Making Grammar-Based Generation Easier to Deploy in Dialogue Systems

David DeVault and **David Traum** and **Ron Artstein**
USC Institute for Creative Technologies
13274 Fiji Way
Marina del Rey, CA 90292
{devault,traum,artstein}@ict.usc.edu

Abstract

We present a development pipeline and associated algorithms designed to make grammar-based generation easier to deploy in implemented dialogue systems. Our approach realizes a practical trade-off between the capabilities of a system's generation component and the authoring and maintenance burdens imposed on the generation content author for a deployed system. To evaluate our approach, we performed a human rating study with system builders who work on a common large-scale spoken dialogue system. Our results demonstrate the viability of our approach and illustrate authoring/performance trade-offs between hand-authored text, our grammar-based approach, and a competing shallow statistical NLG technique.

1 Introduction

This paper gives an overview of a new example-based generation technique that is designed to make grammar-based generation easier to deploy in dialogue systems. Dialogue systems present several specific requirements for a practical generation component. First, the generator needs to be fast enough to support real-time interaction with a human user. Second, the generator must provide adequate coverage for the meanings the dialogue system needs to express. What counts as "adequate" can vary between systems, since the high-level purpose of a dialogue system can affect priorities regarding output fluency, fidelity to the requested meaning, variety of alternative outputs, and tolerance for generation failures. Third, developing the necessary resources for the generation component should be relatively straightforward in terms of time and expertise required. This is especially important since dialogue systems are complex systems with significant development costs. Finally, it should be relatively easy for the dialogue manager to formulate a generation request in the format required by the generator.

Together, these requirements can reduce the attractiveness of grammar-based generation when compared to simpler template-based or canned text output solutions. In terms of speed, off-the-shelf, wide-coverage grammar-based realizers such as FUF/SURGE (Elhadad, 1991) can be too slow for real-time interaction (Callaway, 2003).

In terms of adequacy of coverage, in principle, grammar-based generation offers significant advantages over template-based or canned text output by providing productive coverage and greater variety. However, realizing these advantages can require significant development costs. Specifying the necessary connections between lexico-syntactic resources and the flat, domain-specific semantic representations that are typically available in implemented systems is a subtle, labor-intensive, and knowledge-intensive process for which attractive methodologies do not yet exist (Reiter et al., 2003).

One strategy is to hand-build an application-specific grammar. However, in our experience, this process requires a painstaking, time-consuming effort by a developer who has detailed linguistic knowledge as well as detailed domain knowledge, and the resulting coverage is inevitably limited.

Wide-coverage generators that aim for applicabil-

Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue, pages 198–207,
Columbus, June 2008.

ity across application domains (White et al., 2007; Zhong and Stent, 2005; Langkilde-Geary, 2002; Langkilde and Knight, 1998; Elhadad, 1991) provide a grammar (or language model) for free. However, it is harder to tailor output to the desired wording and style for a specific dialogue system, and these generators demand a specific input format that is otherwise foreign to an existing dialogue system. Unfortunately, in our experience, the development burden of implementing the translation between the system's available meaning representations and the generator's required input format is quite substantial. Indeed, implementing the translation might require as much effort as would be required to build a simple custom generator; cf. (Callaway, 2003; Busemann and Horacek, 1998). This development cost is exacerbated when a dialogue system's native meaning representation scheme is under revision.

In this paper, we survey a new example-based approach (DeVault et al., 2008) that we have developed in order to mitigate these difficulties, so that grammar-based generation can be deployed more widely in implemented dialogue systems. Our development pipeline requires a system developer to create a set of training examples which directly connect desired output texts to available application semantic forms. This is achieved through a streamlined authoring task that does not require detailed linguistic knowledge. Our approach then processes these training examples to automatically construct all the resources needed for a fast, high-quality, run-time grammar-based generation component. We evaluate this approach using a pre-existing spoken dialogue system. Our results demonstrate the viability of the approach and illustrate authoring/performance trade-offs between hand-authored text, our grammar-based approach, and a competing shallow statistical NLG technique.

2 Background and Motivation

The generation approach set out in this paper has been developed in the context of a research program aimed at creating interactive virtual humans for social training purposes (Swartout et al., 2006). Virtual humans are embodied conversational agents that play the role of people in simulations or games. They interact with human users and other virtual humans using spoken language and non-verbal behavior such as eye gaze, gesture, and facial displays.

Figure 1: Doctor Perez.

The case study we present here is the generation of output utterances for a particular virtual human, Doctor Perez (see Figure 1), who is designed to teach negotiation skills in a multi-modal, multi-party, non-team dialogue setting (Traum et al., 2005; Traum et al., 2008). The human trainee who talks to the doctor plays the role of a U.S. Army captain named Captain Kirk. We summarize Doctor Perez's generation requirements as follows.

In order to support compelling real-time conversation and effective training, the generator must be able to identify an utterance for Doctor Perez to use within approximately 200ms on modern hardware.

Doctor Perez has a relatively rich internal mental state including beliefs, goals, plans, and emotions. As Doctor Perez attempts to achieve his conversational goals, his utterances need to take a variety of syntactic forms, including simple declarative sentences, various modal constructions relating to hypothetical actions or plans, yes/no and wh-questions, and abbreviated dialogue forms such as elliptical clarification and repair requests, grounding, and turn-taking utterances. Doctor Perez currently uses about 200 distinct output utterances in the course of his dialogues.

Doctor Perez is designed to simulate a non-native English speaker, so highly fluent output is not a necessity; indeed, a small degree of disfluency is even desirable in order to increase the realism of talking to a non-native speaker.

Finally, in reasoning about user utterances, dialogue management, and generation, Doctor Perez

$$
\left[\begin{array}{ll}
\text{addressee} & \text{captain-kirk} \\
\text{dialogue-act} & \left[\begin{array}{ll} \text{addressee} & \text{captain-kirk} \\ \text{type} & \text{assign-turn} \\ \text{actor} & \text{doctor-perez} \end{array}\right] \\
\text{speech-act} & \left[\begin{array}{ll} \text{actor} & \text{doctor-perez} \\ \text{addressee} & \text{captain-kirk} \\ \text{action} & \text{assert} \\ \text{content} & \left[\begin{array}{ll} \text{type} & \text{state} \\ \text{polarity} & \text{negative} \\ \text{time} & \text{present} \\ \text{attribute} & \text{resourceAttribute} \\ \text{value} & \text{medical-supplies} \\ \text{object-id} & \text{market} \end{array}\right] \end{array}\right]
\end{array}\right]
$$

(a) Attribute-value matrix

addressee	captain-kirk
dialogue-act.addressee	captain-kirk
dialogue-act.type	assign-turn
dialogue-act.actor	doctor-perez
speech-act.actor	doctor-perez
speech-act.addressee	captain-kirk
speech-act.action	assert
speech-act.content.type	state
speech-act.content.polarity	negative
speech-act.content.time	present
speech-act.content.attribute	resourceAttribute
speech-act.content.value	medical-supplies
speech-act.content.object-id	market

(b) Corresponding frame

Figure 2: An example of Doctor Perez's representations for utterance semantics: Doctor Perez tells the captain that there are no medical supplies at the market.

exploits an existing semantic representation scheme that has been utilized in a family of virtual humans. This scheme uses an attribute-value matrix (AVM) representation to describe an utterance as a set of core speech acts and other dialogue acts. Speech acts generally have semantic contents that describe propositions and questions about states and actions in the domain, as well as other features such as polarity and modality. See (Traum, 2003) for some more details and examples of this representation. For ease of interprocess communication, and certain kinds of statistical processing, this AVM structure is linearized so that each non-recursive terminal value is paired with a path from the root to the final attribute. Thus, the AVM in Figure 2(a) is represented as the "frame" in Figure 2(b).

Because the internal representations that make up Doctor Perez's mental state are under constant development, the exact frames that are sent to the generation component change frequently as new reasoning capabilities are added and existing capabilities are reorganized. Additionally, while only hundreds of frames currently arise in actual dialogues, the number of potential frames is orders of magnitude larger, and it is difficult to predict in advance which frames might occur.

In this setting, over a period of years, a number of different approaches to natural language generation have been implemented and tested, including hand-authored canned text, domain specific hand-built grammar-based generators (e.g., (Traum et al., 2003)), shallow statistical generation techniques, and the grammar-based approach presented in this paper. We now turn to the details of our approach.

3 Technical Approach

Our approach builds on recently developed techniques in statistical parsing, lexicalized syntax modeling, generation with lexicalized grammars, and search optimization to automatically construct all the resources needed for a high-quality run-time generation component.

The approach involves three primary steps: specification of training examples, grammar induction, and search optimization. In this section, we present the format that training examples take and then summarize the subsequent automatic processing steps. Due to space limitations, we omit the full details of these automatic processing steps, and refer the reader to (DeVault et al., 2008) for additional details.

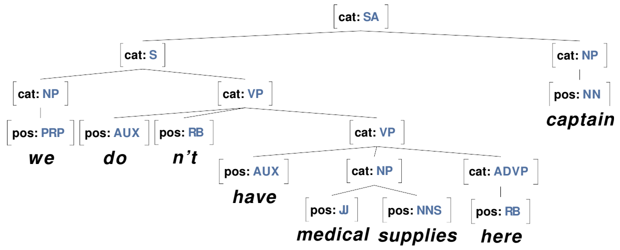
3.1 Specification of Training Examples

Each training example in our approach specifies a target output utterance (string), its syntax, and a set of links between substrings within the utterance and system semantic representations. Formally, a training example takes the form $(u, \mathrm{syntax}(u), \mathrm{semantics}(u))$. We will illustrate this format using the training example in Figure 3.

In this example, the generation content author

Utterance — we don't have medical supplies here captain

Syntax



Semantics

we do n't	speech-act.action = assert speech-act.content.polarity = negative
have	speech-act.content.attribute = resourceAttribute
medical supplies . .	speech-act.content.value = medical-supplies
here	speech-act.content.object-id = market
captain	addressee = captain-kirk dialogue-act.addressee = captain-kirk speech-act.addressee = captain-kirk

Figure 3: A generation training example for Doctor Perez.

suggests the output utterance u = *we don't have medical supplies here captain*. Each utterance u is accompanied by $\mathrm{syntax}(u)$, a syntactic analysis in Penn Treebank format (Marcus et al., 1994). In this example, the syntax is a hand-corrected version of the output of the Charniak parser (Charniak, 2001; Charniak, 2005) on this sentence; we discuss this hand correction in Section 4.

To represent the meaning of utterances, our approach assumes that the system provides some set $M = \{m_1, ..., m_j\}$ of semantic representations. The meaning of any individual utterance is then identified with some subset of M. For Doctor Perez, M comprises the 232 distinct key-value pairs that appear in the system's various generation frames. In this example, the utterance's meaning is captured by the 8 key-value pairs indicated in the figure.

Our approach requires the generation content author to link these 8 key-value pairs to contiguous surface expressions within the utterance. The technique is flexible about which surface expressions are chosen (e.g. they need not correspond to constituent boundaries); however, they do need to be compatible with the way the syntactic analysis tokenizes the utterance, as follows. Let $t(u) = \langle t_1, ..., t_n \rangle$ be the terminals in the syntactic analysis, in left-to-right order. Formally, $\mathrm{semantics}(u) = \{(s_1, M_1), ..., (s_k, M_k)\}$, where $t(u) = s_1 @ \cdots @ s_k$ (with @ denoting concatenation), and where $M_i \subseteq M$ for all $i \in 1..k$. In this example, the surface expression *we don't*, which tokenizes as $\langle \mathrm{we}, \mathrm{do}, \mathrm{n't} \rangle$, is connected to key-values that indicate a negative polarity assertion.

This training example format has two features that are crucial to our approach. First, the semantics of an utterance is specified *independently* of its syntax. This greatly reduces the amount of linguistic expertise a generation content author needs to have. It also allows making changes to the underlying syntax without having to re-author the semantic links.

Second, the assignment of semantic representations to surface expressions must span the *entire utterance*. No words or expressions can be viewed as "meaningless". This is essential because, otherwise, the semantically motivated search algorithm used in generation has no basis on which to include those particular expressions when it constructs its output utterance. Many systems, including Doctor Perez, lack some of the internal representations that would be necessary to specify semantics down to the lexical level. An important feature of our approach is that it allows an arbitrary semantic granularity to be employed, by mapping the representations available in the system to appropriate multi-word chunks.

3.2 Automatic Grammar Induction and Search Optimization

The first processing step is to induce a productive grammar from the training examples. We adopt the probabilistic tree-adjoining grammar (PTAG) formalism and grammar induction technique of (Chiang, 2003). We induce our grammar from training examples such as Figure 3 using heuristic rules to assign derivations to the examples, as in (Chiang, 2003). Once derivations have been assigned, subtrees within the training example syntax are incrementally detached. This process yields the reusable linguistic resources in the grammar, as well as the statistical model needed to compute operation probabilities when the grammar is later used in generation. Figure 5 in the Appendix illustrates this process by presenting the linguistic resources inferred from the training example of Figure 3.

Our approach uses this induced grammar to treat generation as a search problem: given a desired semantic representation $M' \subseteq M$, use the grammar to incrementally construct an output utterance u that expresses M'. We treat generation as anytime search by accruing multiple goal states up until a specified timeout (200ms for Doctor Perez) and returning a list of alternative outputs ranked by their derivation probabilities.

The search space created by a grammar induced in this way is too large to be searched exhaustively in most applications. The second step of automated processing, then, uses the training examples to learn an effective search policy so that good output sentences can be found in a reasonable time frame. The solution we have developed employs a beam search strategy that uses weighted features to rank alternative grammatical expansions at each step. Our algorithm for selecting features and weights is based on the search optimization algorithm of (Daumé and Marcu, 2005), which decides to update feature weights when mistakes are made during search on training examples. We use the boosting approach of (Collins and Koo, 2005) to perform feature selection and identify good weight values.

4 Empirical Evaluation

In the introduction, we identified run-time speed, adequacy of coverage, authoring burdens, and NLG request specification as important factors in the selection of a technology for a dialogue system's NLG component. In this section, we evaluate our technique along these four dimensions.

Hand-authored utterances. We collected a sample of 220 instances of frames that Doctor Perez's dialogue manager had requested of the generation component in previous dialogues with users. Some frames occurred more than once in this sample.

Each frame was associated with a single hand-authored utterance. Some of these utterances arose in human role plays for Doctor Perez; some were written by a script writer; others were authored by system builders to provide coverage for specific frames. All were reviewed by a system builder for appropriateness to the corresponding frame.

Training. We used these 220 (frame, utterance) examples to evaluate both our approach and a shallow statistical method called *sentence retriever* (discussed below). We randomly split the examples into 198 training and 22 test examples; we used the same train/test split for our approach and sentence retriever.

To train our approach, we constructed training examples in the format specified in Section 3.1. Syntax posed an interesting problem, because the Charniak parser frequently produces erroneous syntactic analyses for utterances in Doctor Perez's domain, but it was not obvious how detrimental these errors would be to overall generated output. We therefore constructed two alternative sets of training examples – one where the syntax of each utterance was the uncorrected output of the Charniak parser, and another where the parser output was corrected by hand (the syntax in Figure 3 above is the corrected version). Hand correction of parser output requires considerable linguistic expertise, so uncorrected output represents a substantial reduction in authoring burden. The connections between surface expressions and frame key-value pairs were identical in both uncorrected and corrected training sets, since they are independent of the syntax. For each training set, we trained our generator on the 198 training examples. We then generated a single (highest-ranked) utterance for each example in both the test and training sets. The generator sometimes failed to find a successful utterance within the 200ms timeout; the success rate of our generator was 95% for training examples and 80% for test examples. The successful utterances were rated by our judges.

Sentence retriever is based on the cross-language information retrieval techniques described in (Leuski et al., 2006), and is currently in use for Doctor Perez's NLG problem. Sentence retriever does not exploit any hierarchical syntactic analysis of utterances. Instead, sentence retriever views NLG as an information retrieval task in which a set of training utterances are the "documents" to be retrieved, and the frame to be expressed is the query. At run-time, the algorithm functions essentially as a classifier: it uses a relative entropy metric to select the highest ranking training utterance for the frame that Doctor Perez wishes to express. This approach has been used because it is to some extent robust against changes in internal semantic representations, and against minor deficiencies in the training corpus, but as with a canned text approach, it requires each utterance to be hand-authored before it can be used in dialogue. We trained sentence retriever on the 198 training examples, and used it to generate a single (highest-ranked) utterance for each example in both the test and training sets. Sentence retriever's success rate was 96% for training examples and 90% for test examples. The successful utterances were rated by our judges.

Figure 7 in the Appendix illustrates the alternative utterances that were produced for a frame present in the test data but not in the training data.

Run-time speed. Both our approach and sentence retriever run within the available 200ms window.

Adequacy of Coverage. To assess output quality, we conducted a study in which 5 human judges gave overall quality ratings for various utterances Doctor Perez might use to express specific semantic frames. In total, judges rated 494 different utterances which were produced in several conditions: hand-authored (for the relevant frame), generated by our approach, and sentence retriever.

We asked our 5 judges to rate each of the 494 utterances, in relation to the specific frame for which it was produced, on a single 1 ("very bad") to 5 ("very good") scale. Since ratings need to incorporate accuracy with respect to the frame, our judges had to be able to read the raw system semantic representations. This meant we could only use judges who were deeply familiar with the dialogue system; however, the main developer of the new generation algorithms (the first author) did not participate as a judge. Judges were blind to the conditions under which utterances were produced. The judges rated the utterances using a custom-built application which presented a single frame together with 1 to 6 candidate utterances for that frame. The rating interface is shown in Figure 6 in the Appendix. The order of candidate utterances for each frame was randomized, and the order in which frames appeared was randomized for each judge.

The judges were instructed to incorporate both fluency and accuracy with respect to the frame into a single overall rating for each utterance. While it is possible to have human judges rate fluency and accuracy independently, ratings of fluency alone are not particularly helpful in evaluating Doctor Perez's generation component, since for Doctor Perez, a certain degree of disfluency can contribute to believability (as noted in Section 2). We therefore asked judges to make an overall assessment of output quality for the Doctor Perez character.

The judges achieved a reliability of $\alpha = 0.708$ (Krippendorff, 1980); this value shows that agreement is well above chance, and allows for tentative conclusions. Agreement between subsets of judges ranged from $\alpha = 0.802$ for the most concordant pair of judges to $\alpha = 0.593$ for the most discordant pair. We also performed an ANOVA comparing three conditions (generated, retrieved and hand-authored utterances) across the five judges; we found significant main effects of condition ($F(2, 3107) = 55, p < 0.001$) and judge ($F(4, 3107) = 17, p < 0.001$), but no significant interaction ($F(8, 3107) = 0.55, p > 0.8$). We therefore conclude that the individual differences among the judges do not affect the comparison of utterances across the different conditions, so we will report the rest of the evaluation on the mean ratings per utterance.

Due to the large number of factors and the differences in the number of utterances corresponding to each condition, we ran a small number of planned comparisons. The distribution of ratings across utterances is not normal; to validate our results we accompanied each t-test by a non-parametric Wilcoxon rank sum test, and significance always fell in the same general range. We found a significant difference between generated

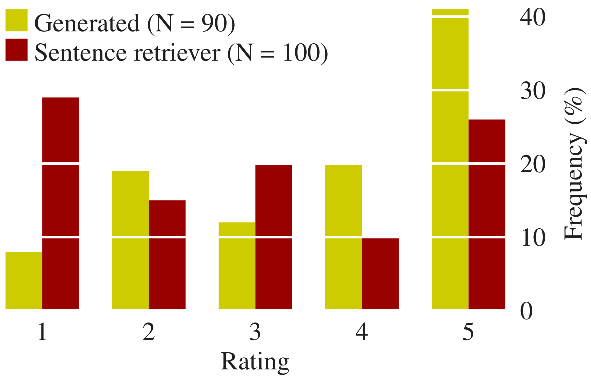


Figure 4: Observed ratings of generated (uncorrected syntax) vs. retrieved sentences for test examples.

output for all examples, retrieved output for all examples, and hand-authored utterances ($F(2, 622) = 16, p < 0.001$); however, subsequent t-tests show that all of this difference is due to the fact that hand-authored utterances (mean rating 4.4) are better than retrieved ($t(376) = 3.7, p < 0.001$) and generated ($t(388) = 5.9, p < 0.001$) utterances, whereas the difference between generated (mean rating 3.8) and retrieved (mean rating 4.0) is non-significant ($t(385) = 1.6, p > 0.1$).

Figure 4 shows the observed rating frequencies of sentence retriever (mean 3.0) and our approach (mean 3.6) on the test examples. While this data does not show a significant difference, it suggests that retriever's selected sentences are most frequently either very bad or very good; this reflects the fact that the classification algorithm retrieves highly fluent hand-authored text which is sometimes semantically very incorrect. (Figure 7 in the Appendix provides such an example, in which a retrieved sentence has the wrong polarity.) The quality of our generated output, by comparison, appears more graded, with very good quality the most frequent outcome and lower qualities less frequent. In a system where there is a low tolerance for very bad quality output, generated output would likely be considered preferable to retrieved output.

In terms of generation failures, our approach had poorer coverage of test examples than sentence retriever (80% vs. 90%). Note however that in this study, our approach only delivered an output if it could completely cover the requested frame. In the future, we believe coverage could be improved, with perhaps some reduction in quality, by allowing outputs that only partially cover requested frames.

In terms of output variety, in this initial study our judges rated only the highest ranked output generated or retrieved for each frame. However, we observed that our generator frequently finds several alternative utterances of relatively high quality (see Figure 7); thus our approach offers another potential advantage in output variety.

Authoring burdens. Both canned text and sentence retriever require only frames and corresponding output sentences as input. In our approach, syntax and semantic links are additionally needed. We compared the use of corrected vs. uncorrected syntax in training. Surprisingly, we found no significant difference between generated output trained on corrected and uncorrected syntax ($t(29) = 0.056, p > 0.9$ on test items, $t(498) = -1.1, p > 0.2$ on all items). This is a substantial win in terms of reduced authoring burden for our approach.

If uncorrected syntax is used, the additional burden of our approach lies only in specifying the semantic links. For the 220 examples in this study, one system builder specified these links in about 6 hours. We present a detailed cost/benefit analysis of this effort in (DeVault et al., 2008).

NLG request specification. Both our approach and sentence retriever accept the dialogue manager's native semantic representation for NLG as input.

Summary. In exchange for a slightly increased authoring burden, our approach yields a generation component that generalizes to unseen test problems relatively gracefully, and does not suffer from the frequent very bad output or the necessity to author every utterance that comes with canned text or a competing statistical classification technique.

5 Conclusion and Future Work

In this paper we have presented an approach to specifying domain-specific, grammar-based generation by example. The method reduces the authoring burden associated with developing a grammar-based NLG component for an existing dialogue system. We have argued that the method delivers relatively high-quality, domain-specific output without requiring that content authors possess detailed linguistic knowledge. In future work, we will study the performance of our approach as the size of the training set grows, and assess what specific weaknesses or problematic disfluencies, if any, our human rating study identifies in output generated by our technique. Finally, we intend to evaluate the performance of our generation approach within the context of the complete, running Doctor Perez agent.

Acknowledgments

Thanks to Arno Hartholt, Susan Robinson, Thomas Russ, Chung-chieh Shan, and Matthew Stone. This work was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM), and the content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

Stephen Busemann and Helmut Horacek. 1998. A flexible shallow approach to text generation. In *Proceedings of INLG*, pages 238–247.

Charles B. Callaway. 2003. Evaluating coverage for large symbolic NLG grammars. *Proceedings of the International Joint Conferences on Artificial Intelligence*.

Eugene Charniak. 2001. Immediate-head parsing for language models. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 124–131, Morristown, NJ, USA. Association for Computational Linguistics.

Eugene Charniak. 2005. ftp://ftp.cs.brown.edu/pub/nlparser/parser05Aug16.tar.gz.

David Chiang. 2003. Statistical parsing with an automatically extracted tree adjoining grammar. In Rens Bod, Remko Scha, and Khalil Sima'an, editors, *Data Oriented Parsing*, pages 299–316. CSLI Publications, Stanford.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

Hal Daumé, III and Daniel Marcu. 2005. Learning as search optimization: approximate large margin methods for structured prediction. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 169–176, New York, NY, USA. ACM.

David DeVault, David Traum, and Ron Artstein. 2008. Practical grammar-based NLG from examples. In *Fifth International Natural Language Generation Conference (INLG)*.

Michael Elhadad. 1991. FUF: the universal unifier user manual version 5.0. Technical Report CUCS-038-91.

Klaus Krippendorff, 1980. *Content Analysis: An Introduction to Its Methodology*, chapter 12, pages 129–154. Sage, Beverly Hills, CA.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *COLING-ACL*, pages 704–710.

I. Langkilde-Geary. 2002. An empirical verification of coverage and correctness for a general-purpose sentence generator.

Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. 2006. Building effective question answering characters. In *The 7th SIGdial Workshop on Discourse and Dialogue*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

E. Reiter, S. Sripada, and R. Robertson. 2003. Acquiring correct knowledge for natural language generation. *Journal of Artificial Intelligence Research*, 18:491–516.

William Swartout, Jonathan Gratch, Randall W. Hill, Eduard Hovy, Stacy Marsella, Jeff Rickel, and David Traum. 2006. Toward virtual humans. *AI Mag.*, 27(2):96–108.

David Traum, Michael Fleischman, and Eduard Hovy. 2003. Nl generation for virtual humans in a complex social environment. In *Working Notes AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue*, March.

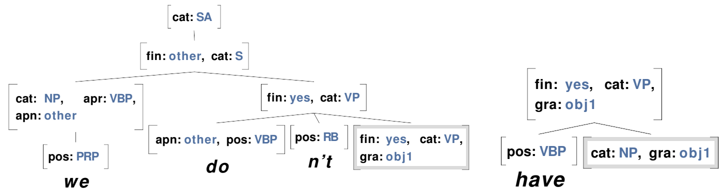
David Traum, William Swartout, Jonathan Gratch, Stacy Marsella, Patrick Kenny, Eduard Hovy, Shri Narayanan, Ed Fast, Bilyana Martinovski, Rahul Baghat, Susan Robinson, Andrew Marshall, Dagen Wang, Sudeep Gandhe, and Anton Leuski. 2005. Dealing with doctors: A virtual human for non-team interaction. In *SIGdial*.

D. R. Traum, W. Swartout, J Gratch, and S Marsella. 2008. A virtual human dialogue model for non-team interaction. In Laila Dybkjaer and Wolfgang Minker, editors, *Recent Trends in Discourse and Dialogue*. Springer.

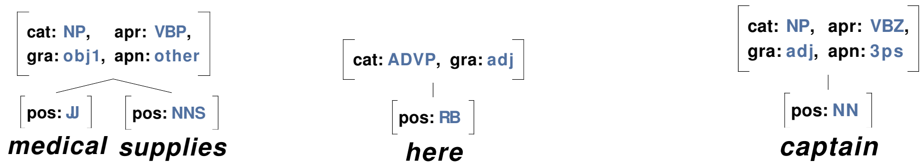
David Traum. 2003. Semantics and pragmatics of questions and answers for dialogue agents. In *proceedings of the International Workshop on Computational Semantics*, pages 380–394, January.

Michael White, Rajakrishnan Rajkumar, and Scott Martin. 2007. Towards broad coverage surface realization with CCG. In *Proc. of the Workshop on Using Corpora for NLG: Language Generation and Machine Translation (UCNLG+MT)*.

Huayan Zhong and Amanda Stent. 2005. Building surface realizers automatically from corpora using general-purpose tools. In *Proc. Corpus Linguistics '05 Workshop on Using Corpora for Natural Language Generation*.



syntax:	we do n't	have
operations:	initial tree	comp
semantics:	speech-act.action = assert speech-act.content.polarity = negative	speech-act.content.attribute = resourceAttribute



syntax:	medical supplies	here	captain
operations:	comp	left/right adjunction	left/right adjunction
semantics:	speech-act.content.value = medical-supplies	speech-act.content.object-id = market	addressee = captain-kirk dialogue-act.addressee = captain-kirk speech-act.addressee = captain-kirk

Figure 5: The linguistic resources automatically inferred from the training example in Figure 3.

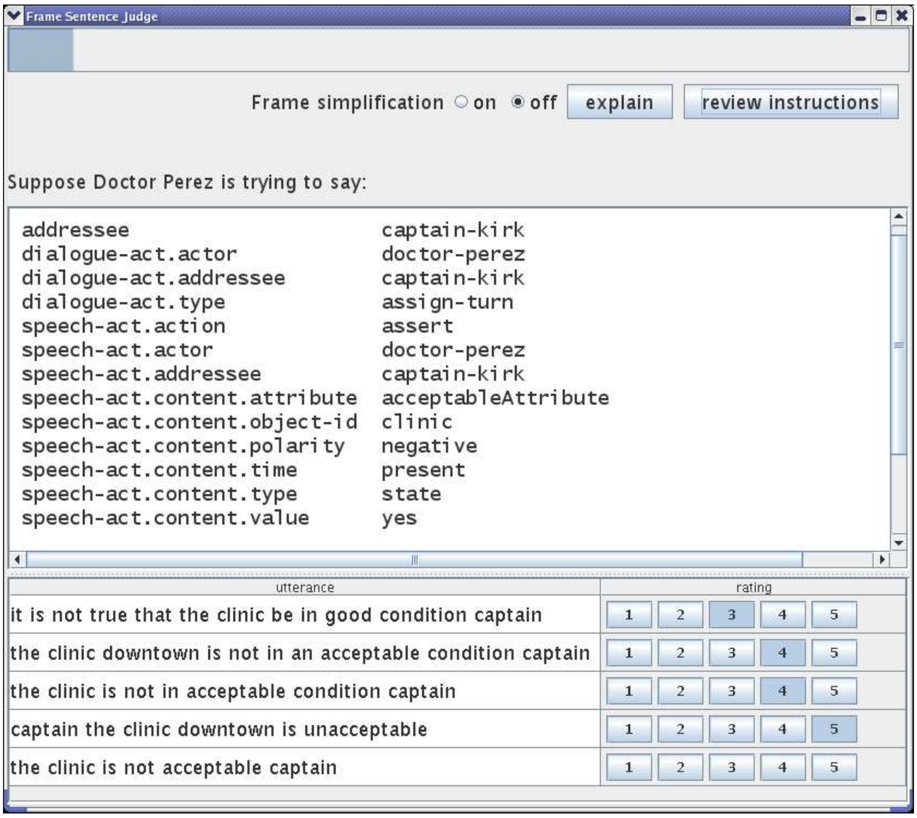


Figure 6: Human rating interface.

Input semantic form

addressee	captain-kirk
dialogue-act.actor	doctor-perez
dialogue-act.addressee	captain-kirk
dialogue-act.type	assign-turn
speech-act.action	assert
speech-act.actor	doctor-perez
speech-act.addressee	captain-kirk
speech-act.content.attribute	acceptableAttribute
speech-act.content.object-id	clinic
speech-act.content.time	present
speech-act.content.type	state
speech-act.content.value	yes

Outputs

Hand-authored

the clinic is acceptable captain

Generated (uncorrected syntax)

Rank	Time (ms)	
1	16	*the clinic is up to standard captain*
2	94	*the clinic is acceptable captain*
3	78	*the clinic should be in acceptable condition captain*
4	16	*the clinic downtown is currently acceptable captain*
5	78	*the clinic should agree in an acceptable condition captain*

Generated (corrected syntax)

Rank	Time (ms)	
1	47	*it is necessary that the clinic be in good condition captain*
2	31	*i think that the clinic be in good condition captain*
3	62	*captain this wont work unless the clinic be in good condition*

Sentence retriever

the clinic downtown is not in an acceptable condition captain

Figure 7: The utterances generated for a single test example by different evaluation conditions. Generated outputs whose rank (determined by derivation probability) was higher than 1 were not rated in the evaluation reported in this paper, but are included here to suggest the potential of our approach to provide a variety of alternative outputs for the same requested semantic form. Note how the output of sentence retriever has the opposite meaning to that of the input frame.