UNIVERSITÉ DE GENÈVE

Archive ouverte UNIGE
https://archive-ouverte.unige.ch

Chapitre d'actes | 2008 | Open Access

This version of the publication is provided by the author(s) and made available in accordance with the copyright holder(s).

Training Statistical Language Models from Grammar-Generated Data: a Comparative Case-Study

Hockey, Beth Ann; Rayner, Emmanuel; Christian, Gwen

How to cite

HOCKEY, Beth Ann, RAYNER, Emmanuel, CHRISTIAN, Gwen. Training Statistical Language Models from Grammar-Generated Data: a Comparative Case-Study. In: Proceedings of the GoTAL Conference. Gothenburg (Sweden). [s.l.] : [s.n.], 2008.

This publication URL: https://archive-ouverte.unige.ch//unige:3475

© This document is protected by copyright. Please refer to copyright holder(s) for terms of use.

Training Statistical Language Models from Grammar-Generated Data: a Comparative Case-Study

Beth Ann Hockey[1], Manny Rayner[2], and Gwen Christian[3]

[1] UCSC UARC, Mail Stop 19-26
NASA Ames Research Center
Moffet Field, CA 94035
bahockey@ucsc.edu
[2] University of Geneva, TIM/ISSCO
40 bvd du Pont-d'Arve
CH-1211 Geneva 4, Switzerland
Emmanuel.Rayner@issco.unige.ch
[3] Dept of Linguistics
UC Santa Cruz
gwenlle@gmail.com

Abstract. Statistical language models (SLMs) for speech recognition have the advantage of robustness, and grammar-based models (GLMs) the advantage that they can be built even when little corpus data is available. A known way to attempt to combine these two methodologies is first to create a GLM, and then use that GLM to generate training data for an SLM. It has however been difficult to evaluate the true utility of the idea, since the corpus data used to create the GLM has not in general been explicitly available. We exploit the Open Source Regulus platform, which supports corpus-based construction of linguistically motivated GLMs, to perform a methodologically sound comparison: the same data is used both to create an SLM directly, and also to create a GLM, which is then used to generate data to train an SLM. An evaluation on a medium-vocabulary task showed that the indirect method of constructing the SLM is in fact only marginally better than the direct one. The method used to create the training data is critical, with PCFG generation heavily outscoring CFG generation.

1 Introduction

Non-trivial speech recognition always requires some kind of language model [1]. At least in the world of research, it has generally been assumed that language models are best constructed using some kind of data-driven process; the most common alternative in practice is the N-gram grammar. We will generically refer to models built in this way as "Statistical Language Models" or SLMs.

SLMs perform extremely well when there is adequate training data available, but in practice this is not always the case. When training data is limited or, in the worst case, completely unavailable, an alternative method is to construct the language model as a hand-coded grammar [2–5]. We will refer to models of this kind as "Grammar-based Language Models" or GLMs. GLMs appear to be particularly suitable for applications which require high levels of accuracy, and which will also be used by expert users, who can reasonably be expected to produce a high percentage of in-coverage material [6–9]. The distinction between SLMs and GLMs is by no means black-and-white. SLMs can contain embedded GLM-style subgrammars that define simple types of phrase like dates or times [10]. In the other direction, once a GLM has been created, it is possible to use available data to perform statistical tuning, which technically transforms the GLM into a type of SLM. We will have more to say about this later.

Statistical tuning of a GLM is certainly one way to add some of the advantages associated with SLMs. It fails, however, to address the key problem, which is brittleness. In general, grammar-based speech recognition tends to be unforgiving for naive users, since it gives results only for utterances within grammar coverage. This suggests another compromise position between the two methodologies. As noted, SLMs clearly perform well when they are trained on enough data. The grammar in a GLM can also be used to *generate* data; this data can be used to train an SLM. The hope is that the result will combine the advantages of both methodologies. The final language model is an SLM, so it will not be subject to brittleness; but since this language model is created from a GLM, it will be possible to achieve reasonable performance without large amounts of training data.

Although the idea of creating SLMs from GLM-generated data has been used successfully in more than one study [11, 12], one cannot help feeling that there is something, methodologically speaking, that is slightly suspicious about it. It is always clear what data has been used to construct an SLM; it is, however, much harder to be quantative about the process of building a GLM. When a grammar writer hand-codes a grammar, there are always utterances that they have in mind to cover. If those items were recorded as the grammar was built, they would constitute a corpus that represented what data the hand-coded grammar was "trained" on. The same corpus could be used for other purposes, in particular for explicit training of an SLM. It is certainly possible *a priori* that this would produce a recognizer that yielded just as good performance as one built through the roundabout route of first creating a grammar, and then using it to generate training data. However, grammar-writers are rarely, if ever, methodical enough

to write down all their example sentences, and comparisons of the kind suggested are hard to carry out in practice.

This kind of problem is inherent in any comparison between data-driven machine learning and hand-coded rules. However, the Regulus project offers an approach to address this problem. Regulus [13] is an Open Source toolkit for spoken language system development, which builds grammar based language models for the commercial Nuance[4] platform using example-based methods driven by small corpora of examples. In [14], it was shown how Regulus made possible a methodologically fair comparison between a GLM and a normal SLM on a medium-vocabulary speech-understanding task; the same data could be used explicitly to build both language models, rendering irrelevant any speculation about intangible grammar-writer's intuitions.

The paper also showed up another potential methodological pitfall. When recognizers derived from the two language models were compared in terms of Word Error Rate (WER) on a corpus which contained data both in-coverage and out-of-coverage with respect to the GLM, the SLM-based recognizer produced slightly better performance. Further analysis, however, revealed that the raw WER scores were in fact very misleading; they represented the average of better performance of the SLM on out-of-coverage data, and worse performance on in-coverage data. Performance of both recognizers on the out-of-coverage data (WER = 48% for SLM and 58% for GLM) was however so bad as to be essentially uninteresting in the context of the speech translation task, which required precise, fine-grained analysis[5]. Conversely, performance on in-coverage data showed a WER for the GLM (6%) that was less than half that of the SLM (13%), an extremely useful improvement. This is by no means the first study which has shown up the weakness of WER as a metric for evaluating speech *understanding*, as opposed to raw speech *recognition* [15].

In the present paper, we use the Regulus platform and a methodology which borrows several elements from [14] to evaluate the idea of creating SLMs from GLM-generated training data. This approach makes it possible to address the key methodological problems in a sound way, which has not been the case in previous studies. The paper is organized as follows. Section 2 describes the framework that we have adopted for performing the language modeling experiments. Section 3 describes the experiments performed. Section 4 summarizes the results and draws conclusions for language modeling in sparse data situations.

2 Experimental Framework

As discussed in the previous section, a key problem with earlier work has been the impossibility of knowing what "seed corpus" was used to construct the hand-coded grammars used to generate the SLM training data. The Regulus platform

[4] Nuance 8.5 was used for the work discussed in this paper.

[5] In tasks involving coarse-grained speech understanding, for example call-routing, this difference might have been more important.

allows us to address these issues head-on, since it makes the role of the seed corpus completely explicit. The basic idea is to start with a general resource grammar, and then use the seed corpus to drive an example-based process that creates the final domain-specific language model. We now present a brief overview of how this is done; the details of the various compilation steps are described in [13, Chapters 9 and 10].

The Regulus release contains a fairly substantial domain-independent feature grammar for English [13, Chapter 8], which also contains a function-word lexicon of about 500 words. The grammar developer adds to them a domain-specific lexicon containing the necessary content words, a domain-specific seed corpus, and a set of "operationality criteria", whose role will be explained shortly. These resources constitute the input to the grammar creation process. The Regulus parser is first used to convert the seed corpus into a set of parse trees. The operationality criteria then define how each tree is to be cut up into a number of subtrees. The rules in each subtree are collapsed into a derived rule. The set of all such derived rules constitutes a specialised version of the original feature grammar.

By construction, the specialised feature grammar produces analyses compatible with those of the original grammar, and covers all the examples in the seed corpus, but will in general have coverage strictly less than that of the original grammar. The specialised feature grammar is next subjected to another compilation phase, which converts it into a CFG grammar in Nuance's GSL notation. Finally, the seed corpus can optionally be used a second time, as training data to convert the CFG grammar into a PCFG grammar. This final conversion stage is performed by the Nuance `compute-grammar-probs` utility.

Nuance contains another utility, `generate`, which can be used to generate an arbitrary number of sentences from a GSL-formatted CFG grammar. We also wrote a utility of our own, which performs generation on GSL-formatted PCFG grammars produced by the Regulus compilation process. Both Nuance's `generate` and our own generation utility work by sampling the space of generated utterances, starting with the root symbol and expanding non-terminals until the result contains only terminals. The critical difference is that `generate`, when randomly choosing a rule to expand a non-terminal N, assigns equal weights to all the productions where N occurs on the LHS. Our PCFG generation utility, in contrast, weights the productions with the probabilities attached to them.

To recapitulate, the process we have just outlined allows us to use a grammar to generate training data for building an SLM, but does it in a way which makes completely explicit which corpus data was used to construct the generation grammar itself. In effect, the "seed corpus" reifies the linguistic intuitions used to build the generation grammar. This has several very useful consequences. In particular, since the seed corpus is just a normal domain corpus, it is also possible to use it directly to train an SLM.

The concrete experiments we describe were performed using English corpora and an English Regulus grammar taken from MedSLT [16], a medium-vocabulary Open Source speech translation system for medical domains. Vocabulary size was

458 words. Most of the detailed aspects of the MedSLT system are not relevant here, but one turned out to be potentially useful. Translation in the system is interlingua-based: source-language representations are translated into interlingua representations, and then into target-language representations. The space of well-defined interlingua representations is defined by means of another Regulus grammar [17]. Not all representations licensed by the source-language grammar produce well-formed interlingua; it can be the case that constraints are hard to formulate at the source-language level, but easy to capture in interlingua. This means that the interlingua can be used as another source of information. Since some of the randomly generated utterances do not produce well-formed interlingua after being passed through the source-language-to-interlingua transfer phase, it is possible to treat the combination of the transfer rules and the interlingua definition as a filter.

The actual construction of the SLMs was performed using the Nuance Say-Anything© utilities. Each SLM was a class trigram model, created using Good-Turing discounting. The classes were defined using a Regulus utility which extracted sets of words with similar syntactic and semantic properties from the relevant specialized grammars. The properties for each class were defined by specifying a small number, usually two or three, paradigm words, and computing the least common generalization of the corresponding lexicon entries.

In the next section, we describe the concrete experiments we carried out using this basic framework.

3 Experiments

We used all of the following different kinds of corpus as input to train SLMs:

Seed The original "seed corpora". This consisted of 948 examples.

CFG-generated Corpora generated from a CFG grammar derived by Regulus from the seed corpus. We created datasets of several different sizes.

PCFG-generated Corpora generated from a PCFG grammar derived by Regulus from the seed corpus. We created datasets of several different sizes.

CFG-generated-filtered Corpora generated by a CFG grammar derived by Regulus from the seed corpus, and then filtered by removing utterances which do not give rise to well-formed interlingua. We created datasets of several different sizes.

PCFG-generated-filtered Corpora generated by a PCFG grammar derived by Regulus from the seed corpus, and then filtered by removing utterances which do not give rise to well-formed interlingua. We created datasets of several different sizes.

We evaluated the quality of the resulting SLMs by using them to perform recognition on the 810-utterance dataset described in [14], which consisted of spontaneously generated utterances collected during studies carried out on naive subjects who had not been involved in system development. 514 of these utterances (63%) were within the coverage of the GLM grammar, and 296 out of coverage.

The main results from the experiments are presented in Tables 1 to 6. As in [14], we calculate WER and Sentence Error Rate (SER) both for the full datasets, and also for the subset consisting only of in-coverage utterances. Our primary reason for using SER as a metric is that it enables us to apply the McNemar sign test, in order to evaluate the significance of differences between recognition performance of different versions. We present significance as one of the following: "not significant", "significant at $P < 0.05$", "significant at $P < 0.01$" and "significant at $P < 0.001$". In the rest of this section, we discuss the implications of the results.

3.1 Different types of corpora

Tables 1 and 2 presents results contrasting different methods for building the SLM training corpora; the first line, for the GLM built using the "seed" corpus, is intended to provide a reference point. Line 2 shows the SLM built from the "seed" corpus. The other recognizers were all built from GLM-generated training corpora of the same size. The small size of these corpora reflects the fact that CFG generation (lines 3 and 4) produces very low-grade data. The interlingua-based filtering operation discards over 99% of it; 4281 was the number of utterances left by filtering from an initial CFG-generated set of 500K utterances, and the other corpora were then truncated to that length[6] Line 3 shows results for unfiltered, and line 4 for filtered data. Lines 5 and 6 are PCFG-generated sets, with and without interlingua filtering.

Several immediate conclusions can be drawn. First, as shown by line 1 in Table 2, PCFG generation is vastly superior to CFG. Given that CFG-generated data clearly did not deliver interesting performance, we only used PCFG-generated data for the other experiments.

A more interesting result (line 2 in Table 2) is that even the best SLM trained on PCFG-generated data (line 6 in Table 1) is not clearly better than the one trained directly from the original "seed" corpus (line 2, same table). The PCFG-generated data produces a better WER; however, its SER is significantly worse.

Alhough interlingua filtering does result in some improvement (lines 4 and 5 in Table 2), it does not have a very large effect on PCFG-generated data, and in fact the difference in SER is not significant.

Finally, lines 6 and 7 in Table 2 show that the plain GLM recognizer produces significantly better performance than any of the other versions. It should be noted, however, that the generated training sets produced in these first experiments are quite small. The next set of experiments investigates what happens as they are made larger.

[6] To test for the possibility of bias in the truncated unfiltered corpora, we created both "head" (taken from the beginning of the larger file) and "tail" (taken from the end) versions of the needed size. Performance on the head and tail versions was nearly the same, leading us to conclude that it is unlikely that the truncation procedure is creating a skewed corpus. The head versions are used in the paper.

	Version	size	WER	SER
1	seed corpus GLM	948	21.96%	50.62%
2	seed corpus SLM	948	27.74%	58.40%
3	CFG/unfiltered	4281	49.0%	88.4%
4	CFG/filtered	4281	44.68%	85.68%
5	PCFG/unfiltered	4281	25.98%	65.31%
6	PCFG/filtered	4281	25.81%	63.70%

Table 1. Recognition performance for SLMs trained on different types of generated data. "Size" = number of utterances in training set; "WER" = Word Error Rate on test set of in-coverage and out of coverage material; "SER" = sentence error rate on test set of in-coverage and out of coverage material. GLM results included for comparison

	First	Second	Score	Significance
1	CFG/unfiltered	**PCFG/unfiltered**	12–199	$P < 0.001$
2	**seed corpus SLM**	PCFG/filtered	87–44	$P < 0.001$
3	**seed corpus SLM**	CFG/unfiltered	244–15	$P < 0.001$
4	CFG/unfiltered	**CFG/filtered**	27–49	$P < 0.05$
5	PCFG/unfiltered	PCFG/filtered	16–29	not significant
6	**seed corpus GLM**	seed corpus SLM	124–47	$P < 0.001$
7	**seed corpus GLM**	PCFG/filtered	142–36	$P < 0.001$

Table 2. Significance of differences between some of the versions of the recogniser listed in Table 1, according to the McNemar sign test performed on SER. Significantly better results are marked in **bold**.

3.2 Increasing the size of the training set

When SLMs are trained on human-generated data, performance usually improves for some time as more data is added. A common rule of thumb when building commercial SLM-based systems is that one should aim to collect about 20 000 utterances. Tables 3 and 4 presents results for SLMs trained off PCFG-generated corpora of increasing size. As in the first set of experiments, unfiltered data sets were truncated to make them equal in size to the corresponding filtered ones; the labels "50K", "1000K" and "1500K" indicate the number of utterances in the original unfiltered PCFG-generated set, prior to truncation. The amount of training data was incremented until addition of data no longer resulted in an improvement in the error rates.

The recognizers trained on filtered data continued to improve as we increased the size of the training set (lines 1 and 2, Table 4), though the improvement between the largest set (497 798 utterances) and the second-largest (331 328 utterances) was not significant. With unfiltered data, we were surprised to discover that moving from 331 328 utterances to 497 798 utterances actually *degraded* performance (line 4, Table 4). It is not clear why this should be, but we can at least note that the filtering operation appears to make the data less noisy.

	Version	size	WER	SER
1	seed corpus GLM	948	21.96%	50.62%
2	seed corpus SLM	948	27.74%	58.40%
3	50K PCFG/unfiltered	16 619	24.84%	62.47%
4	50K PCFG/filtered	16 619	23.80%	59.51%
5	1000K PCFG/unfiltered	331 328	24.12%	58.77%
6	1000K PCFG/filtered	331 328	23.62%	57.28%
7	1500K PCFG/unfiltered	497 798	24.38%	59.88%
8	1500K PCFG/filtered	497 798	23.76%	57.16%

Table 3. Recognition performance as training set size increases. "Size" = number of utterances in training set; test set includes both in-coverage and out of coverage; "WER" = Word Error Rate; "SER" = sentence error rate

	First	Second	Score	Significance
1	50K PCFG/filtered	**1000K PCFG/filtered**	22–40	$P < 0.05$
2	1000K PCFG/filtered	1500K PCFG/filtered	4–5	not significant
3	50K PCFG/unfiltered	**1000K PCFG/unfiltered**	22–52	$P < 0.001$
4	**1000K PCFG/unfiltered**	1500K PCFG/unfiltered	11–2	$P < 0.05$
5	1000K PCFG/unfiltered	seed corpus SLM	68–71	not significant
6	1500K PCFG/unfiltered	seed corpus SLM	68–80	not significant
7	1500K PCFG/filtered	seed corpus SLM	75–65	not significant
8	1500K PCFG/filtered	**1000K PCFG/unfiltered**	27–14	$P < 0.05$
9	1000K PCFG/unfiltered	**seed corpus GLM**	33–99	$P < 0.001$
10	1500K PCFG/unfiltered	**seed corpus GLM**	32–107	$P < 0.001$
11	1500K PCFG/filtered	**seed corpus GLM**	36–89	$P < 0.001$

Table 4. Significance of differences between some of the versions of the recogniser listed in Table 3, according to the McNemar sign test performed on SER. Significantly better results are marked in **bold**.

The best recognizer trained on unfiltered data (line 5, Table 3) had lower WER than the "seed corpus" SLM recogniser (line 2, same table). SER, however, was almost the same between these two versions, and the difference was not significant (line 5, Table 4). The best recognizers trained on filtered data (lines 6 and 8, Table 3) did better, and outscored the "seed corpus" SLM on both WER and SER. The difference on SER, however, was again not significant (line 7, Table 4). The difference between the best filtered and the best unfiltered versions was significant (line 8, Table 4), again supporting the claim that filtering helps.

In terms of both WER and SER, however, all versions were still clearly inferior to the GLM recognizer (lines 9–11, Table 4). Since the superiority of the GLM is most marked on in-coverage data, our third set of experiments focussed on this.

3.3 In-coverage performance

The third and final set of experiments measured performance only on the 514-utterance subset of the data that was within the coverage of the GLM. As pointed out earlier, comparisons between GLM and SLM models depend heavily on the mix of in-coverage and out of coverage data encountered in the test data. Performance of both models is generally dismal on out-of-coverage data, and consequently not very interesting; performance on in-coverage data is a more useful metric. The results of these tests are shown in Tables 5 and 6.

The relationships between most of the scores are similar to those in Table 3 above. Two points are worth noting. First, as expected, restriction to in-coverage data increases the difference between the GLM recognizer and the others in terms of both WER and SER; for both metrics, we see a relative decrease of over 35% between results for the GLM and the best of the other versions. The second point, rather more interestingly, is that the best SLM version is now the one created from filtered PCFG-generated data (line 8, Table 5). This version is significantly better than the "seed corpus" SLM (Table 6, line 7).

	Version	size	WER	SER
1	seed corpus GLM	948	7.00%	22.37%
2	seed corpus SLM	948	14.40%	42.02%
3	50K PCFG/unfiltered	16 619	14.13%	46.11%
4	50K PCFG/filt	16 619	12.76%	40.86%
5	1000K PCFG/unfiltered	331 328	11.83%	38.91%
6	1000K PCFG/filtered	331 328	11.21%	36.58%
7	1500K PCFG/unfiltered	497 798	12.35%	40.66%
8	1500K PCFG/filtered	497 798	11.25%	36.19%

Table 5. Recognition performance as training set size increases, on in-coverage material only. "Size" = number of utterances in training set; "WER" = Word Error Rate; "SER" = sentence error rate

4 Summary and Conclusions

The idea of creating a statistical language model by using a grammar to generate training data has been known for some time, but previous attempts to evaluate it objectively have run into methodological difficulties. The study we have presented here has solved what we view as the key problem. By using the trainable Regulus grammar-development framework, we have been able to quantify the data that was used to create the grammar. This has made it possible for us to compare, on the one hand, the indirect method of using the data first to create a grammar, which then creates training data for an SLM, and on the other the direct method of simply creating an SLM from the original seed corpus. We have

	First	Second	Score	Significance
1	50K PCFG/filtered	**1000K PCFG/filtered**	16–38	$P < 0.01$
2	1000K PCFG/filtered	1500K PCFG/filtered	2–4	not significant
3	50K PCFG/unfiltered	**1000K PCFG/unfiltered**	15–52	$P < 0.001$
4	**1000K PCFG/unfiltered**	1500K PCFG/unfiltered	11–2	$P < 0.05$
5	1000K PCFG/unfiltered	seed corpus SLM	69–53	not significant
6	1500K PCFG/unfiltered	seed corpus SLM	68–61	not significant
7	**1500K PCFG/filtered**	seed corpus SLM	74–44	$P < 0.01$
8	1000K PCFG/unfiltered	**seed corpus GLM**	13–98	$P < 0.001$
9	1500K PCFG/unfiltered	**seed corpus GLM**	12–106	$P < 0.001$
10	1500K PCFG/filtered	**seed corpus GLM**	17–88	$P < 0.001$
11	**1500K PCFG/filtered**	1000K PCFG/unfiltered	25–11	$P < 0.05$

Table 6. Significance of differences between some of the versions of the recogniser listed in Table 5, evaluated on in-coverage data only, according to the McNemar sign test performed on SER. Significantly better results are marked in **bold**.

also compared the utility of generating SLM training data using CFG and PCFG versions of the grammar, investigated the effect of filtering the generated data using the MedSLT interlingua, and looked at the relationship between the size of the generated training set and the quality of the SLM it produces. Our experiments have used English grammars and data from the Open Source MedSLT project.

The key result, as we see it, is that the indirect method of constructing the SLM actually turns out to be only marginally better than the direct one. When measured on the whole dataset (Tables 3 and 4), several of the versions produced better WER. However, only the best one yielded any reduction in SER, this reduction was not statistically significant, and producing it required the extra interlingua-based filtering step. This is consistent with the intuition that the GLM grammar essentially contains only a little more information than the corpus used to create it. The SLM trained from the PCFG-generated corpus does in fact produce a slight improvement over the one trained from the "seed" corpus. We hypothesize that this improvement is due to a combination of two factors. First, the PCFG generation process probably helps, in effect, to smooth the training data; second, it seems reasonable to believe that the general resource grammar used to build the GLM contributes at least some information.

Restricting evaluation only to in-coverage data did finally produce a result where an SLM recognizer trained on generated data significantly outperformed the one trained on the seed corpus. This is again, unfortunately, still not very interesting, since the main point of the SLM is to achieve greater robustness on out-of-coverage material; as we had expected, the GLM recognizer strongly outperformed all the SLM versions on the in-coverage material.

An incidental result that we found interesting was the large difference between the models trained on PCFG-generated data and those trained on CFG-generated data. In retrospect, this should not have been entirely surprising. How-

ever, looking at previous work, it is worth noting that although [11] used PCFG generation, [12] appeared not to. The experiments where we increased the size of the generated corpus suggest that one needs to produce quite a large amount of data, on the order of hundreds of thousands of sentences, before performance tops out.

In conclusion, we think we can reasonably claim to have put the idea of using grammars to create SLM training data on a sounder theoretical footing. Although the results reported here are more negative than positive, we hope that the methodology we present will open new possibilities for research in this area.

Acknowledgments

We would like to thank Nuance for giving us access to the proprietary software used in this research. Work by Manny Rayner was funded by the Fonds National de la Recherche Scientifique (FNRS) under the project "A Swiss Platform for Controlled Language Spoken Dialog Applications".

References

1. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE **77:2** (1989) 257–286
2. Moore, R.: Using natural language knowledge sources in speech recognition. In: Proceedings of the NATO Advanced Studies Institute. (1998) 115–129
3. Dowding, J., Hockey, B., Gawron, J., Culy, C.: Practical issues in compiling typed unification grammars for speech recognition. In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, Toulouse, France (2001) 164–171
4. Rayner, M., Dowding, J., Hockey, B.: A baseline method for compiling typed unification grammars into context free language models. In: Proceedings of Eurospeech 2001, Aalborg, Denmark (2001) 729–732
5. Bos, J.: Compilation of unification grammars with compositional semantics to speech recognition packages. In: Proceedings of the 19th International Conference on Computational Linguistics, Taipei, Taiwan (2002)
6. Stent, A., Dowding, J., Gawron, J., Bratt, E., Moore, R.: The CommandTalk spoken dialogue system. In: Proceedings of the Thirty-Seventh Annual Meeting of the Association for Computational Linguistics. (1999) 183–190
7. Knight, S., Gorrell, G., Rayner, M., Milward, D., Koeling, R., Lewin, I.: Comparing grammar-based and robust approaches to speech understanding: a case study. In: Proceedings of Eurospeech 2001, Aalborg, Denmark (2001) 1779–1782
8. Rayner, M., Hockey, B., Renders, J., Chatzichrisafis, N., Farrell, K.: A voice enabled procedure browser for the International Space Station. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (interactive poster and demo track), Ann Arbor, MI (2005)
9. Chatzichrisafis, N., Bouillon, P., Rayner, M., Santaholma, M., Starlander, M., Hockey, B.: Evaluating task performance for a unidirectional controlled language medical speech translation system. In: Proceedings of the HLT-NAACL International Workshop on Medical Speech Translation, New York (2006) 9–16

10. Wang, Y.Y., Acero, A., Chelba, C., Frey, B., Wong, L.: Combination of statistical and rule-based approaches for spoken language understanding. In: Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP), Denver, CO (2002) 609–612
11. Jurafsky, A., Wooters, C., Segal, J., Stolcke, A., Fosler, E., Tajchman, G., Morgan, N.: Using a stochastic context-free grammar as a language model for speech recognition. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. (1995) 189–192
12. Jonson, R.: Generating statistical language models from interpretation grammars in dialogue systems. In: Proceedings of the 11th EACL, Trento, Italy (2006)
13. Rayner, M., Hockey, B., Bouillon, P.: Putting Linguistics into Speech Recognition: The Regulus Grammar Compiler. CSLI Press, Chicago (2006)
14. Rayner, M., Bouillon, P., Chatzichrisafis, N., Hockey, B., Santaholma, M., Starlander, M., Isahara, H., Kanzaki, K., Nakao, Y.: A methodology for comparing grammar-based and robust approaches to speech understanding. In: Proceedings of the 9th International Conference on Spoken Language Processing (ICSLP), Lisboa, Portugal (2005) 1103–1107
15. Wang, Y.Y., Acero, A., Chelba, C.: Is Word Error Rate a good indicator for spoken language understanding accuracy. In: Proceedings of Eurospeech 2003, Geneva, Switzerland (2003) 609–612
16. Bouillon, P., Rayner, M., Chatzichrisafis, N., Hockey, B., Santaholma, M., Starlander, M., Nakao, Y., Kanzaki, K., Isahara, H.: A generic multi-lingual open source platform for limited-domain medical speech translation. In: Proceedings of the 10th Conference of the European Association for Machine Translation (EAMT), Budapest, Hungary (2005) 50–58
17. Bouillon, P., Halimi, S., Nakao, Y., Kanzaki, K., Isahara, H., Tsourakis, N., Starlander, M., Hockey, B., Rayner, M.: Developing non-European translation pairs in a medium-vocabulary medical speech translation system. In: Proceedings of LREC 2008, Marrakesh, Morocco (2008)