

# Comparing grammar-based and robust approaches to speech understanding: a case study

*Sylvia Knight<sup>1</sup>, Genevieve Gorrell<sup>2</sup>, Manny Rayner<sup>2</sup>, David Milward<sup>1</sup>, Rob Koeling<sup>2</sup>, Ian Lewin<sup>2</sup>*

<sup>1</sup>SRI International

23 Millers Yard, Mill Lane, Cambridge, CB2 1RQ, UK

sylvia@cam.sri.com, milward@cam.sri.com

<sup>2</sup>netdecisions

Wellington House, East Road, Cambridge CB1 1BH, UK

manny.rayner@netdecisions.co.uk, genevieve.gorrell@netdecisions.co.uk

rob.koeling@netdecisions.co.uk, ian.lewin@netdecisions.co.uk

## Abstract

Previous work has demonstrated the success of statistical language models when enough training data is available [1], but despite that, grammar-based systems are proving the preferred choice in successful commercial systems such as HeyAnita [2], BeVocal [3] and Tellme [4], largely due to the difficulty involved in obtaining a corpus of training data. Here we trained an SLM on data obtained using a grammar-based system and compared the performance of the two systems with regards to recognition. We also parsed the output of the SLM using a robust parser and compared the accuracy of the semantic output of the systems. The SLM/robust parser showed considerable improvement on unconstrained input, and similar precision/recall (per slot value) on utterances provided by trained users.

## 1. Introduction

Once upon a time, say around 1995, people used to know how to build speech language understanding systems. You could go into pretty much any university engineering department (back then, that was where people built speech systems), and you would get a variant of the standard recipe, which went something like this: do some Wizard of Oz simulation to collect a domain corpus; use the corpus to train a statistical language model; incorporate the statistical language model into a recogniser; build a robust phrase-spotting parser to analyse the output of the recogniser and produce semantic representations in the form of slot/filler pairs. The system would employ a user-initiative or mixed-initiative dialogue strategy, probably hard-coded in LISP or C++ – system initiative dialogue was too boring, and didn't fit the training data. Some excellent systems were available to convince sceptics that the methodology worked, the most impressive probably being the front-runners in the yearly DARPA bakeoffs [5, 6, 7]. There was healthy debate about the details, and some interesting speculative alternatives, but it was always clear what the mainstream was.

Fast forward to 2001, and it is remarkable how much the picture has changed. There are still quite a few academics working with some version of the 1995 architecture. However these are rapidly becoming outnumbered by a new breed of implementor, usually employed by a commercial organisation, who is using quite a different recipe. This time, the basic strategy is something like the following: build on top of a standard

commercial platform, usually Nuance [8] or Speechworks [9]; define the language model using a hand-coded grammar written in some subset of CFG, with associated semantic annotations; and use a system-initiative dialogue strategy, coded either in a proprietary framework like Nuance's SpeechObjects [10] or SpeechWorks' Dialogue Modules [11], or, increasingly often, in VoiceXML [12]. People using this kind of development methodology are generally quite happy with it, and in many cases are not even aware that the older SLM/robust methods exist. Considering that Nuance and SpeechWorks do not officially support SLMs, this is not entirely surprising.

The critical problem is training data; the ATIS systems quoted above rely for their success on the availability of a large, carefully transcribed, high-quality domain corpus of over 20 000 utterances. Unfortunately, collecting this kind of data is extremely expensive; creating the ATIS corpus took over a year, and cost on the order of \$1M [13]. Most commercial projects cannot consider making a comparable investment in corpus collection. It is not hard to see why so many people have abandoned the old methodology.

It seems to us, however, that the old and new methodologies are not as incompatible as they first appear. Suppose that we have built a modern Nuance- or SpeechWorks-based system, using a CFG grammar which also serves as a language model. In the course of developing and testing the system, we are bound to acquire a substantial corpus of utterances, representing more or less typical system input. The wavefiles cost nothing to record, and only a modest sum to transcribe. The basic question we will investigate here is whether it is practically feasible to use the resulting corpus as the starting point for constructing a second version of the system, built using the old (statistical/robust) methodology. If this "robustified" version turns out to have concrete advantages over the original grammar-based one, then we have achieved an interesting synthesis of our two contrasting methodologies.

To descend to specific details, the rest of the paper describes a concrete experiment designed to investigate the feasibility of the approach sketched in the preceding paragraph. We started by implementing a new-style spoken language understanding system, described in Section 2.3. This system is built on top of the Nuance Toolkit, and uses a hand-coded CFG language model. In the course of developing the system, we collected and transcribed a non-trivial domain corpus (Section 2.2). We then

used this corpus to train a statistical language model, and also to guide the construction of a hand-coded robust parser (Section 2.4), whose output format was compatible with that of the original system. We were now in a position to carry out a detailed comparison of the two systems (Section 3). The final section presents our overall conclusions; basically, the bottom line is that the robust/statistical approach is better for relatively unconstrained utterances by naive users of the system, but there is little difference for experienced and well motivated users who have an idea of what the system can cope with.

## 2. Experimental Setup

### 2.1. Domain

The systems process utterances appropriate to the domain of home device control. They simulate English spoken language control of devices around the home. Device states can only be “on” or “off”. Typical utterances might be “Turn off the light in the bathroom”, “Is the heater switched on?”, “What is there in the kitchen?” and “Are the hall and kitchen lights switched on?”.

Mode of interaction is primarily user-initiative. Utterances may include ellipsis (“turn the light on,” “now the heater,”) conjunction (“turn on the fridge and the microwave,”) universal quantification (“turn everything off,”) and pronominal anaphora (“where’s the radio?” “Is it on?”).

Semantic slots filled by the systems include “spec”, for example, “all” or “the”. This information is particularly relevant in this domain; consider “turn the light on” and “turn all the lights on”. Also device slots (light/heater/TV/...) and location slots (kitchen/bedroom/bathroom/...) are filled. There is a pronoun slot, for “it” or “they”, an operation slot for “query” or “command” and an “onoff” slot.

### 2.2. Corpus

The initial training corpus consisted of 4000 utterances collected throughout the development process of the grammar-based system, and consisting of everything that was said to the system over a period of time while it was deployed as a demonstrator over a telephone line with a simple dialogue manager. It includes utterances from both experienced and naive users. Of these utterances, 464 are out of coverage of the grammar, which shows that the corpus is highly conformant to the grammar. This is to be expected, as the collection process strongly encourages the users to modify their commands until they achieve the required response.

200 sentences from this corpus were used to train an extremely basic statistical language model (Section 2.4), which was used in conjunction with the robust parser to collect a more varied corpus (“bootstrapping”). Users were encouraged to speak to the system without prior knowledge of its capabilities, and they could type in corrections if the utterance was misidentified. These utterances tended to be longer, and the majority were out of coverage of the grammar. 407 utterances were collected in this manner, of which 207 were reserved as test data.

Following completion of development on the grammar-based system, it was used to collect a further 438 utterances, which again consisted of everything that was said to the system over a period of time. These form the second part (the “constrained data” category) of the test set.

	In coverage	Out of coverage
Constrained	373	65
Unconstrained	39	168

Table 1: Test set composition

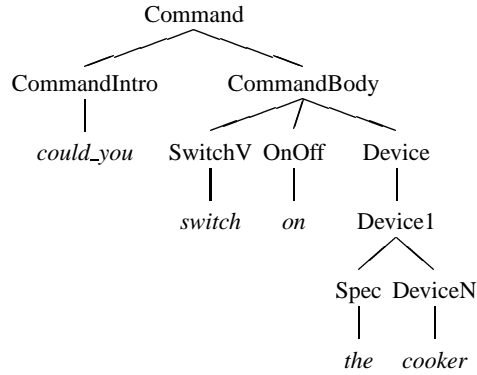


Figure 1: Parse tree for “could you switch on the cooker”

### 2.3. Grammar-Based System

The grammar-based system involves a grammar, implemented in Nuance Toolkit Grammar Specification Language (GSL; [14]), which directly encodes slot-value semantics, being used within a context provided by a dialog manager, encoded using the Nuance DialogueBuilder API. The grammar contains 504 context-free rules; of these, 114 are “lexical” (i.e. have only terminal symbols on their right-hand sides), while the remaining 390 are “grammatical” (i.e. contain at least one non-terminal on the right-hand side).

To give an idea of how the grammar works, Figure 1 presents a slightly simplified derivation of the utterance “could you switch on the cooker”. This yields the following semantic representation;

```

<operation command> <onoff on>
<device1 cooker> <spec1 existential>

```

The dialogue manager simulates a house consisting of eight rooms and containing some 20 devices. Within this context it handles such problems as ambiguity (“turn the heater on”, “I’m sorry, I don’t know which heater you mean”, “the one in the bathroom”) and presupposition errors (“turn the heater on”, “the bathroom heater is already on”). The state of the world can be queried and the effects of commands verified. This context serves to broaden the range of utterances inspired by the system.

### 2.4. Statistical Language Model/Robust Parser

A standard back-off trigram model was created from the training corpus discussed above, using the SRILM Toolkit [15]. For the creation of the initial model necessary for the corpus collection stage, only 200 constrained utterances were used, and in this case performance was significantly improved either by manually clustering by room name and device name, or by using the automatic clustering algorithm described in [16]. However, these methods did not improve performance in the full-size model which used 4200 sentences (200 of which being unconstrained). It seems that for this particular domain, this size is at the level where problems with data sparseness are outweighed by the extra information given by specific devices and

Grammar	Onoff	Device	Location	Operation	Spec	Total
Constrained	94/95	92/96	94/94	95/94	48/88	82/93
Unconstrained	29/49	62/53	61/32	80/72	10/35	50/62
In coverage	95/96	93/96	95/94	95/95	52/89	84/94
Out of coverage	38/42	60/33	63/44	31/37	45/60	47/48
Overall	80/71	68/71	66/68	85/75	49/76	67/73

Precision/recall for semantic slots using grammar-based system

Robust sytem	Onoff	Device	Location	Operation	Spec	Total
Constrained	93/92	83/82	79/92	88/92	88/52	87/87
Unconstrained	86/53	71/60	65/69	84/72	53/24	75/60
In coverage	96/86	90/78	84/80	91/84	90/77	91/82
Out of coverage	81/61	63/63	62/72	81/86	47/16	69/66
Overall	90/74	82/71	73/77	87/84	72/44	82/75

Precision and recall for semantic slots using robust system

Grammar	WER	SER	SEM
Constrained	19	34	23
Unconstrained	59	83	75
In coverage	9	21	11
Out of coverage	67	100	90
Overall	31	50	40

Recognition rates for grammar-based system

Robust	WER	SER	SEM
Constrained	19	36	30
Unconstrained	30	66	70
In coverage	12	28	22
Out of coverage	36	77	81
Overall	25	46	44

Recognition rates for robust system

locations—“the lounge tv” is much more probable than “the lounge toaster”, but the reverse will apply if the first word is “kitchen”.

The 1-best output of the speech recogniser is passed to a robust parser which is designed to allow graceful degradation in the face of ungrammatical or noisy input. The original version of the system [17] was designed for a route planning domain where input often contains redundant information, and a plausible fragment in a particular context (including the dialogue history) is to be preferred over an implausible analysis of an utterance as a full sentence.

Home command dialogues presents a much more challenging environment. Firstly we are assuming user initiative, so we cannot rely upon context to help with the interpretation (in fact, the evaluation assumes a null context). Secondly, there is less redundancy. To interpret a phrase such as “turn all kitchen lights on” we have to interpret every word correctly. Moreover, we cannot just extract slot values such as locations, or devices, but also have to associate them appropriately: the interpretation of “turn off the heater in the living room and the kitchen oven” must associate “living room” with “heater” and “kitchen” with “oven”.

To achieve appropriate associations, and to minimise the number of specific rules, the approach adopted combines phrase spotting with a compositional approach to building an interpretation. The input is parsed providing structural relationships between indexed items. This may or may not correspond to a complete parse of the input. There are then mapping rules which take pieces of the input (which are individually indexed) and map to pieces of output. There are just over 50 mapping rules. The simplest look for particular words and map directly to a slot value e.g.

goodbye --> <meta goodbye>

Other rules require both content words and structural information from the input, and preserve indexing information to allow appropriate associations in the output. For example, the interpretation of “kitchen” in “all the kitchen lights” requires kitchen to be a modifier of a noun, and passes up the index for “light” to the “the” and subsequently to the “all”, and asserts that the location of this index is “kitchen”. Development of these rules, and the addition of lexical entries for this domain took of the order of one person week.

Evaluation of the parser on training examples showed relatively few errors on the one-best input. We investigated letting the parser choose from an n-best list, but this currently

gives worse performance. This appears to be due to a lack of a weighting strategy which prefers globally consistent output. The system currently prefers hypotheses which provide a larger number of slot values, even when this means the same device is “on” and “off”.

### 3. Results

The above tables give comparisons of the grammar-based and robust systems, with the utterances broken down by collection method (constrained/unconstrained), and the alternative breakdown (in/out of coverage of the grammar). Recognition performance is evaluated by word error rate (WER) and sentence error rate (SER) as percentages. For semantic performance, the semantic error rate (SEM) measures the percentage of utterance results which do not match the correct interpretation exactly. However, this measure gives no indication of whether the interpretation is mostly correct or wildly wrong. The other tables give a more detailed analysis of the semantic performance, showing percentage precision followed by percentage recall in total and broken down for each slot.

#### 3.1. Recognition

The grammar-based approach has a lower WER for in-coverage sentences, but performs very poorly on out-of-coverage sentences. This is despite the fact that the recogniser was configured to return fragments of the grammar if a complete utterance could not be found, and also to have a confidence threshold of 0, meaning that it should always try to return something, however unlikely. The SLM makes slightly more word errors for in-coverage sentences, often tending to miss out words such as “the”, but its greater flexibility gives it a clear advantage on the unconstrained material which, as well as having more out-of-coverage examples, has many stutters, hesitations and more background noise.

#### 3.2. Understanding

The semantic error rate is the only figure which shows the grammar-based system outperforming the robust one. This is due to its far better performance on in-coverage data, benefiting from tight coupling to the recogniser and features such as the enforcement of agreement constraints. The robust system does badly by this measure (at least currently) because it does not try to impose a consistent, full interpretation.

In situations where a partial interpretation is still useful,

the precision/recall figures provide a fairer indication of performance. The table shows that the robust system in particular is good at distinguishing operation(query) from operation(command), and in many cases the errors made by both systems are in the “spec” slot and may not be crucial. Both systems are able to distinguish “on” and “off” well, despite the single-phoneme difference. Overall, the precision/recall figures support the view that the SLM’s performance degrades gracefully in difficult conditions, and can return mostly-accurate slot values when the grammar-based system is struggling.

## 4. Conclusion

The basic question we are asking is whether the robust version of the system is capable of achieving better performance on the speech understanding task than the original grammar-based system. Not entirely to our surprise, it turns out that the answer hinges crucially on what exactly we mean by “understand”, and what population of users the system is evaluated on. There is in fact a spectrum of choices along both these dimensions.

At one extreme, we can take a hard-line position, and require full understanding of the intended content of an utterance; we can also base our evaluation on sophisticated users, who have had time to build up a detailed model of the types of utterance the system can deal with. Under these assumptions, the evidence suggests that a robust system has little to offer compared to the grammar-based alternative. The users know the system’s coverage, and are usually able to stay inside its bounds. If they do so, the system finds it relatively easy to understand everything they say. The rigidity of the grammar works to the system’s advantage, by constraining recognition enough that many utterances will be recognised without any loss of content.

At the opposite extreme, we can treat understanding as an inherently incomplete or partial process, and assume that users will have only a vague idea of the system’s capabilities. It is unrealistic to aim for full comprehension of utterances produced by this kind of user, and a partial understanding model is natural, with a dialogue manager which can exploit partial results during its interaction with the user. Under these changed circumstances, we found that the grammar-based system was comfortably outperformed by the robust alternative. Neither system did very well at complete understanding, but the robust system tended to retrieve much more content.

What we find interesting and unexpected is that it turned out to be so easy to use the corpus data painlessly collected during the development of the grammar-based system, and use it to create a robust version which was better than the original one in many plausible circumstances of use. It seems to us that this opens up several interesting avenues for further exploration: most immediately, we intend to experiment with a hybrid architecture which combines both types of speech understanding. This could either involve running both systems in parallel, using appropriate evaluation metrics to choose between the outputs, or alternatively could use the grammar based approach initially, but back off to the robust approach when confidence is low. We hope to report on this in due course.

## 5. Acknowledgements

This work was funded by the EU 5th Framework project, D’Homme, Dialogues in the Home Machine Environment.

## 6. References

- [1] Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin, “A statistical approach to machine translation,” *Computational Linguistics*, vol. 16, pp. 79–85, 1990.
- [2] HeyAnita, *HeyAnita*, <http://heyanita.com/html/main/index.html>, 2001, As of 7 February 2001.
- [3] BeVocal, *BeVocal*, <http://www.bevocal.com/index.html>, 2001, As of 7 February 2001.
- [4] Tellme, *Tellme*, <http://www.tellme.com/>, 2001, As of 7 February 2001.
- [5] P. J. Price, “Evaluation of spoken language systems: The atis domain,” in *Proc. of the Speech and Natural Language Workshop*, Hidden Valley, PA, 1990, pp. 91–95.
- [6] DARPA ’92: *Proceedings of the Speech and Natural Language Workshop*, Morgan Kaufman, San Mateo, CA, 1992.
- [7] ARPA ’94: *Proceedings of the Speech and Natural Language Workshop*, Morgan Kaufman, San Mateo, CA, 1994.
- [8] Nuance Communications, *Nuance Home*, <http://www.nuance.com>, 2001, As of 7 February 2001.
- [9] SpeechWorks International, *SpeechWorks International*, <http://www.speechworks.com>, 2001, As of 7 February 2001.
- [10] Nuance, *SpeechObjects*, <http://www.nuance.com>, 2001, As of 30 March 2001.
- [11] SpeechWorks International Inc., *DialogModules*, <http://www.speechworks.com>, 2001, As of 30 March 2001.
- [12] VoiceXML Forum, *VoiceXML Forum*, <http://www.voicexml.org/index.html>, 2001, As of 7 February 2001.
- [13] C. Hemphill, J. Godfrey, and G. Doddington, “The ATIS spoken language systems pilot corpus,” in *Proc. of the Speech and Natural Language Workshop*, Hidden Valley, PA, 1990, pp. 96–101.
- [14] Nuance Communications, *Nuance Speech Recognition System Developer’s Manual version 6.2*, 1380 Willow Road, Menlo Park, CA 94025, 1999.
- [15] Andreas Stolke, *SRI Language Modelling Toolkit*, <http://www.speech.sri.com/projects/srlm>, 2001.
- [16] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, “Class based n-gram models of natural language,” *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.
- [17] D. Milward, “Distributing representation for robust interpretation of dialogue utterances,” in *Proceedings of the 38th Annual Meeting of the Association of Computational Linguistics, ACL-2000*, Hong Kong, 2000, pp. 133–141.