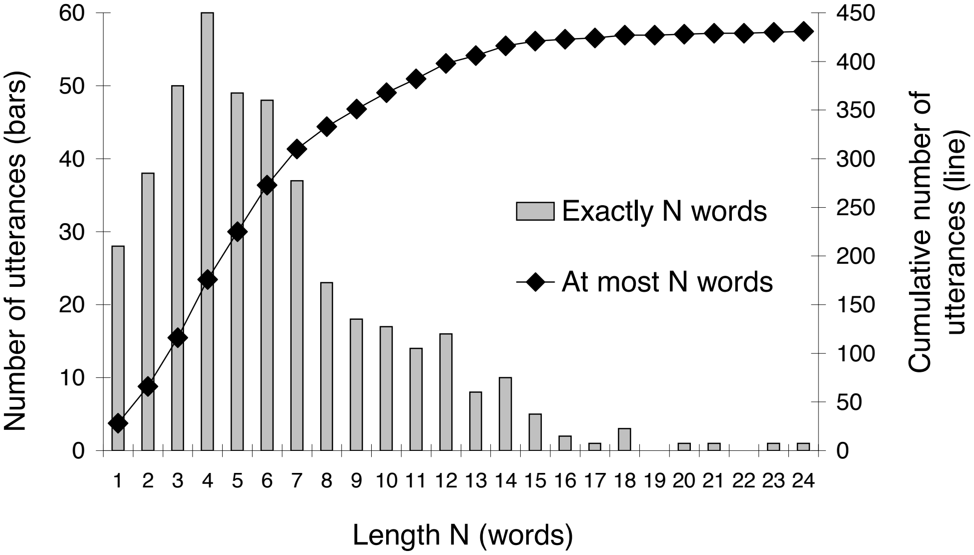


Figure 5: Length of utterances in the development set.

is completed. To measure the extent to which our NLU module can predict the frame for an input utterance when it sees only a partial ASR result with the first N words, we examine two aspects of NLU with partial ASR results. The first is *correctness* of the NLU output with partial ASR results of varying lengths, if we take the gold-standard manual annotation for the entire utterance as the correct frame for any of the partial ASR results for that utterance. The second is *stability*: how similar the NLU output with partial ASR results of varying lengths is to what the NLU result would have been for the entire utterance.

The most straightforward way to perform NLU of partial ASR results is simply to process the partial utterances using the NLU module trained on complete ASR output. However, as we will show, better results may be obtained by training separate NLU models for analysis of partial utterances of different lengths.

To train these separate NLU models, we first ran the audio of the utterances in the training data through our ASR module, in 200 millisecond increments, and recorded all the partial ASR results for each utterance. Then, to train a model to analyze partial ASR results containing N words, we used only those partial ASR results in our training set containing N words (unless the complete ASR result contained less than N words, in which case we simply used the complete ASR result). In some cases, multiple partial ASR results for a single utterance contained the same number of words, and we used the last partial result with the appropriate number of words.[1]

We trained separate NLU models for N varying from one to ten.

[1] At run-time, this can be closely approximated by taking the partial utterance immediately preceding the first partial utterance of length $N + 1$.

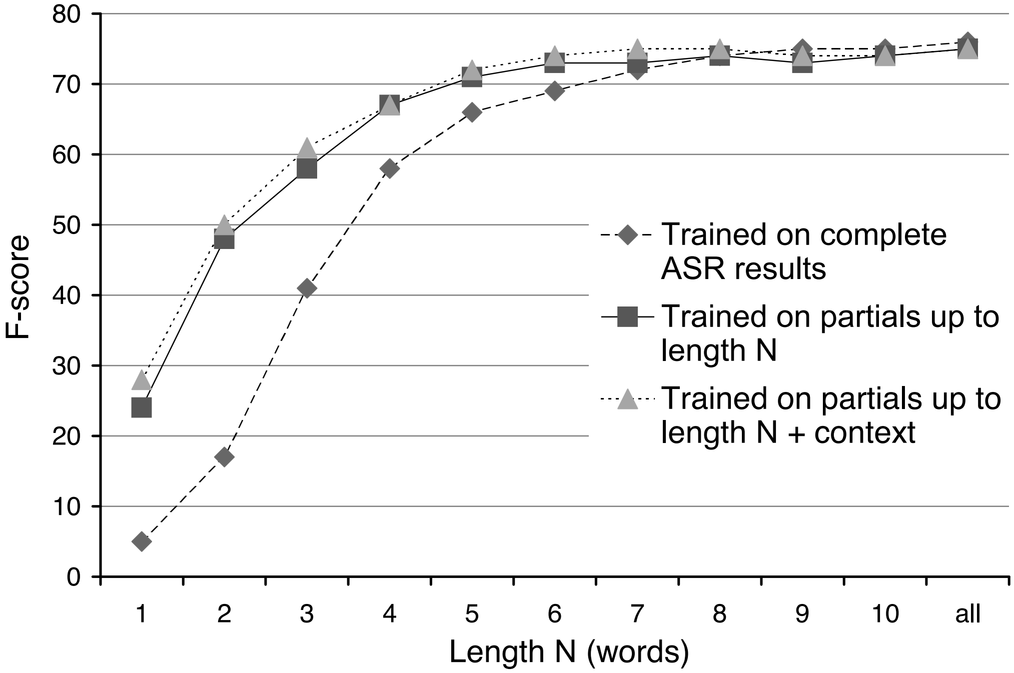


Figure 6: Correctness for three NLU models on partial ASR results up to N words.

Figure 6 shows the F-score for frames obtained by processing partial ASR results up to length N using three variants of mxNLU. The dashed line is our baseline NLU model, trained on complete utterances only (model 1). The solid line shows the results obtained with length-specific NLU models (model 2), and the dotted line shows results for length-specific models that also use features that capture dialogue context (model 3). In these experiments, we used unigram and bigram word features extracted from the most recent system utterance to represent context, but found that these context features did not improve NLU performance. Our final NLU approach for partial ASR hypotheses is then to train separate models for specific lengths, using hypotheses of that length during training (solid line in figure 6).

As seen in Figure 6, there is a clear benefit to training NLU models specifically tailored for partial ASR results. Training a model on partial utterances with four or five words allows for relatively high F-score of frame elements (0.67 and 0.71, respectively, compared to 0.58 and 0.66 when the same partial ASR results are analyzed using model 1). Considering that half of the utterances are expected to have more than five words (based on the length of the utterances in the training set), allowing the system to start processing user input when four or five-word partial ASR results are available provides interesting opportunities. Targeting partial results with seven words or more is less productive, since the time savings are reduced, and the gain in accuracy is modest.

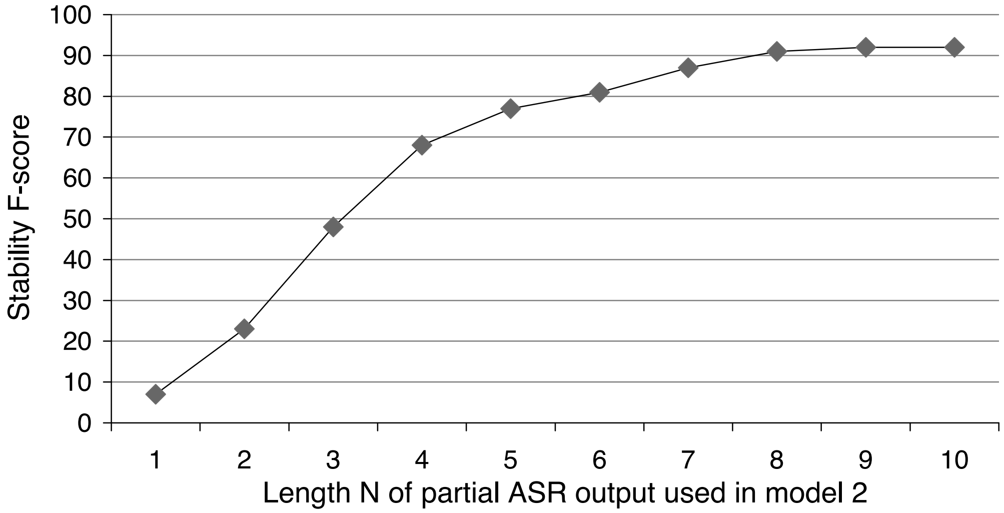


Figure 7: Stability of NLU results for partial ASR results up to length N.

The context features used in model 3 did not provide substantial benefits in NLU accuracy. It is possible that other ways of representing context or dialogue state may be more effective. This is an area we are currently investigating.

Finally, Figure 7 shows the *stability* of NLU results produced by model 2 for partial ASR utterances of varying lengths. This is intended to be an indication of how much the frame assigned to a partial utterance differs from the ultimate NLU output for the entire utterance. This ultimate NLU output is the frame assigned by model 1 for the complete utterance. Stability is then measured as the F-score between the output of model 2 for a particular partial utterance, and the output of model 1 for the corresponding complete utterance. A stability F-score of 1.0 would mean that the frame produced for the partial utterance is identical to the frame produced for the entire utterance. Lower values indicate that the frame assigned to a partial utterance is revised significantly when the entire input is available. As expected, the frames produced by model 2 for partial utterances with at least eight words match closely the frames produced by model 1 for the complete utterances. Although the frames for partial utterances of length six are almost as accurate as the frames for the complete utterances (Figure 6), Figure 7 indicates that these frames are still often revised once the entire input utterance is available.

5. Detecting points of maximal understanding

In this section, we present a strategy that uses machine learning to more closely characterize the performance of a maximum entropy based incremental NLU module, such as the mxNLU module described in Section 4. Our aim is to identify strategic points in time, as a specific utterance is occurring, when the system might react with confidence that the interpretation will not significantly improve during the rest of the utterance (DeVault et al. 2009). This reaction could take several forms, including providing feedback, or, as described in Section 6 an agent might use this information to opportunistically choose to initiate a completion of the user's utterance.

5.1 Motivating example

Figure 8 illustrates the incremental output of mxNLU as a user asks, *elder do you agree to move the clinic downtown?* Our ASR processes captured audio in 200ms chunks. The figure shows the partial ASR result after the ASR has processed each 200ms of audio, along with the F-score achieved by mxNLU on each of these partials. Note that the NLU F-score fluctuates somewhat as the ASR revises its incremental hypotheses about the user utterance, but generally increases over time.

For the purpose of initiating an overlapping response to a user utterance such as this one, the agent needs to be able (in the right circumstances) to make an assessment that it has already understood the utterance "well enough", based on the partial ASR results that are currently available. We have implemented a specific approach to this assessment which views an utterance as understood "well enough" if the agent would not understand the utterance any better than it currently does even if it were to wait for the user to finish their utterance (and for the ASR to finish interpreting the complete utterance).

Concretely, Figure 8 shows that after the entire 2800ms utterance has been processed by the ASR, mxNLU achieves an F-score of 0.91. However, in fact, mxNLU already achieves this maximal F-score at the moment it interprets the partial ASR result *elder do you agree to move the* at 1800ms. The agent therefore could, in principle, initiate an overlapping response at 1800ms without sacrificing any accuracy in its understanding of the user's utterance.

Of course the agent does not automatically realize that it has achieved a maximal F-score at 1800ms. To enable the agent to make this assessment, we have trained a classifier, which we call MAXF, that can be invoked for any specific partial ASR result, and which uses various features of the ASR result and the current mxNLU output to estimate whether the NLU F-score for the current partial ASR result is at least as high as the mxNLU F-score would be if the agent were to wait for the entire utterance.

5.2 Machine learning setup

To facilitate the construction of our MAXF classifier, we identified a range of potentially useful features that the agent could use at run-time to assess its confidence in mxNLU's output for a given partial ASR result. These features are exemplified in Figure 9, and include: K, the number of partial results that have been received from the ASR; N, the length (in words) of the current partial ASR result; Entropy, the entropy in the probability distribution mxNLU assigns to alternative output frames; $P_{\max}$, the probability mxNLU assigns to the most probable output frame; NLU, the most probable output frame (represented for convenience as fI, where I is an integer index corresponding to a specific complete frame). We also define MAXF (GOLD), a boolean value giving the ground truth about whether mxNLU's F-score for this partial is at least as high as mxNLU's F-score for the final partial for the same utterance. In the example, note that MAXF (GOLD) is true for each partial where mxNLU's F-score ($F(K)$) is $\geq$ 0.91, the value achieved for the final partial (*elder do you agree to move the clinic downtown*). Of course, the actual F-score $F(K)$ is not available at run-time, and so cannot serve as an input feature for the classifier.

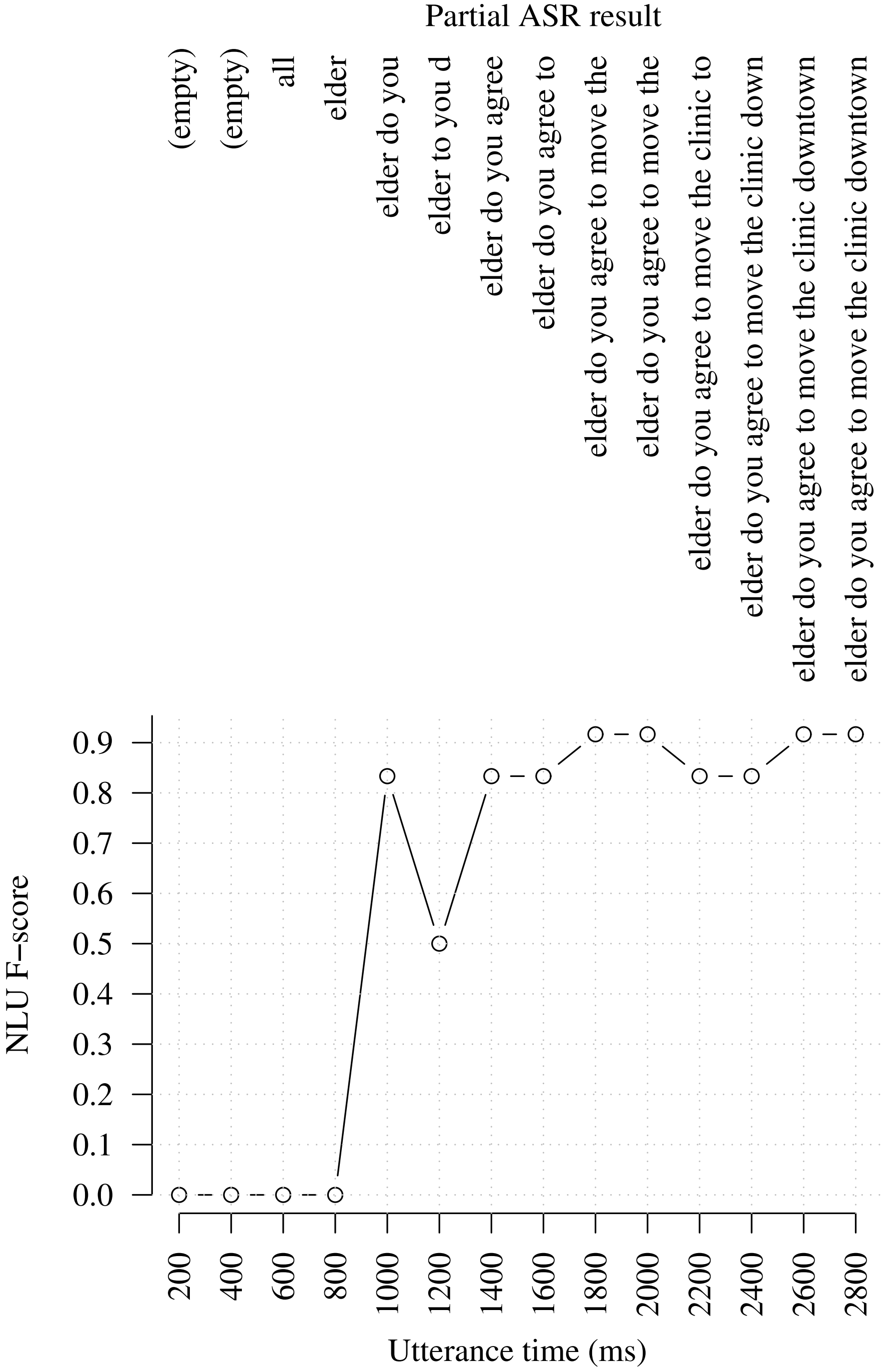


Figure 8: Incremental interpretation of a user utterance.

		MAXF model training features					
Partial ASR result	$F(K)$	K	N	Entropy	$P_{\max}$	NLU	MAXF (GOLD)
(empty)	0.00	1	0	2.96	0.48	f82	FALSE
(empty)	0.00	2	0	2.96	0.48	f82	FALSE
all	0.00	3	1	0.82	0.76	f72	FALSE
elder	0.00	4	1	0.08	0.98	f39	FALSE
elder do you	0.83	5	3	1.50	0.40	f68	FALSE
elder to you d	0.50	6	3	1.31	0.75	f69	FALSE
elder do you agree	0.83	7	4	1.84	0.35	f68	FALSE
elder do you agree to	0.83	8	5	1.40	0.61	f68	FALSE
elder do you agree to move the	0.91	9	7	0.94	0.49	f10	TRUE
elder do you agree to move the	0.91	10	7	0.94	0.49	f10	TRUE
elder do you agree to move the clinic to	0.83	11	9	1.10	0.58	f68	FALSE
elder do you agree to move the clinic down	0.83	12	9	1.14	0.66	f68	FALSE
elder do you agree to move the clinic downtown	0.91	13	9	0.50	0.89	f10	TRUE
elder do you agree to move the clinic downtown	0.91	14	9	0.50	0.89	f10	TRUE

Figure 9: Features used to train the MAXF model.

Our general aim, then, is to train a classifier, MAXF, whose output predicts the value of MAXF (GOLD) as a function of the input features. To create a data set for training and evaluating this classifier, we observed and recorded the values of these features for the 6068 partial ASR results in a corpus of ASR output for 449 actual user utterances.[2]

We chose to train a decision tree using Weka's J48 training algorithm (Witten and Frank 2005).[3] To assess the trained model's performance, we carried out a 10-fold cross-validation on our data set.[4] We present our results in the next section.

5.3 Results

We will present results for a trained decision tree model that reflects a specific precision/recall tradeoff. In particular, given our aim to enable an agent to sometimes initiate overlapping speech, while minimizing the chance of making a wrong assumption about the user's meaning, we selected a model with high precision at the expense of lower recall. Various precision/recall tradeoffs are possible in this framework; the choice of a specific tradeoff is likely to be system and domain-dependent and motivated by specific design goals.

We evaluate our model using several features which are exemplified in Figure 10. These include MAXF (PREDICTED), the trained MAXF classifier's output (TRUE or FALSE) for each partial;

[2] This corpus was not part of the training data for mxNLU.

[3] Of course, other classification models could be used.

[4] All the partial ASR results for a given utterance were constrained to lie within the same fold, to avoid training and testing on the same utterance.

		MAXF model evaluation features		
K	$F(K)$	$\Delta F(K)$	$T(K)$	MAXF (PREDICTED)
1	0.00	-0.91	2.6	FALSE
2	0.00	-0.91	2.4	FALSE
3	0.00	-0.91	2.2	FALSE
4	0.00	-0.91	2.0	FALSE
5	0.83	-0.08	1.8	FALSE
6	0.50	-0.41	1.6	FALSE
7	0.83	-0.08	1.4	FALSE
8	0.83	-0.08	1.2	FALSE
9 (= K_{MAXF})	0.91	0.00 (=$\Delta F(K_{\text{MAXF}})$)	1.0	TRUE
10	0.91	0.00	0.8	TRUE
11	0.83	-0.08	0.6	FALSE
12	0.83	-0.08	0.4	FALSE
13	0.91	0.00	0.2	TRUE
14	0.91	0.00	0.0	TRUE

Figure 10: Features used to evaluate the MAXF model.

K_{MAXF}, the first partial number for which MAXF (PREDICTED) is TRUE; $\Delta F(K) = F(K) - F(K_{\text{final}})$, the "loss" in F-score associated with interpreting partial K rather than the final partial K_{final} for the utterance; $T(K)$, the remaining length (in seconds) in the user utterance at each partial.

We begin with a high level summary of the trained MAXF model's performance, before discussing more specific impacts of interest in the dialogue system. We found that our trained model predicts that MAXF = TRUE for at least one partial in 79.2% of the utterances in our corpus. For the remaining utterances, the trained model predicts MAXF = FALSE for all partials. The precision/recall/F-score of the trained MAXF model are 0.88/0.52/0.65 respectively. The high precision means that 88% of the time that the model predicts that F-score is maximized at a specific partial, it really is. On the other hand, the lower recall means that only 52% of the time that F-score is in fact maximized at a given partial does the model predict that it is.

For the 79.2% of utterances for which the trained model predicts MAXF = TRUE at some point, Figure 11 shows the amount of time in seconds, $T(K_{\text{MAXF}})$, that remains in the user utterance at the time partial K_{MAXF} becomes available from the ASR. The mean value is 1.6 seconds; as the figure shows, the time remaining varies from 0 to nearly 8 seconds per utterance. This represents a substantial amount of time that an agent could use strategically, for example by immediately initiating overlapping speech (perhaps in an attempt to improve communication efficiency), or by exploiting this time to plan an optimal response to the user's utterance.

However, it is also important to understand the cost associated with interpreting partial K_{MAXF} rather than waiting to interpret the final ASR result K_{final} for the utterance. We therefore analyzed

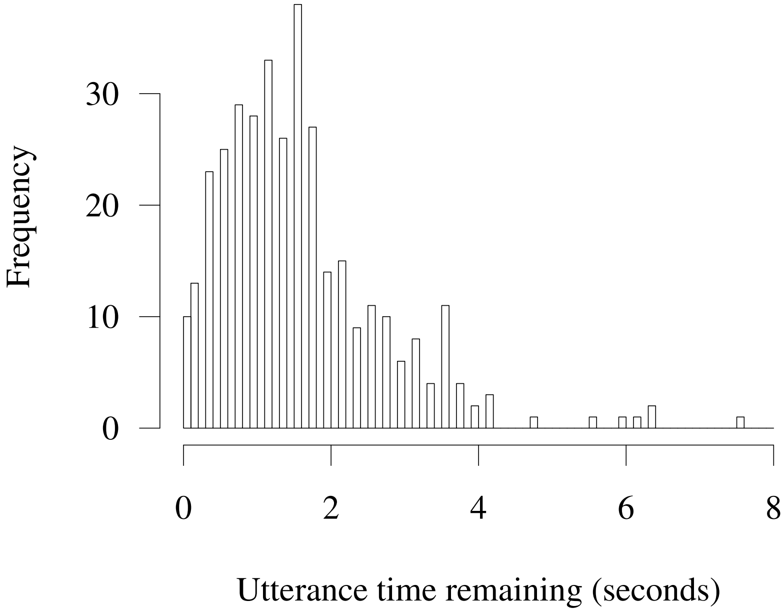


Figure 11: Distribution of $T(K_{\mathrm{MAXF}})$.

the distribution in $\Delta F(K_{\mathrm{MAXF}}) = F(K_{\mathrm{MAXF}}) - F(K_{\mathrm{final}})$. This value is at least 0.0 if mxNLU's output for partial K_{MAXF} is no worse than its output for K_{final} (as intended). The distribution is given in Figure 12. As the figure shows, 62.35% of the time (the median case), there is no difference in F-score associated with interpreting K_{MAXF} rather than K_{final}. 10.67% of the time, there is a loss of -1, which corresponds to a completely incorrect frame at K_{MAXF} but a completely correct frame at K_{final}. The converse also happens 2.52% of the time: mxNLU's output frame is completely correct at the early partial but completely incorrect at the final partial. The remaining cases are mixed. While the median is no change in F-score, the mean case is a loss in F-score of -0.1484. This is the mean penalty in NLU performance that could be paid in exchange for the potential gain in communication efficiency suggested by Figure 11.

6. Completing user utterances

To illustrate one use of the techniques described in the previous sections, we have implemented a prototype module that performs *user utterance completion*. This allows an agent to jump in during a user's utterance, and say a completion of the utterance before it is finished. This type of completion is often encountered in human-human dialogue, e.g., Skuplik (1999) as reported by Poesio and Rieser (2010) found 126 completions in a corpus of 3675 utterances. Completions may be used, for example, for grounding or for bringing the other party's turn to a conclusion. In this section, we present and discuss a prototype implementation that builds on the incremental understanding and

$\Delta F(K_{\mathrm{MAXF}})$ range	Percent of utterances
-1	10.67%
$(-1, 0)$	17.13%
0	62.35%
$(0, 1)$	7.30%
1	2.52%
mean($\Delta F(K_{\mathrm{MAXF}})$)	-0.1484
median($\Delta F(K_{\mathrm{MAXF}})$)	0.0000

Figure 12: The distribution in $\Delta F(K_{\mathrm{MAXF}})$, the "loss" associated with interpreting partial K_{MAXF} rather than K_{final}.

prediction models we have presented above, and which has allowed us to equip one of our virtual humans, Doctor Perez, with an ability to perform utterance completions.

The decision to complete another speaker's utterance is potentially a complex one. In our prototype implementation, we have incorporated the mxNLU and MAXF classifiers into a relatively simple model of this decision-making, as a way to begin to explore the value and limitations of these classifiers in the decision to complete a user's utterance.

The process proceeds as follows. The first step is for the agent to recognize whether it has reached a moment of maximal understanding for the utterance. In our prototype, Doctor Perez never attempts to complete an utterance until MAXF becomes true. This is a heuristic that means Doctor Perez will never try to complete an utterance if it thinks its understanding could be improved by waiting. (Note that even if the classifier judges MAXF to be true, it is not guaranteed that the utterance has been understood correctly. The MAXF classifier may have made a mistake, and the utterance would in fact be understood better with additional user speech. Or, it may be the case that the utterance will never be understood well, and the MAXF classifier has detected this unfortunate state of affairs before the utterance concludes.)

As discussed in Section 5, MAXF often (but not always) becomes true before the user has completed the utterance. NLU is performed on partial ASR hypotheses as they become available, and MAXF decides whether the agent's understanding of the current partial hypothesis is likely to improve given more time. Once MAXF indicates that the agent's understanding is already maximized, our prototype takes the current partial ASR hypothesis, and attempts to generate text to complete it in a way that is fluent and agrees with the predicted meaning of the utterance the user has in mind.

The generation of the surface text for completions takes advantage of the manual transcriptions in the corpus of utterances used to train the NLU module. For each frame that the agent understands, our training set contains a set of user utterances that correspond to the meaning in that frame. At the point where the agent is ready to formulate a completion, mxNLU has already predicted a frame for the user's utterance (even though it is still incomplete). We then consider only the set of known utterances that correspond to that frame as possible sources of completions. As a simple distance metric, we compute the word error rate (WER) between the current partial hypothesis for the user's utterance and a prefix of each of these known utterances. In our prototype, these prefixes have the same length as the current partial ASR hypothesis. We then select the utterance whose prefix has the lowest WER against the current partial ASR hypothesis. As a final step, we look in the prefix of our selected utterance for the last occurrence of the last word in the partial ASR, and if such a word is found, we take the remainder of the utterance as the agent's completion. Considering only the set of utterances that correspond to the frame predicted by mxNLU makes it likely that the completion will have the appropriate meaning. Since the completion is a suffix of a transcript of a previous user utterance, and this suffix follows the last word uttered by the user, it is likely to form a fluent completion of the user's partial utterance.

We applied this process to 449 user utterances.[5] For 356 of these utterances (79.2%), MAXF becomes true at some point during the utterance. Of these 356, our prototype is able to generate a (non-empty) completion for 190 utterances (53.3% of the MAXF utterances), and fails to generate a completion for 166 utterances (46.6% of the MAXF utterances). Thus, overall, our prototype is able both to identify a moment of maximal understanding and also to generate an utterance completion for 42.3% (190 of 449) of user utterances.

We now discuss the completions that our prototype generates. We provide some representative examples of its completions in Tables 2 and 3. These tables divide the generated completions into cases when the NLU output frame at KMAXF is perfectly correct vs. those in which it is either partially or completely incorrect.

The case of perfectly correct NLU output at KMAXF, which is exemplified in Table 2, occurs in 103 (54.2%) of the 190 cases for which our completion process is able to produce a completion of the user's utterance. These cases reflect the ideal situation in which the MAXF classifier allows the agent to detect that it has reached a moment of maximal understanding, and further, the NLU has in fact correctly understood the utterance.

The alternative case, in which the NLU output at KMAXF is either partially or completely incorrect, is exemplified in Table 3. This case includes 87 (45.8%) of the 190 utterances in which our implementation is able to produce a completion of the user's utterance. In these cases, the generated completions exhibit a number of weaknesses, including incorrect predictions about the user's meaning (for example, predicting *a vehicle* when the user was going to say *a well for the village*) as well as disfluent completions caused by mistakes in the ASR prefix; see the examples marked *Full Transcript*, in which an incorrect ASR prefix can affect the system's completion.

Together, these examples highlight both strengths as well as limitations in our prototype implementation of utterance completions. Table 2 includes a number of cases in which this system is able to complete the user's utterance in a manner that seems fluent and consistent with the user's communicative intention. On the other hand, together with the examples of Table 3, in which MAXF is true but the NLU output is not completely accurate, these results suggest that a more sophisticated decision-making process would be beneficial in deciding to complete a user's utterance. In particular, MAXF detects that the system's understanding is unlikely to improve by waiting, but it does not guarantee that the system's understanding is actually correct. (The NLU output might be incorrect, and likely to remain incorrect.) Thus, it could be useful to train the system to identify situations in which it is sure that its understanding is in fact highly accurate, before risking attempting a com-

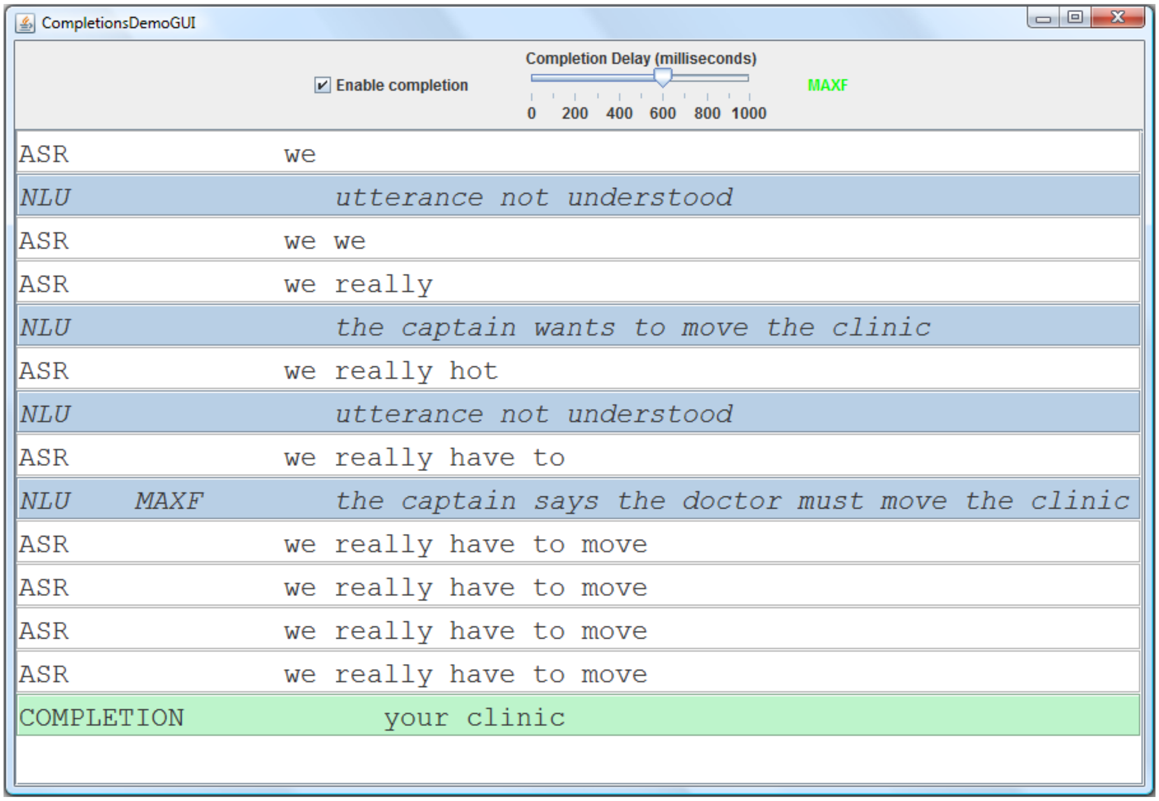
[5] This was performed in such a way that the to-be-completed utterances were never included in the training process for either mxNLU or MAXF.

Partial ASR result at KMAXF	Predicted completion	Actual user completion	F-Score at KMAXF
yes we'd like to	move you out of here	move your location	1
we need	to move the clinic soon	to move your clinic	1
we can provide you	with supplies you need	with supplies	1
we can have	locals to move you	locals help with the move	1
we can provide transportation	to move the patient there	(empty)	1
we should	move this facility	move the clinic right now	1
we will bring trucks then move it to	the trucks and move them to the new safe clinic	*Full transcript:* we will bring trucks in move your patients and your supplies	1
it is not	safe here we can't protect you	safe here	1
there are supplies	where we are going	there	1

Table 2: Some examples of generated completions when the predicted NLU frame is correct.

Partial ASR result at KMAXF	Predicted completion	Actual user completion	F-Score at KMAXF
we have to	move to a safer place	move the clinic because it is not safe here	0.33
we can provide	a vehicle	a well for the village	0.71
we have	supplies available	to move this clinic	0.38
so we	are trying to help these people also	will move the clinic downtown away from the insurgent activity	0.1
if we can	give you transportation	*Full transcript:* we can not protect you here	0.4
yes i	see	agree	0
doctor would the	market isn't a safe place for a clinic v	*Full transcript:* doctor what do you think	0
we can i	can help you with supplies	*Full transcript:* we cannot protect you	0.3
would like that	you move your clinic	*Full transcript:* we'd like to talk to you both about the clinic	0

Table 3: Some examples of generated completions when the predicted NLU frame is incorrect, or partially incorrect.



☑ Enable completion

```
ASR              we
NLU                 utterance not understood
ASR              we we
ASR              we really
NLU                 the captain wants to move the clinic
ASR              we really hot
NLU                 utterance not understood
ASR              we really have to
NLU     MAXF        the captain says the doctor must move the clinic
ASR              we really have to move
ASR              we really have to move
ASR              we really have to move
ASR              we really have to move
COMPLETION             your clinic
```

Figure 13: A graphical user interface for interactive utterance completion.

pletion of the user's utterance. We are pursuing this extension, and possibilities for combining the MAXF judgment with a decision about NLU correctness, in ongoing work.

Finally, it is worth emphasizing that even when an agent has the ability to generate a completion, clearly a number of broader strategic considerations could be relevant in deciding whether to do so. Determining a broader dialogue policy that results in natural behavior with respect to the frequency of completions for different types of agents is a topic under current investigation.

6.1 Toward completion of live user utterances

The above analysis of our prototype utterance completion capability is based on off-line, batch mode completion of pre-captured user utterances. In more recent work, we have developed a run-time demonstration that allows the utterance completion capability to be explored interactively (Sagae et al. 2010). To help visualize the agent's decision to complete an utterance, we have developed a simple GUI, depicted in Figure 13, that highlights important state changes within our mxNLU and MAXF incremental NLU models, and also provides some control over the timing of utterance completions.

The figure shows an example in which the character judges that the user's partial utterance *we really have to move* could be completed by *your clinic*.[6] The agent's incremental reasoning each 200ms is depicted in the GUI, in chronological order, from the top of the GUI to the bottom. First, the GUI shows the evolution of incremental ASR results during the user's speech; see the lines marked 'ASR', with a white background. The GUI also includes an entry each time there is a change in the semantic frame predicted by mxNLU; see the lines marked 'NLU', highlighted in blue. For compactness, the GUI presents an English language gloss of the predicted semantic frame, such as `the captain wants to move the clinic`, rather than the full AVM structure. The gloss `utterance not understood` is used to represent the "garbage" frame.

Finally, the GUI also indicates those moments when the MAXF classifier judges that MAXF is true. In this example, MAXF becomes true when the ASR output is *we really have to*. At this point, an attempt could be made to generate an utterance completion. However, it is very likely that this would result in the character barging in and interrupting the user's speech, or talking simultaneously with the user. While this aggressive behavior is interesting, we have found it undesirable in most cases. As we have not yet developed a refined model of the decision to interrupt ongoing user speech with utterance completions, we have instead provided an interactive control which requires a threshold amount of silence, after MAXF has become true, before an utterance completion will be initiated by the character. This control, seen at the top of the figure and marked "Completion Delay (milliseconds)", is set here to 600 milliseconds. In this example, the user does indeed pause for 600 milliseconds after saying *we really have to move*. (This can be seen by the repeated partial ASR results in the figure.) After this threshold period has elapsed, the character initiates its completion and utters *your clinic*.

7. Discussion and future work

In this section, we present some discussion and context for the techniques we have presented above. We address issues of generality, alternative approaches, integration into dialogue system design, limitations, and planned future work.

7.1 Generality and training requirements

Because we use data-driven techniques, which draw on domain-specific utterance data to train a domain-specific NLU capability, we can expect that our techniques will transfer to some extent into new dialogue domains where similar utterance data is available or can be acquired. Unfortunately, due to the numerous and varied empirical details that may affect NLU performance, it is difficult to specify in advance, with precision, the amount of data that is required in order for acceptable performance to be achieved in a new domain with our techniques.

In our SASO-EN domain, we achieve a relatively high F-score (0.76) for NLU, given the complexity of the task and the high word error rate in ASR (0.54). This is accomplished using a training set of about 3,500 annotated user utterances. However, even in situations where collecting and annotating a training corpus of this size may be impractical, it is still possible for the multiclass classification approach to produce high quality NLU results. When mxNLU is trained using a re-

[6] Note that the text *your clinic* is drawn from previous user utterances. For our prototype completion capability, we have not taken the step of replacing user pronouns (*your clinic*) with corresponding character pronouns (*my clinic*).

duced training set of 1,000 utterances, its F-score is about 0.7, with small gains resulting from the addition of more training material.

Of course, the performance of mxNLU or a similar module based on multiclass classification depends on several factors that are specific to each dialogue system. These include ASR performance, the size of the domain-specific vocabulary that users will employ, the number of different concepts and semantic frames the system is expected to understand, and the target user population. In general, it is important that there be reliable features in the ASR output, similar to the features displayed in Table 1, that the NLU can (learn to) use to identify the correct interpretation. The extent to which such features exist for a domain will depend on many empirical details.

See also Heintze et al. (2010) for a useful comparative analysis of incremental NLU performance in different domains.

7.2 Alternative approaches to incremental understanding

The approach to incremental understanding we have presented in this paper is fundamentally predictive. As the user is speaking, the NLU module attempts at each moment to predict the *complete* semantic frame that will be the correct analysis of their *entire* utterance. Alternative approaches to incremental NLU are possible. For example, the incremental NLU might not be predictive, and instead try to identify only those frame elements that have been directly expressed by the user in their speech so far. Or, as a variant on this approach, the incremental NLU could also try to identify which specific words or expressions in the user's partial utterance have expressed which frame elements (Heintze et al. 2010).

These alternative approaches solve different problems than our predictive model solves, and we view them as providing complementary information to incremental dialogue systems. As one example of how this information might be combined, in a situation where a confident prediction could not be made of the user's complete meaning, a system could nevertheless begin to reason about and respond to those aspects of the user's meaning about which it is confident.

7.3 System architecture

To date, implemented dialogue systems have generally attempted to understand user utterances only after the conclusion of the user's speech. The kinds of modular architectures that have been used to implement these systems have often assumed that a sequential pipeline of processing steps follows the conclusion of the user's utterance.[7] It may not be straightforward to substitute incremental modules, such as our mxNLU and MAXF classifiers, into these pipelines, unless the other modules are also modified to respond appropriately to the incremental processing results; see Schlangen and Skantze (2009) for an approach to this general problem. For example, a dialogue manager might be modified to delay the initiation of any response to a user utterance until MAXF is true. Such a policy could prove useful in some systems, but it is undoubtedly a simplistic approach to a complex problem. Our prototype implementation of utterance completion, which initiates completion only once MAXF is true, was designed to help explore the potential value of these incremental processing models. However, there remain a number of questions about how system architectures should be adjusted to take maximum advantage of them.

[7] For example: User speech → ASR → NLU → DM → NLG → TTS

7.4 Limitations of the utterance completion method

The utterance completion capability presented in Section 6 has some limitations. Because each completion text used by the character is extracted verbatim from the set of existing utterance transcripts in our NLU training corpus, the set of possible completion texts is finite. Although the set of possible completion texts is not so small (as there are thousands of utterance transcripts to draw from), it could nevertheless be argued that our prototype supports little or no linguistic creativity by the character as it performs the completions. A more general capability would productively generate the completion text, perhaps using a semantic analysis of the user's partial utterance (identifying the subset of frame elements the user has directly expressed in their speech) and a productive grammar (to allow the character to express linguistically the predicted frame elements which the user has not yet expressed). It would be interesting to explore the development of such a refined completion capability in future work.

We should note however that our current prototype does support linguistic creativity on the user's side, as our mxNLU and MAXF models, which underlie the completion capability, do accept arbitrary user speech. Indeed, the analysis and results for our completion capability, presented in Section 6, were confined to completions of new user utterances which were not drawn from the training set of either the mxNLU or MAXF models.

Finally, as mentioned in Section 6, the decision to complete a user's utterance is in reality a complex turn-taking decision, which should be based on a range of additional factors not taken into account in our prototype. We are just beginning to explore these issues in our ongoing work.

7.5 Backchannels

We are currently exploring the use of incremental processing for providing backchannels, using both verbal and non-verbal channels. The incremental NLU and speech results are used by the dialogue manager, which attempts to perform reference resolution against the internal domain model, and extracts information about whether or not the agent understands the utterance, agrees with it, and the agent's emotions (Gratch and Marsella (2004)) toward the described state or action, or the participants. This information, along with the words, semantic frame, and MAXF information are provided to the verbal and non-verbal generators (DeVault et al. (2008), Lee and Marsella (2006)), which decide whether to produce a backchannel and the form (e.g. repeating words, saying "uh huh", "yes" or "no", head nods or shakes, and facial expressions).

7.6 Evaluation

In this paper we have presented several kinds of evaluation for our techniques. In Section 3.4, we analyzed the performance of mxNLU on complete ASR output, in terms of its recall and precision of semantic frame elements. In Section 4, we presented a refined performance analysis for several alternative approaches to training mxNLU to predict complete semantic frames from only partial ASR output. In Section 5.3, we presented an analysis of our MAXF classifier's performance, in terms of precision and recall of points of maximal understanding, as well as the potential time saved and penalty in F-score. Finally, in Section 6, we presented statistics on the frequency with which our completion prototype is able to generate completions for new user utterances, as well as the frequency with which mxNLU's prediction at the moment of completion was correct vs. incorrect.

What we have not evaluated to date, however, is what progress we have made toward the higher-level goal of improving the responsiveness and naturalness of interacting with virtual human characters. While this remains our long-term motivation, there remains more work to be done before it will be possible to demonstrate the usefulness in live dialogue sessions of an utterance completion capability, or other overlap behaviors that draw on our techniques. In particular, we expect that this will require that virtual humans engage in more detailed decision-making about when and how to initiate an overlapping response based on incremental processing results. Beyond utterance completions and back-channels, our virtual humans have a number of additional options for overlapping responses, including eye gaze, facial displays, head nods or head shakes, and other non-verbal behavior such as gesture. As we explore these options, we also expect that more detailed consideration of the fine-grained timing of overlapping responses than we have presented here will likely prove essential to achieving natural behavior. Here we have presented several fundamental techniques for the incremental interpretation and prediction of utterance meaning, which we believe are likely to prove valuable in implementing this more complex decision-making. We are eagerly exploring the issues involved in translating these techniques into live capabilities in our ongoing work.

8. Conclusion

We have presented a framework for interpretation of partial ASR hypotheses of user utterances, and high-precision identification of points within user utterances where the system has reached maximal understanding of the intended meaning. Our initial implementation of an utterance completion ability for a virtual human serves to illustrate the capabilities of this framework, but only scratches the surface of the new range of dialogue behaviors and strategies it allows.

Acknowledgments

The work described here has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred. We would also like to thank Anton Leuski for facilitating the use of incremental speech results, and the ICT dialogue group and David Schlangen for helpful discussions, and our anonymous reviewers for helpful suggestions on the presentation of the paper.

References

Gregory Aist, James Allen, Ellen Campana, Carlos Gomez Gallo, Scott Stoness, Mary Swift, and Michael K. Tanenhaus. Incremental dialogue system faster than and preferred to its nonincremental counterpart. In *Proc. of the 29th Annual Conference of the Cognitive Science Society*, 2007.

Jens Allwood. An activity based approach to pragmatics. Technical Report (GPTL) 75, Gothenburg Papers in Theoretical Linguistics, University of Göteborg, 1995.

Jens Allwood, Joakim Nivre, and Elisabeth Ahlsen. On the semantics and pragmatics of linguistic feedback. *Journal of Semantics*, 9, 1992.

Adam L. Berger, Stephen D. Della Pietra, and Vincent J. D. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth Churchill, editors. *Embodied Conversational Agents*. MIT Press, Cambridge, MA, 2000.

Herbert H. Clark. *Arenas of Language Use*. University of Chicago Press, 1992.

Herbert H. Clark and Edward F. Schaefer. Collaborating on contributions to conversation. *Language and Cognitive Processes*, 2:1–23, 1987.

Herbert H. Clark and Deanna Wilkes-Gibbs. Referring as a collaborative process. *Cognition*, 22: 1–39, 1986. Also appears as Chapter 4 in Clark (1992).

David DeVault, David Traum, and Ron Artstein. Making grammar-based generation easier to deploy in dialogue systems. In *Fifth INLG Conference*, 2008.

David DeVault, Kenji Sagae, and David Traum. Can I finish? Learning when to respond to incremental interpretation results in interactive dialogue. In *The 10th Annual SIGDIAL Meeting on Discourse and Dialogue (SIGDIAL 2009)*, 2009.

Kotaro Funakoshi, Mikio Nakano, Kazuki Kobayashi, Takanori Komatsu, and Seiji Yamada. Nonhumanlike spoken dialogue: A design perspective. In *Proceedings of the SIGDIAL 2010 Conference*, pages 176–184, Tokyo, Japan, September 2010.

Charles Goodwin. The interactive construction of a sentence in natural conversation. In G. Psathas, editor, *Everyday Language: Studies in Ethnomethodology*, pages 97–121. Ervington Press, New York, 1979.

Jonathan Gratch and Stacy Marsella. A domain-independent framework for modeling emotion. *Journal of Cognitive Systems Research*, 2004.

Brian Hansen, David Novick, and Stephen Sutton. Prevention and repair of breakdowns in a simple task domain. In *Proceedings of the* AAAI-96 Workshop on Detecting, Repairing, and Preventing Human-Machine Miscommunication, pages 5–12, 1996.

Arno Hartholt, Thomas Russ, David Traum, Eduard Hovy, and Susan Robinson. A common ground for virtual humans: Using an ontology in a natural language oriented virtual human architecture. In European Language Resources Association (ELRA), editor, *Proc. of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008.

Silvan Heintze, Timo Baumann, and David Schlangen. Comparing local and sequential models for statistical incremental natural language understanding. In *The 11th Annual Meeting of the Special Interest Group in Discourse and Dialogue (SIGDIAL 2010)*, 2010.

David Huggins-Daines, Mohit Kumar, Arthur Chan, Alan W. Black, Mosur Ravishankar, and Alex I. Rudnicky. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *Proceedings of ICASSP*, 2006.

Geert-Jan M. Kruijff, Pierre Lison, Trevor Benjamin, Henrik Jacobsson, and Nick Hawes. Incremental, multi-level processing for comprehending situated dialogue in human-robot interaction. In *Language and Robots: Proc. from the Symposium (LangRo'2007)*. University of Aveiro, 12 2007.

Jina Lee and Stacy Marsella. Nonverbal behavior generator for embodied conversational agents. In Jonathan Gratch, Michael Young, Ruth Aylett, Daniel Ballin, and Patrick Olivier, editors, *IVA*, pages 243–255. Springer, 2006. ISBN 3-540-37593-7.

Yukiko I. Nakano, Gabe Reinstein, Tom Stocky, and Justine Cassell. Towards a model of face-to-face grounding. In *ACL*, pages 553–561, 2003.

Bryan Pellom. Sonic: The university of colorado continuous speech recognizer. In *University of Colorado, tech report #TR-CSLR-2001-01*, Boulder, Colorado, 2001.

Massimo Poesio and Hannes Rieser. Completions, coordination, and alignment in dialogue. *Dialogue and Discourse*, 1:1–89, 2010. URL `http://elanguage.net/journals/index.php/dad/article/view/91`.

Jeff Rickel and W. Lewis Johnson. Virtual humans for team training in virtual reality. In *Proceedings of the Ninth International Conference on Artificial Intelligence in Education*, pages 578–585. IOS Press, 1999.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50:696–735, 1974.

M. David Sadek. Dialogue acts are rational plans. In *Proceedings of the ESCA/ETR workshop on multi-modal dialogue*, 1991.

Kenji Sagae, Gwen Christian, David DeVault, and David R. Traum. Towards natural language understanding of partial speech recognition results in dialogue systems. In Short Paper Proceedings of NAACL HLT, 2009.

Kenji Sagae, David DeVault, and David R. Traum. Interpretation of partial utterances in virtual human dialogue systems. In *The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2010 Demonstration)*, 2010.

David Schlangen and Gabriel Skantze. A general, abstract model of incremental dialogue processing. In *Proc. of the 12th Conference of the European Chapter of the ACL*, 2009.

David Schlangen, Timo Baumann, and Michaela Atterer. Incremental reference resolution: The task, metrics for evaluation, and a bayesian filtering model that is sensitive to disfluencies. In *The 10th Annual SIGDIAL Meeting on Discourse and Dialogue (SIGDIAL 2009)*, 2009.

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

Kristina Skuplik. Satzkooperationen. Definition und empirische untersuchung. Technical Report 1999/03 of SFB 360, Bielefeld, 1999.

David Traum. Semantics and pragmatics of questions and answers for dialogue agents. In *Proc. of the International Workshop on Computational Semantics*, pages 380–394, January 2003.

David Traum, Stacy Marsella, Jonathan Gratch, Jina Lee, and Arno Hartholt. Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. In *Proc. of Intelligent Virtual Agents Conference IVA-2008*, 2008.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

Victor H. Yngve. On getting a word in edgewise. In *Papers from the Sixth Regional Meeting*, pages 567–78. Chicago Linguistic Society, 1970.