Enhancing commercial grammar-based applications using robust approaches to speech understanding

Matthieu Hébert
Network ASR R+D, Nuance Communications
1500, Université, Suite 935, Montréal, Québec, H3A 3T2, Canada
hebert@nuance.com

Abstract

This paper presents a series of measurements of the accuracy of speech understanding when grammar-based or robust approaches are used. The robust approaches considered here are based on statistical language models (SLMs) with the interpretation being carried out by phrase-spotting or robust parsing methods. We propose a simple process to leverage existing grammars **and** logged utterances to upgrade grammar-based applications to become more robust to out-of-coverage inputs. All experiments herein are run on data collected from deployed directed dialog applications and show that SLM-based techniques outperform grammar-based ones without requiring any change in the application logic.

1 Introduction

The bulk of the literature on spoken dialog systems is based on the simple architecture in which the input speech is processed by a statistical language model-based recognizer (SLM-based recognizer) to produce a word string. This word string is further processed by a robust parser (Ward, 1990) or call router (Gorin et al, 1997) to be converted in a semantic interpretation. However, it is striking to see that a large portion of deployed commercial applications do not follow this architecture and approach the recognition/interpretation problem by relying on hand-crafted rules (context-free grammars - CFGs). The apparent reasons for this are the up-front cost and additional delays of collecting domain-specific utterances to properly train the SLM (not to mention semantic tagging needed to train the call router) (Hemphill et al, 1990; Knight et al, 2001; Gorin et al, 1997). Choosing to use a grammar-based approach also makes the application predictable and relatively easy to design. On the other hand, these applications are usually very rigid: the users are allowed only a finite set of ways to input their requests and, by way of consequences, these applications suffer from high out-of-grammar (OOG) rates or out-of-coverage rates.

A few studies have been published comparing grammar-based and SLM-based approaches to speech understanding. In (Knight et al, 2001), a comparison of grammar-based and robust approaches is presented for a user-initiative home automation application. The authors concluded that it was relatively easy to use the corpus collected during the course of the application development to train a SLM which would perform better on out-of-coverage utterances, while degrading the accuracy on in-coverage utterances. They also reported that the SLM-based system showed slightly lower word error rate but higher semantic error rate for the users who know the application's coverage. In (Rayner et al, 2005), a rigorous test protocol is presented to compare grammar-based and robust approaches in the context of a medical translation system. The paper highlights the difficulties to construct a clean experimental set-up. Efforts are spent to control the *training set* of both approaches to

Bridging the Gap: Academic and Industrial Research in Dialog Technologies Workshop Proceedings, pages 76–83,
NAACL-HLT, Rochester, NY, April 2007.

have them align. The *training sets* are defined as the set of data available to build each system: for a grammar-based system, it might be a series of sample dialogs. (ten Bosch, 2005) presents experiments comparing grammar-based and SLM-based systems for naïve users and an expert user. They conclude that the SLM-based system is most effective in reducing the error rate for naïve users. Recently (see (Balakrishna et al, 2006)), a process was presented to automatically build SLMs from a wide variety of sources (in-service data, thesaurus, WordNet and world-wide web). Results on data from commercial speech applications presented therein echo earlier results (Knight et al, 2001) while reducing the effort to build interpretation rules.

Most of the above studies are not based on data collected on deployed applications. One of the conclusions from previous work, based on the measured fact that in-coverage accuracy of the grammar-based systems was far better than the SLM one, was that as people get more experience with the applications, they will naturally learn its coverage and gravitate towards it. While this can be an acceptable option for some types of applications (when the user population tends to be experienced or captive), it certainly is not a possibility for large-scale commercial applications that are targeted at the general public. A few examples of such applications are public transit schedules and fares information, self-help applications for utilities, banks, telecommunications business, and etc. Steering application design and research based on in-coverage accuracy is not suitable for these types of applications because a large fraction of the users are naïves and tend to use more natural and unconstrained speech inputs.

This paper exploits techniques known since the 90's (SLM with robust parsing, (Ward, 1990)) and applies them to build robust speech understanding into existing large scale directed dialog grammar-based applications. This practical application of (Ward, 1990; Knight et al, 2001; Rayner et al, 2005; ten Bosch, 2005) is cast as an upgrade problem which must obey the following constraints.

1. No change in the application logic and to the voice user interface (VUI)
2. Roughly similar CPU consumption
3. Leverage existing grammars
4. Leverage existing transcribed utterances
5. Simple process that requires little manual intervention

The first constraint dictates that, for each context, the interpretation engines (from the current and upgraded systems) must return the same semantics (i.e. same set of slots).

The rest of this paper is organized as follows. The next Section describes the applications from which the data was collected, the experimental set-up and the accuracy measures used. Section 3 describes how the semantic truth is generated. The main results of the upgrade from grammar-based to SLM-based recognition are presented in Section 4. The target audience for this paper is composed of application developers and researchers that are interested in the robust information extraction from directed dialog speech applications targeted at the general public.

2 Applications, corpus and experimental set-up

2.1 Application descriptions

As mentioned earlier, the data for this study was collected on deployed commercial directed dialog applications. AppA is a self-help application in the internet service provider domain, while AppB is also a self-help application in the public transportation domain. Both applications are grammar-based directed dialogs and receive a daily average of 50k calls. We will concentrate on a subset of contexts (dialog states) for each application as described in Table 1. The *mainmenu* grammars (each application has its own *mainmenu* grammar) contain high-level targets for the rest of the application and are active once the initial prompt has been played. The *command* grammar contains universal commands like "help", "agent", etc. The *origin* and *destination* grammars contain a list of $\sim$ 2500 cities and states with the proper prefixes to discriminate origin and destination. *num_type_passenger* accepts up to nine passengers of types adults, children, seniors, etc. Finally *time* is self explanatory. For each application, the prompt directs the user to provide a specific

Context	Description	Active grammars	Training sentences	Testing utts
AppA_MainMenu	Main menu for the application	mainmenu and commands	5000 (350)	5431 (642)
AppB_MainMenu	Main menu for the application	mainmenu and commands	5000 (19)	4039 (987)
AppB_Origin	Origin of travel	origin, destination and commands	5000 (20486)	8818 (529)
AppB_Passenger	Number and type of passenger	num_type_passenger and commands	1500 (32332)	2312 (66)
AppB_Time	Time of departure	time and commands	1000 (4102)	1149 (55)

Table 1: Description of studied contexts for each application. Note that the AppB_Origin context contains a *destination* grammar: this is due to the fact that the same set of grammars was used in the AppB_Destination context (not studied here). "Training" contains the number of training sentences drawn from the corpus and used to train the SLMs. As mentioned in Sec. 2.3, in the case of word SLMs, we also use sentences that are covered by the grammars in each context as backoffs (see Sec. 2). The number of unique sentences covered by the grammars is in parenthesis in the "Training" column. The "Testing" column contains the number of utterances in the test set. The number of those utterances that contain no speech (noise) is in parenthesis.

piece of information (directed dialog). Each grammar fills a single slot with that information. The information contained in the utterance "two adults and one child" (AppB_Passenger context) would be collapsed to fill the **num_type_passenger** slot with the value "Adult2_Child1". From the application point of view, each context can fill only a very limited set of slots. To keep results as synthesized as possible, unless otherwise stated, the results from all studied contexts will be presented per application: as such results from all contexts in AppB will be pooled together.

2.2 Corpus description

Table 1 presents the details of the corpus that we have used for this study. As mentioned above the entire corpora used for this study is drawn from commercially deployed systems that are used by the general public. The user population reflects realistic usage (expert vs naïve), noise conditions, handsets, etc. The training utterances do not contain noise utterances and is used primarily for SLM training (no acoustic adaptation of the recognition models is performed).

2.3 Experimental set-up description

The baseline system is the grammar-based system; the recognizer uses, on a per-context basis, the grammars listed in Table 1 in parallel. The SLM systems studied all used the same interpretation engine: robust parsing with the grammars listed in Table 1 as rules to fill slots. Note that this allows the application logic to stay unchanged since the set of potential slots returned within any given context is the same as for the grammar-based systems (see first constraint in Sec. 1). Adhering to this experimental set-up also guarantees that improvements measured in the lab will have a direct impact on the raw accuracy of the deployed application.

We have considered two different SLM-based systems in this study: standard SLM (wordSLM) and class-based SLM (classSLM) (Jelinek, 1990; Gillett and Ward, 1998). In the classSLM systems, the classes are defined as the rules of the interpretation engine (i.e. the grammars active for each context as defined in Table 1). The SLMs are all trained on a per-context basis (Xu and Rudnicky, 2000; Goel and Gopinath, 2006) as bi-grams with Witten-Bell discounting. To insure that the word-SLM system covered all sentences that the grammar-based system does, we augmented the training set of

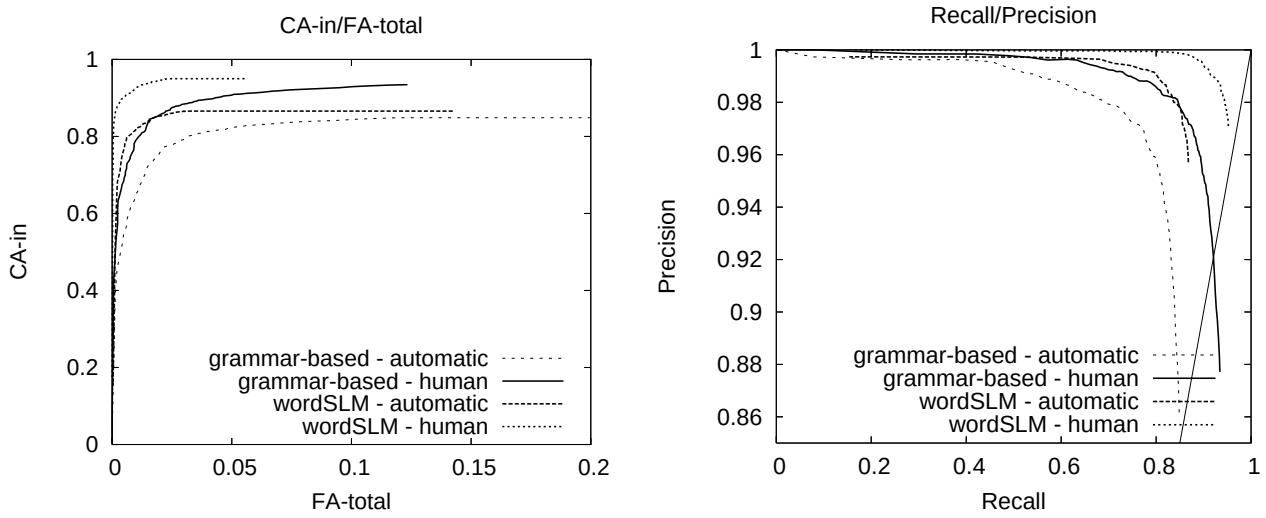

Figure 1: ROC curves for AppA_MainMenu with the automatic or human-generated truth. In each the grammar-based and SLM-based systems are compared.

the wordSLM (see Table 1) with the list of sentences that are covered by the baseline grammar-based system. This acts as a backoff in case a word or bi-gram is not found in the training set (not to be confused with bi-gram to uni-gram backoffs found in standard SLM training). This is particularly helpful when a little amount of data is available for training the wordSLM (see Sec. 4.3).

2.4 Accuracy measures

Throughout this paper, we will use two sets of measures. This is motivated by the fact that application developers are familiar with the concepts of correct/false acceptance at the utterance level. For information extraction (slot filling) from utterances, these concepts are restrictive because an utterance can be partly correct or wrong. In this case we prefer a more relevant measure from the information retrieval field: precision and recall on a per-slot basis. We use the following definitions.

- CA-in = #utts that had ALL slots correct (slot name and value) / #utts that are in-coverage (i.e. truth has at least a slot filled)
- FA-total = #utts that had at least one erroneous slot (slot name or value) / total #utts
- Precision = #slot correct slots (slot name and value) / #slots returned by system
- Recall = #slot correct slots (slot name and value) / #slots potential slots (in truth)

Since applications use confidence extensively to guide the course of dialogue, it is of limited interest to study forced-choice accuracy (accuracy with no rejection). Hence, we will present receiver operating characteristic (ROC) curves. The slot confidence measure is based on redundancy of a slot/value pair across the NBest list. For CA-in and FA-total, the confidence is the average confidence of all slots present in the utterance. Note that in the case where each utterance only fills a single slot, CA-in = Recall.

3 Truth

Due to the large amount of data processed (see Table 1), semantic tagging by a human may not be available for all contexts (orthographic transcriptions are available however). We need to resort to a more automatic way of generating the truth files while maintaining a strong confidence in our measurements. To this end, we need to ensure that any automatic way of generating the truth will not bias the results towards any of the systems.

The automatic truth can be generated by simply using the robust parser (see Sec. 2.3) on the orthographic transcriptions which are fairly cheap to acquire. This will generate a semantic interpretation for those utterances that contain fragments that

parse rules defined by the interpretation engine. The human-generated truth is the result of semantically tagging all utterances that didn't yield a full parse by one of the rules for the relevant context.

Figure 1 presents the ROC curves of human and automatic truth generation for the grammar-based and wordSLM systems. We can see that human semantic tagging increases the accuracy substantially, but this increase doesn't seem to favor one system over the other. We are thus led to believe that in our case (very few well defined non-overlapping classes) the automatic truth generation is sufficient. This would not be the case, for example if for a given context a *time* grammar and *number* were active classes. Then, an utterance like "seven" might lead to an erroneous slot being automatically filled while a human tagger (who would have access to the entire dialog) would tag it correctly.

In our experiments, we will use the human semantically tagged truth when available (AppA_MainMenu and AppB_Origin). We have checked that the conclusions of this paper are not altered in any way if the automatic semantically tagged truth had been used for these two contexts.

4 Results and analysis

4.1 Out-of-coverage analysis

Context (#utts)	grammar-based	SLM-based
AppA_MainMenu	1252	1086
AppB_MainMenu	1287	1169
AppB_Origin	1617	1161
AppB_Passenger	492	414
AppB_Time	327	309

Table 2: Number of utterances out-of-coverage for each context.

Coverage is a function of the interpretation engine. We can readily analyze the effect of going from a grammar-based interpretation engine (grammars in Table 1 are in parallel) to the robust approach (rules from grammars in Table 1 are used in robust parsing). This is simply done by running the interpretation engine on the orthographic transcriptions. As expected, the coverage increased. Table 2 shows the number of utterances that didn't fire any rule for each of the interpretation engines. These include noise utterances as described in Table 1. If we remove the noise utterances, going from the grammar-based interpretation to an SLM-based one reduces the out-of-coverage by 31%. This result is interesting because the data was collected from directed-dialog applications which should be heavily guiding the users to the grammar-based system's coverage.

4.2 Results with recognizer

The main results of this paper are found in Figure 2. It presents for grammar-based, wordSLM and classSLM systems the four measurements mentioned in Sec.2.4 for AppA and AppB. We have managed, with proper Viterbi beam settings, to keep in the increase in CPU (grammar-based system $\rightarrow$ SLM-based system) between 0% and 24% relative. We can see that the wordSLM is outperforming the classSLM. The SLM-based systems outperform the grammar-based systems substantially ($\sim 30-50\%$ error rate reduction on most of the confidence domain). The only exception to this is the classSLM in AppA: we will come back to this in Sec. 4.4. This can be interpreted as a different conclusion than those of (Knight et al, 2001; ten Bosch, 2005). The discrepancy can be tied to the fact that the data we are studying comes from a live deployment targeted to the general public. In this case, we can make the hypothesis that a large fraction of the population is composed of naïve users. As mentioned in (ten Bosch, 2005), SLM-based systems perform better than grammar-based ones on that cross-section of the user population.

One might argue that the comparison between the grammar-based and wordSLM systems is unfair because the wordSLM intrinsically records the *a priori* probability that a user says a specific phrase while the grammar-based system studied here didn't benefit from this information. In Sec. 4.4, we will address this and show that *a priori* has a negligible effect in this context.

Note that these impressive results are surprisingly easy to achieve. A simple process could be as follows. An application is developed using grammar-based paradigm. After a limited deployment or pilot with real users, a wordSLM is built from transcribed (orthographic) data from the field. Then the recog-

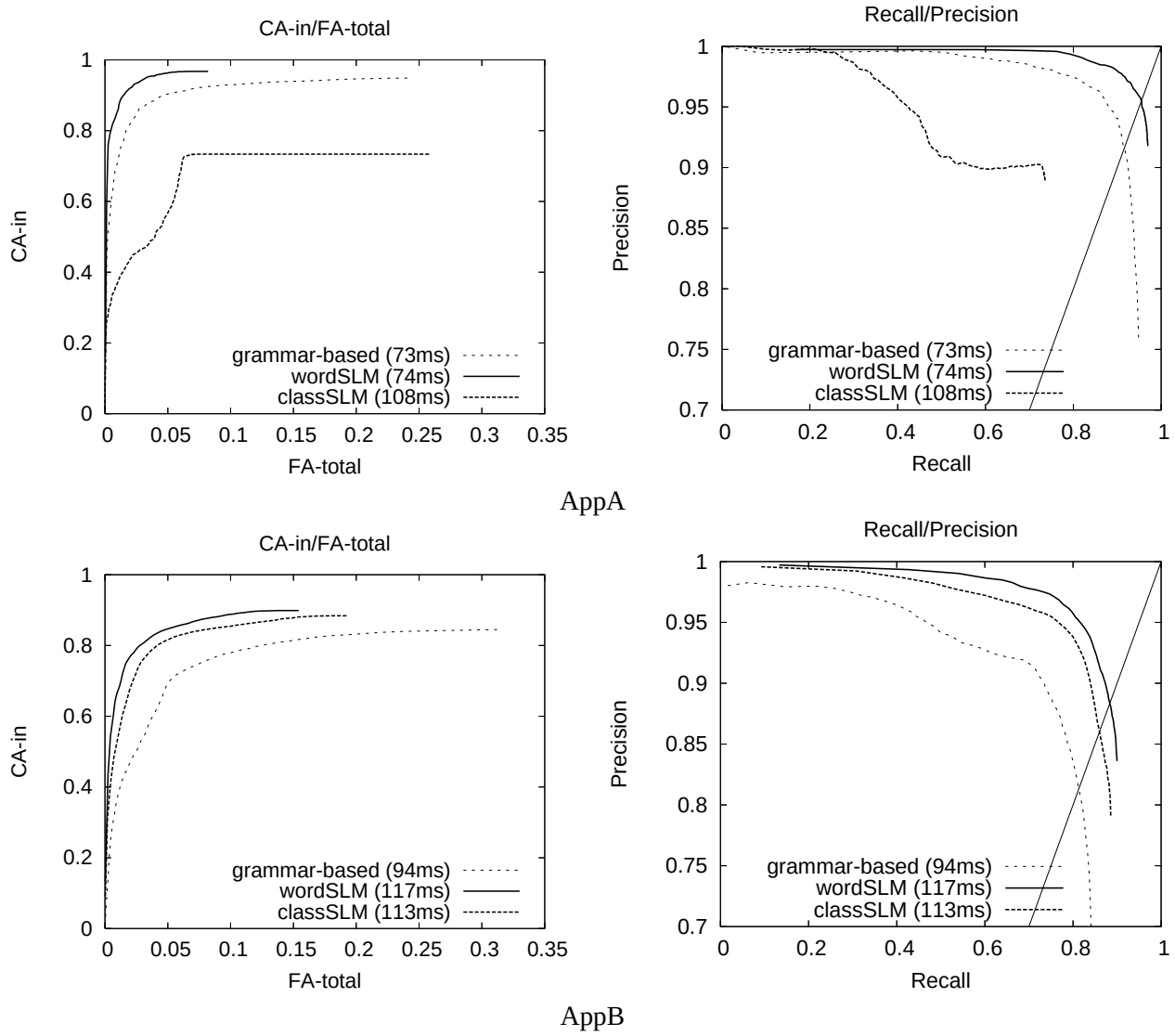

Figure 2: ROC curves for AppA (top) and AppB (bottom). In parenthesis is the average time for the recognition and interpretation.

nition and interpretation engines are upgraded. The grammars built in the early stages of development can largely be re-used as interpretation rules.

4.3 Amount of training data for SLM training

For the remaining Sections, we will use precision and recall for simplicity. We will discuss an extreme case where only a subset of 250 sentences from the standard training set is used to train the SLM. We have run experiments with two contexts: AppA_MainMenu and AppB_Origin. These contexts are useful because a) we have the human-generated truth and b) they represent extremes in the complexity of grammars (see Section 2). On one hand, the grammars for AppA_MainMenu can cover a total of 350 unique sentences while AppB_Origin can cover over 20k. As the amount of training data for the SLMs is reduced from 5000 down to 250 sentences, the accuracy for AppA_MainMenu is only perceptibly degraded for the wordSLM and classSLM systems on the entire confidence domain (not shown here). On the other hand, in the case of the more complex grammar (class), it is a different story which highlights a second regime. For AppB_Origin, the precision and recall curve is presented on Figure 3. In the case of classSLM (left),

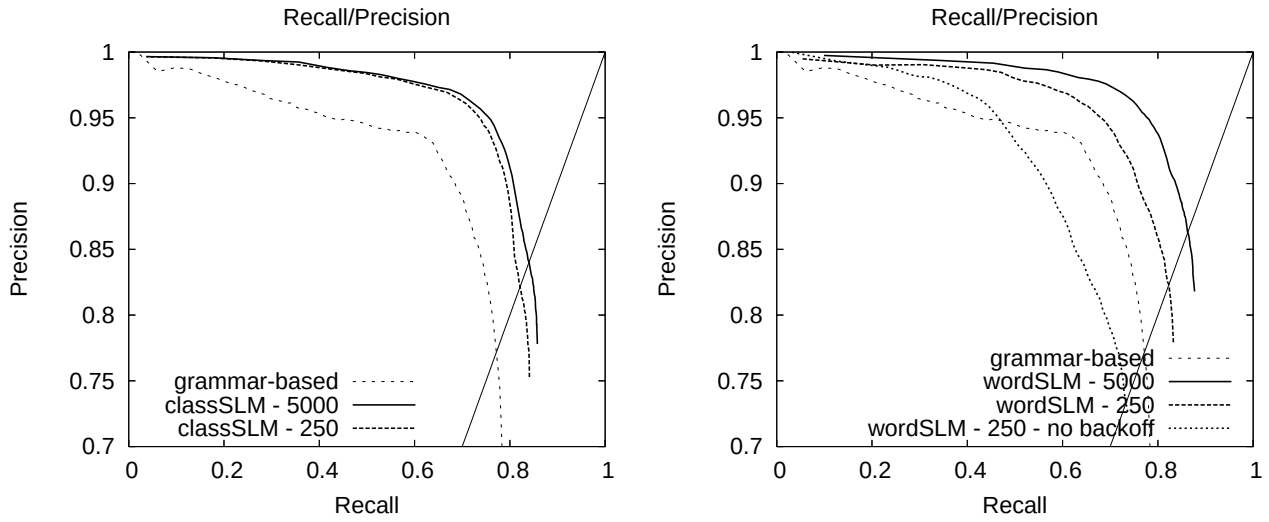

Figure 3: Precision and recall for the AppB_Origin context as the amount of training data for the SLMs is reduced. On the left, classSLM systems are presented; on the right it is the wordSLM.

even with very little training data, the accuracy is far better than the grammar-based system and only slightly degraded by reducing the size of the training set. In the case of wordSLM (right), we can still see that the accuracy is better than the grammar-based system (refer to "wordSLM - 250" on the graph), but the reduction of training data has a much more visible effect. If we remove the sentences that were drawn from the grammar-based system's coverage (backoff - see Sec. 2.3), we can see that the drop in accuracy is even more dramatic.

4.4 Coverage of interpretation rules and priors

As seen in Sec. 4.2, the classSLM results for AppA are disappointing. They, however, shed some light on two caveats of the robust approach described here. The first caveat is the coverage of the interpretation rules. As described in Sec. 2, the SLM-based systems' training sets and interpretation rules (grammars from Table 1) were built in isolation. This can have a dramatic effect: after error analysis of the classSLM system's results, we noticed a large fraction of errors for which the recognized string was a close (semantically identical) variant of a rule in the interpretation engine ("cancellations" vs "cancellation"). In response, we implemented a simple tool to increase the coverage of the grammars (and hence the coverage of the interpretation rules) using the list of words seen in the training set. The criteria for selection is based on common stem with a word in the grammar.

The second caveat is based on fact that the classSLM suffers from a lack of prior information once the decoding process enters a specific class since the grammars (class) do not contain priors. The wordSLM benefits from the full prior information all along the search. We have solved this by training a small wordSLM **within** each grammar (class): for each grammar, the training set for the small wordSLM is composed of the set of fragments from all utterances in the main training set that fire that specific rule. Note that this represents a way to have the grammar-based and SLM-based systems share a common *training set* (Rayner et al, 2005).

In Figure 4, we show the effect of increasing the coverage and adding priors in the grammars. The first conclusion comes in comparing the grammar-based results with and without increased coverage (enhanced+priors in figure) and priors. We see that the ROC curves are one on top of the other. The only differences are: a) at low confidence where the enhanced+priors version shows better precision, and b) the CPU consumption is greatly reduced (73ms $\rightarrow$ 52ms). When the enhanced+priors version of the grammars (for classes and interpretation rules) is used in the context of the classSLM system, we can see that there is a huge improvement in the accuracy: this shows the importance of keeping the SLM

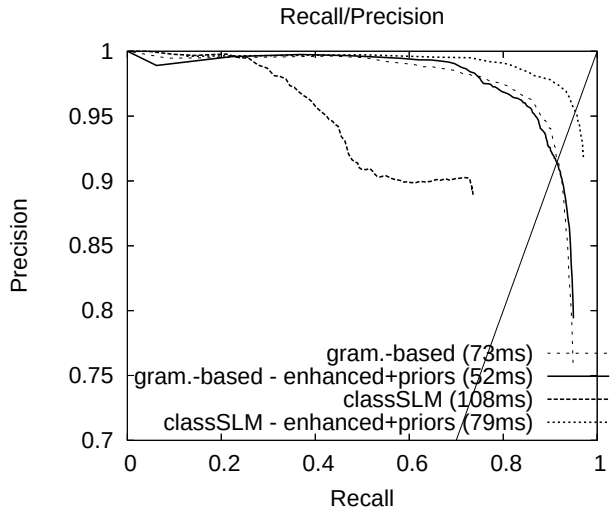

Figure 4: ROC curves for AppA showing the effect of increasing the grammar coverage and adding prior information in the grammars.

and interpretation rules in-sync. The final classSLM ROC curve (Figure 4) is now comparable with its wordSLM counter-part (Figure 2 upper right graph).

5 Conclusion

We have demonstrated in this paper that grammar-based systems for commercially deployed directed dialog applications targeted at the general public can be improved substantially by using SLMs with robust parsing. This conclusion is different than (Rayner et al, 2005) and can be attributed to that fact that the general public is likely composed of a large portion of naïve users. We have sketched a very simple process to upgrade an application from using a grammar-based approach to a robust approach when in-service data and interpretation rules (grammars) are available. We have also shown that only a very small amount of data is necessary to train the SLMs (Knight et al, 2001). Class-based SLMs should be favored in the case where the amount of training data is low while word-based SLMs should be used when enough training data is available. In the case of non-overlapping classes, we have demonstrated the soundness of automatically generated semantic truth.

6 Acknowledgements

The author would like to acknowledge the helpful discussions with M. Fanty, R. Tremblay, R. Lacouture and K. Govindarajan during this project.

References

W. Ward. 1990. The CMU Air Travel Information Service: Understanding spontaneous speech . *Proc. of the Speech and Natural Language Workshop*, Hidden Valley PA, pp. 127–129.

A.L. Gorin, B.A. Parker, R.M. Sachs and J.G. Wilpon. 1997. How may I help you?. *Speech Communications*, 23(1):113–127.

C. Hemphill, J. Godfrey and G. Doddington. 1990. The ATIS spoken language systems and pilot corpus. *Proc. of the Speech and Natural Language Workshop*, Hidden Valley PA, pp. 96–101.

S. Knight, G. Gorrell, M. Rayner, D. Milward, R. Koeling and I. Lewin. 2001. Comparing grammar-based and robust approaches to speech understanding: a case study. *Proc. of EuroSpeech*.

M. Rayner, P. Bouillon, N. Chatzichrisafis, B.A. Hockey, M. Santaholma, M. Starlander, H. Isahara, K. Kanzaki and Y. Nakao. 2005. A methodology for comparing grammar-based and robust approaches to speech understanding. *Proc. of EuroSpeech*.

L. ten Bosch. 2005. Improving out-of-coverage language modelling in a multimodal dialogue system using small training sets. *Proc. of EuroSpeech*.

M. Balakrishna, C. Cerovic, D. Moldovan and E. Cave. 2006. Automatic generation of statistical language models for interactive voice response applications. *Proc. of ICSLP*.

J. Gillett and W. Ward. 1998. A language model combining tri-grams and stochastic context-free grammars. *Proc. of ICSLP*.

F. Jelinek. 1990. Readings in speech recognition, Edited by A. Waibel and K.-F. Lee , pp. 450-506. Morgan Kaufmann, Los Altos.

W. Xu and A. Rudnicky. 2000. Language modeling for dialog system. *Proc. of ICSLP*.

V. Goel and R. Gopinath. 2006. On designing context sensitive language models for spoken dialog systems. *Proc. of ICSLP*.