D1.3: SLM generation in the Grammatical Framework

Karl Weilhammer, Rebecca Jonson, Aarne Ranta, Matt Stuttle and Steve Young

Distribution: Public

TALK
Talk and Look: Tools for Ambient Linguistic Knowledge
IST-507802 Deliverable 1.3

February 10, 2006

Project funded by the European Community under the Sixth Framework Programme for Research and Technological Development

The deliverable identification sheet is to be found on the reverse of this page.

Project ref. no.	IST-507802
Project acronym	TALK
Project full title	Talk and Look: Tools for Ambient Linguistic Knowledge
Instrument	STREP
Thematic Priority	Information Society Technologies
Start date / duration	01 January 2004 / 36 Months

Security	Public
Contractual date of delivery	M24 = December 2005
Actual date of delivery	February 10, 2006
Deliverable number	1.3
Deliverable title	D1.3: SLM generation in the Grammatical Framework
Type	Report
Status & version	Final 1.0
Number of pages	38 (excluding front matter)
Contributing WP	1
WP/Task responsible	UCAM
Other contributors	UGOT
Author(s)	Karl Weilhammer, Rebecca Jonson, Aarne Ranta, Matt Stuttle and Steve Young
EC Project Officer	Evangelia Markidou (Anne Bajart)
Keywords	Grammar, Language Model, Speech Recognition

The partners in TALK are:

Saarland University	USAAR
University of Edinburgh HCRC	UEDIN
University of Gothenburg	UGOT
University of Cambridge	UCAM
University of Seville	USE
Deutches Forschungszentrum fur Künstliche Intelligenz	DFKI
Linguamatics	LING
BMW Forschung und Technik GmbH	BMW
Robert Bosch GmbH	BOSCH

For copies of reports, updates on project activities and other TALK-related information, contact:

The TALK Project Co-ordinator
Prof. Manfred Pinkal
Computerlinguistik
Fachrichtung 4.7 Allgemeine Linguistik
Postfach 15 11 50
66041 Saarbrücken, Germany
pinkal@coli.uni-sb.de
Phone +49 (681) 302-4343 - Fax +49 (681) 302-4351

Copies of reports and other material can also be accessed via the project's administration homepage, `http://www.talk-project.org`

Contents

Summary

In this deliverable, we explore methods to create statistical language models (SLMs) by generating corpora from application grammars using the Grammatical Framework. We create statistical language models directly from our interpretation grammars and compare recognition performance of these models against speech recognition grammars (e.g. Nuance grammars and HTK grammars) compiled from the same interpretation grammar. The results show an important WER reduction for the SLMs modelling the language of the interpretation grammar while maintaining a good in-grammar performance in comparison with the Speech Recognition Grammars. We show that interpolation of SLMs trained from grammar generated corpora with SLMs trained on Wizard Of Oz corpora or standard speech and language corpora perform even better. We also investigate how domain relevant data can be extracted from large diverse corpora and demonstrate that sentence selection based on in-domain vocabulary or perplexity filtering can improve statistical language models. All experiments were carried out by UCAM on an English tourist information task and by UGOT on a Swedish MP3 player interface. In both cases recognisers run with the respective best statistical language models had about half the word error rate compared to recognisers run with a grammar network. In a last experiment we show that using dialogue move specific SLMs derived from a GF grammar gain some improvement vs. using a general grammar-based SLM.

Chapter 1

Introduction

Ideally when building spoken dialogue systems we would like to use a corpus of transcribed dialogues, corresponding to the specific task of the dialogue system, in order to build a statistical language model (SLM). However, it is rarely the case that such a corpus exists in an early stage of development. Collecting and transcribing such a corpus is very time-consuming, and delays the process of building the actual dialogue system. Since a lot of assumptions about how users would interact with a dialogue system are involved in such a data collection, often carried out under the Wizard Of Oz paradigm, it is not guaranteed that such a corpus will approximate the actual user behaviour well. In this document we will investigate methods to speed up the process. However, when the system is up and running it is possible to collect real data that can be used to improve the model.

1.1 Recognition Grammars and grammar generated statistical language models

The standard approach is to compile the interpretation grammar into a speech recognition grammar as the Gemini compiler does [RHJ+00]. In this way it is assured that the linguistic coverage of speech recognition and interpretation are kept in sync. With this approach all sentences that are recognised are guaranteed to have a parse in the interpretation grammar. Within TALK the Grammatical Framework (GF) has been extended with a facility that compiles GF grammars into Nuance Speech grammars and HTK grammars [Ran05].

Another approach that has become more popular, both in dialogue systems and dictation applications, is to first write an interpretation grammar and from that generate an artificial corpus which is used as training corpus for a statistical language model [RLBE00, PSB01, FLK01]. These corpora have a number of problems:

- The richness of phenomena in these corpora depends on how well the grammar writer captured all different ways to address the system.
- Sentences are either randomly generated or all possible sentences are listed. In both cases only rough approximations to real frequencies are produced.

Speech recognition for commercial dialogue systems has focused on grammar-based approaches despite

the fact that statistical language models seem to have a better overall performance [GLR02]. The reason for this is probably, that it is quite time-consuming to collect corpora for training the language models compared with the more rapid and straightforward development of speech recognition grammars. However, statistical language models are more robust, can handle out-of-coverage input, perform better in difficult conditions and seem to work better for naive users (see [KGR+01]) while speech recognition grammars are limited in their coverage depending on how well grammar writers succeed in predicting what users may say [HAH01]. Nevertheless, as grammars only output phrases that can be interpreted their output makes the following interpretation task easier than with the unpredictable output from a statistical language model (especially if the speech recognition grammar has been compiled from the interpretation grammar and these both are in sync). In addition, the grammar-based approach in the experiments reported in [KGR+01] outperforms the language model approach on semantic error rate on in-coverage data.

1.2 Using large corpora to adapt language models

However carefully designed, grammar generated corpora only cover a limited number of ways to address the system and only rough approximations to real n-gram frequencies are produced. One possibility to overcome these problems is to mix them with spontaneous speech transcriptions or topic related text collections.

Often large standard corpora contain a number of topics and speaking styles and it makes sense to select only those sentences from them which are in the actual scope of the domain of the dialogue system. This can be done by hand or using filtering techniques.

1.3 Software

For both UCAM and UGOT one of the major infrastructure issues in TALK is setting up a speech recogniser for the domains addressed by this project. UGOT approaches this starting from a GF interpretation grammar developed for their GoDiS System. Whereas UCAM's approach is centered around the ATK Speech recogniser. In pursuing this agenda a number of new features were included in GF and ATK.

- compliers to Nuance and ATK/HTK grammars (GF),
- random generation of sentences (GF),
- generation of all possible sentences of a grammar (GF),
- Support for class based statistical language models (ATK).

Download information for the latest versions of GF and ATK is provided in sections 2.2 and 2.4 respectively.

1.4 Structure of the deliverable

In this deliverable, we will consider several different language models based on different corpora generated from GF interpretation grammars and compare their recognition performance with the baseline: a Nuance or HTK Grammar compiled from the same interpretation grammar.

In chapter 2 we describe how GF can be used for generating sentences from grammars, list all language modelling toolkits and shortly describe how we used them for our experiments. Chapter 3 describes different tasks and compares recognition results for Nuance and HTK recognition grammars and statistical language models built from grammar generated corpora. Chapter 4 contains a section on choosing relevant data from large speech corpora through perplexity filtering and sentence selection based on in-domain vocabulary. The described methods are then applied to various tasks and recognition results are given. We describe recognition experiments with dialogue move specific models generated from GF in chapter 5. Finally, we review the main conclusions and discuss future work in chapter 6.

Chapter 2

Software for generating statistical language models from GF

A method for generation of statistical language models (SLMs) from GF grammars has been explored in task 1.3. The method consists of two parts: generating a corpus from a grammar and generating SLMs from that corpus. The method is described part-wise below.

2.1 Generating a corpus with GF

For any GF grammar, the command `generate_trees` (short name `(gt)`) can be used to generate all syntax trees with certain criteria that can be defined by flags and arguments. The most important flags are:

1. `-depth` generate to this depth (default 3)
2. `-cat` generate in this category
3. `-lang` use the abstract syntax of this grammar
4. `-number` generate (at most) this number of trees

Examples:

1. `gt -depth=10 -cat=NP` – generate all NP's to depth 10
2. `gt (PredVP ? (NegVG ?))` – generate all trees of this form
3. `gt -cat=S -tr | l` – generate S's and then linearise

The last-mentioned command is the one typically used to generate a corpus of sentences. It will actually generate a tree bank, since the trees are saved due to the `-tr` flag, and the trees can be paired with the sentences.

The algorithm assumes that the abstract syntax is context-free, i.e. has no dependent types or higher-order functions. The result is overgeneration, if dependent types are present. However, it is possible to filter the results through a type checker:

```
gt -cat=S | pt -transform=solve | l
```

in which case only semantically meaningful examples are left. This is what was done in the MP3 domain to guarantee that actions and their objects were only combined in meaningful ways (see section3.1). It is also possible to perform random generation with equally distributed or biased probabilities.

GF also includes the facility of compiling Nuance and HTK recognition grammars from a GF interpretation grammar (see TALK deliverable D1.1). In this deliverable we will compare the language models we obtain with the correspondent speech recognition grammars.

From a GF interpretation grammar we can generate not only all sentences but also all dialogue moves. As dialogue moves are seen as concrete syntax in GF it is possible to generate all abstract GF-trees and finally linearise every tree to both Swedish utterances and dialogue moves. In this way, GF can be used to obtain a corpus with each dialogue move and the utterances corresponding with it. This has been done in chapter 5 to generate dialogue move specific language models.

2.2 Grammar formats

Unfortunately ATK and Nuance use different formats for recognition grammars or recognition networks. Therefore GF was extended, such that it can now convert GF-grammars into both Nuance recognition grammars and HTK grammars[1]. The most recent version of GF supporting these features can be downloaded from the GF homepage:

Grammatical Framework

Description: Grammar Toolkit with converters to ATK and Nuance recognition grammars. The latest version can be checked out from a Darcs archive.

Homepage: `http://www.cs.chalmers.se/~aarne/GF/`

Download: Last stable release at Sorceforge:
`http://sourceforge.net/project/showfiles.php?group_id=132285`
Latest sources and documents at the publicly accessible Darcs repository:
`http://www.cs.chalmers.se/Cs/Research/Language-technology/darcs/GF/doc/darcs.html`

2.3 Language modelling toolkits

Language modelling is a standard task when building a speech recogniser. Therefore a few toolkits tackling this problem have been made available by different research groups. Most major language modelling techniques are supported by at least one of these toolkits, but each of them offers certain unique features.

[1] An example for a HTK grammar can be found in section 3.2.2

SRI language modelling toolkit

Description: Very general toolkit, supports a huge number of backup and smoothing schemes (e.g. Kneser Ney smoothing is supported.) and can convert SLMs from ARPA format into Nuance format.

Homepage: `http://www.speech.sri.com/projects/srilm/`

Download: `http://www.speech.sri.com/projects/srilm/download.html`

CMU-Cambridge language modelling toolkit

Description: In this investigation only the "interpolate" command is used to automatically determine the weights λ for the linear interpolation of two language models using a variant of the expectation maximisation (EM) algorithm. Both other tools apparently lack this feature.

Homepage/Download: `http://mi.eng.cam.ac.uk/~prc14/toolkit.html`

HTK language modelling tools

Description: The HLM tools are packaged with HTK. They offer a number of different smoothing and backoff methods. In this investigation they were mainly used to analyse perplexities.

Homepage: `http://htk.eng.cam.ac.uk/`

Download: `http://htk.eng.cam.ac.uk/download.shtml`

Because of their different features all three language modelling toolkits were used in this investigation.

2.4 Real time speech recongisers

In this investigation two speech recognisers were used. UCAM used their own ATK software. UGOT used the Nuance speech recogniser.

Nuance

Description: The Nunace speech recogniser is a commercial product sold by Nuance Communications, Inc.

Homepage: `http://www.nuance.com/nuancerecognition/`

Application Toolkit for HTK (ATK)

Description: ATK is an API designed to facilitate building experimental applications for HTK. It consists of a C++ layer sitting on top of the standard HTK libraries. This allows novel recognisers built using customised versions of HTK to be compiled with ATK and then tested in working systems.

Homepage: `http://mi.eng.cam.ac.uk/~sjy/software.htm`

Download: `http://htk.eng.cam.ac.uk/develop/atk.shtml` (Please register as HTK user first.)

2.5 ARPA language model format

Fortunately there is one common standard for n-gram backoff language model files: The so-called ARPA format[2]. The structure of such an ASCII file is displayed below. The variables `n1` ... `nN` in the `data` section hold the numbers of n-grams of a certain context. In the `N-grams:` sections the n-gram values are stored. The first item `p` is a float representing the probability and `[bow]` the backoff weight (optional). Both numbers are logarithms with respect to the base 10. The words `w1 ... wN` making up the actual n-gram are printed between these numbers.

```
\data\
  ngram 1=n1
  ngram 2=n2
  ...
  ngram N=nN
\1-grams:
  p w [bow]
  ...

\2-grams:
  p w1 w2 [bow]
  ...

\N-grams:
  p w1 ... wN
  ...
\end\
```

ARPA format is the default language model output format of all three language modelling toolkits. Both the ATK and HTK speech recogniser can read this format. SRILM provides a tool to convert SLMs given in ARPA format into a format that the Nuance recogniser can read.

2.6 Using GF together with language modelling tools

All of the previously mentioned tools interact with the user via the command line. This provides a lot of freedom for experimenting directly with them and makes it easy to write scripts for automatic training and

[2] According to the SRILM manual page describing this standard, it was developed by Doug Paul at MIT Lincoln Labs.

testing. In the case of recognition grammars the interface between GF and both speech recognisers is very simple:

- *Nuance:* The nuance grammar directly generated from GF
- *ATK:* The HTK grammar generated by GF which has to be converted in a HTK grammar network before it can be used in a recogniser.

For statistical language models the following steps are necessary:

1. Generate corpus from GF
2. Generate a SLM in ARPA format by applying a combination of tools on the corpus
3. Run speech recogniser
    - When using ATK:
      Run speech recogniser with ARPA language model file
    - When using Nuance:
      Convert ARPA language model file into Nuance format using SRILM
      Run speech recogniser with Nuance format language model file

2.7 Conclusion

By including a number of new features in GF and ATK both toolkits were extended in such a way, that it is now easy to convert GF grammars into recognition grammars and to generate corpora for training statistical language models. Both software packages can be downloaded for free from UGOT and UCAM websites. They harmonise very well with language modelling software that is already available on the Internet thus providing a useful contribution to the research community.

Chapter 3

Comparing the performance of recognition grammars and statistical language models

In this chapter we present work carried out in Gothenburg and Cambridge. The close collaboration has resulted in a useful knowledge transfer between both sites, where Gothenburg contributed expertise in grammar design and Cambridge provided guidance for speech recognition experiments and statistical language modelling. Parts of the work reported in this Chapter will be presented in [Jon06].

First we report work from Gothenburg on Swedish grammars and statistical language models using a Nuance speech recogniser for a GoDiS MP3 player application. In the second part of this chapter we describe research done in Cambridge on English models for a tourist information application based on an ATK speech recogniser and a Dipper dialogue manager.

3.1 MP3 Domain

3.1.1 The MP3 corpus

The MP3 player application (DJGoDiS) is a multimodal interface to a graphical MP3 player and has been built with GoDiS and TrindiKit (see status report T1.6s2). The interpretation and generation grammars are written with the GF grammar formalism. DJGoDiS allows voice control of an audio player. The user can among other things change settings, choose stations or songs to play or create playlists. The current version of the GoDiS MP3 application works in speech and text mode in English and Swedish. For the work presented here we have used the Swedish version.

The interpretation grammar for the domain, written in GF, translates user utterances to dialogue moves and thereby holds all possible interpretations of user utterances. We used GF's facilities to generate a corpus in Swedish representing most of the content in the grammar consisting of all possible meaningful utterances to a certain depth of analysis. As the current grammar is under development it is not complete and some linguistic structures are missing. The grammar is written on the phrase level accepting spoken language utterances such as e.g. "next, please". The corpus of possible user utterances resulted in around 320 000 user utterances (about 3 million words) corresponding to a vocabulary of only 301 words. The database of songs and artists in this first version of the application is limited to 60 Swedish songs, 60 Swedish artists, 3 albums and 3 radio stations. The vocabulary may seem small if you consider the number of songs and

artists included, but the small size is due to a huge overlap of words in songs and artists as pronouns (such as *Jag* (*I*) and *Du* (*You*)) and articles (such as *Det* (*The*)) are very common. This corpus is very domain specific as it includes many artist names, songs and radio stations that often consist of rare words. It is also very repetitive covering all combinations of songs and artists in utterances such as "I want to listen to Mamma Mia by Abba". However, all utterances in the corpus occur exactly once.

3.1.2 The GSLC corpus

The Gothenburg Spoken Language (GSLC) corpus consists of transcribed Swedish spoken language from different social activities such as auctions, phone calls, meetings, lectures and task-oriented dialogue [All99]. To be able to use the GSLC corpus for language modelling it was pre-processed to remove annotations and all non-alphabetic characters. The final GSLC corpus consisted of a corpus of about 1,300,000 words with a vocabulary of almost 50,000 words.

3.1.3 The Swedish newspaper corpus

We have also used a corpus consisting of a collection of Swedish newspaper texts (`GNC`) of 397 million words to get hold of a larger corpus [1]. This corpus is part of a collection of written texts that has been collected at the Department of Linguistics at Göteborg University. The corpus consists of newspaper text from several Swedish newspapers (including Göteborgsposten, the Gothenburg Newspaper) collected mainly during the second half of the 90s.

3.1.4 The MP3 SLM

The first model was generated directly from the MP3 corpus we got from the GF grammar. This simple LM (named `MP3GFLM`) has the same vocabulary as the Nuance Grammar and models the same language as the GF grammar. This model was chosen to see if we could increase flexibility and robustness in such a simple way while maintaining in-grammar performance.

We also created two other simple LMs: a class-based one (with the classes `Song`, `Artist` and `Radiostation`) and a model based on a variant of the MP3 corpus where the utterances in which songs and artists co-occur would only match real artist-song pairs (i.e. including some music knowledge in the model).

These three SLMs were the three basic MP3 models considered although we only report results for the `MP3GFLM` in this article (the class-based model gave a slightly worse result and the other a slightly better result).

In addition to this we used our general corpora to produce two different models: `GSLCLM` from the GSLC corpus and `NewsLM` from the newspaper corpus.

3.1.5 Interpolating the GSLC corpus and the MP3 corpus

A technique used in language modelling to combine different SLMs is linear interpolation [JM80]. This is often used when the domain corpus is too small and a bigger corpus is available. There have been many attempts at combining domain corpora with news corpora, as this has been the biggest type of corpus

[1] Made available by Leif Grönqvist, Dept. of Linguistics, Gothenburg

available and this has given slightly better models [JDB98, Ros00b]. Linear interpolation has also been used when building state dependent models by combining the state models with a general domain model [XR00, SFLK+02].

Rosenfeld [Ros00b] argues that a little more domain corpus is always better than a lot more training data outside the domain. Many of these interpolation experiments have been carried out by adding news text, i.e. written language. In this experiment we are going to interpolate our domain model (`MP3GFLM`) with a spoken language corpus, the GSLC, to see if this improves perplexity and recognition rates. As the MP3 corpus is generated from a grammar without probabilities this is hopefully a way to obtain better and more realistic estimates on words and word sequences. Ideally, what we would like to capture from the GSLC corpus is language that is invariant from domain to domain. However, Rosenfeld [Ros00a] is quite pessimistic about this, arguing that this is not possible with today's interpolation methods. The GSLC corpus is also quite small.

The interpolation was carried out with the SRILM toolkit [Sto02] based on equation 3.1.

$$MixGSLCMP3GF = \lambda * MP3GFLM + (1-\lambda) * GSLCLM \tag{3.1}$$

The optimal lambda weight was estimated to 0.65 with the SRILM toolkit using the development test set.

3.1.6 Interpolating the newspaper corpus and the MP3 corpus

We also created another model in the same way as above by interpolating the news corpus with our simplest model.

$$MixNewsMP3GF = \lambda * MP3GFLM + (1-\lambda) * NewsLM \tag{3.2}$$

Choice of vocabulary

The resulting mixed models have a huge vocabulary as the GSLC corpus and the newspaper corpus include thousands of words. This is not a convenient size for recognition as it will affect accuracy and speed. Therefore we tried to find an optimal vocabulary combining the small MP3 vocabulary of around 300 words with a smaller part of the GSLC vocabulary and the newspaper vocabulary.

In a first experiment we used the CMU toolkit [CR97] to obtain the most frequent words of the GSLC corpus. We selected three different sizes of the most frequent words: 300, 500 and 750. These different vocabularies were merged with the MP3 vocabulary resulting in three mixed vocabularies of 500, 700 and 900 words. The overlap was quite low (73 words for the smallest vocabulary) showing the peculiarity of the MP3 domain. We used these vocabularies to generate three new versions of `MixGSLCMP3GF`. After testing we decided on the 500 word vocabulary for the mixed GSLC model.

In a second test we created a vocabulary that was a mixture of the most frequent words in the GSLC corpus, the most frequent ones in the newspaper corpus, the vocabulary used for extracting domain data and the small MP3 vocabulary. This resulted in a vocabulary of 1153 words. The mixed models obtained from the news models have all used this mixed vocabulary in the tests.

3.1.7 The Test Corpus

To collect a test set we asked students to describe how they would address a speech-enabled MP3 player by writing Nuance grammars that would cover the domain and its functionality. Another group of students evaluated these grammars by recording utterances they thought they would say to an MP3 player. One of the Nuance grammars was used to create a development test set by generating a corpus of 1500 utterances from it. The corpus generated from another grammar written by some other students was used as evaluation test set. Added to the evaluation test set were the transcriptions of the recordings made by the third group of students that evaluated both grammars. This resulted in a evaluation test set of 1700 utterances.

The recording test set was made up partly of the students' recordings. Additional recordings were carried out by letting people at our lab record randomly chosen utterances from the evaluation test set. We also had a demo running for a short time to collect user interactions at a demo session. The final test set included 500 recorded utterances from 26 persons. This test set has been used to compare recognition performance between the different models under consideration.

The recording test set is just an approximation to the real task and conditions as the students only capture how they think they would act in an MP3 task. Their actual interaction in a real dialogue situation may differ considerably so ideally, we would want more recordings from dialogue system interactions which at the moment constitutes only a fifth of the test set. However, until we can collect more recordings we will have to rely on this approximation.

In addition to the recorded evaluation test set, a second set of recordings was created covering only in-grammar utterances by randomly generating a test set of 300 utterances from the GF grammar. These were recorded by 8 persons. This test set was used to contrast with a comparison of in-grammar recognition performance.

3.1.8 Perplexity measures

The 8 LMs were evaluated by measuring perplexity with the tools SRI provides on the evaluation test set of 1700 utterances.

Table 3.1: *Perplexity for the different LMs.*

LM	Perplexity
MP3GFLM	587
GSLCLM	492
NewsLM	386
MixGSLCMP3GF	61
MixNewsMP3GF	78

In Table 3.1 we can see a dramatic perplexity reduction with the mixed models compared to the simplest of our models the `MP3GFLM`. Surprisingly, the `GSLCLM` models the test set better than the `MP3GFLM` which indicates that our MP3 grammar is too restricted and differs considerably from the students' grammars.

Lower perplexity does not necessarily mean lower word error rates and the relation between these two measures is not very clear. One of the reasons that language model complexity does not measure the recognition task complexity is that language models do not take into account acoustic confusability [HAH01].

According to Rosenfeld [Ros00b], a perplexity reduction of 5% is usually practically not significant, 10-20% is noteworthy and a perplexity reduction of 30% or more is quite significant. The above results of the mixed models could then mean an improvement in word error rate over the baseline model MP3GFLM. This has been tested and is reported in the next section. In addition, we wanted to test if we could reduce word error rate using our simple language model opposed to the Nuance grammar (MP3NuanceGr) which is our recognition baseline.

3.1.9 Recognition rates

The 8 LMs under consideration were converted with the SRILM toolkit into a format that Nuance accepts and then compiled into recognition packages. These were evaluated with Nuance [Nua05] on the recorded evaluation test set of 500 utterances (26 speakers). Table 3.2 presents word error rates (WER) and in parenthesis N-Best (N=10) WER for the models under consideration and for the Nuance Grammar.

Table 3.2: *Word error rates(WER) for the recording test set*

LM	WER(NBest)
MP3GFLM	37.11 (29.48)
GSLCLM	83.04 (71.51)
NewsLM	61.62 (49.53)
MixGSLCMP3GF	34.58 (22.68)
MixNewsMP3GF	38.00 (27.37)
MP3NuanceGr	59.37 (53.19)

As seen, our simple statistical language model, MP3GFLM, improves recognition performance considerably compared with the Nuance grammar baseline (MP3NuanceGr) showing a much more robust behaviour to the data. Remember that these two models have the same vocabulary and are both derived from the same GF interpretation grammar. However the flexibility of the language model gives a relative improvement of 37% over the Nuance grammar. The models giving the best results are the models interpolated with the GSLC corpus and the domain news corpus in different ways which at best gives a relative reduction in WER of 8% in comparison with MP3GFLM and 43% compared with the baseline. It is interesting to see that the simple way we used to create a domain specific newspaper corpus gives a model that better fits our data than the original much larger newspaper corpus.

3.1.10 In-grammar recognition rates

To contrast the word error rate performance with in-grammar utterances i.e. utterances that the original GF interpretation grammar covers, we carried out a second evaluation with the in-grammar recordings. We also used Nuance's parsing tool to extract the utterances that were in-grammar from the recorded evaluation test set. These few recordings (5%) were added to the in-grammar test set. The results of the second recognition experiment are reported in Table 3.3.

The in-grammar results reveal an increase in WER for all the language models in comparison to the baseline MP3NuanceGr. However, the simplest model (MP3GFLM), modelling the language of the grammar, do not show any greater reduction in recognition performance.

Table 3.3: *WER on the in-grammar test set*

LM	WER (NBest)
MP3GFLM	4.95 (2.04)
GSLCLM	78.07 (64.15)
NewsLM	48.03 (36.64)
MixGSLCMP3GF	14.23 (6,29)
MixNewsMP3GF	18.63 (10.22)
MP3NuanceGr	3.69 (1.49)

3.1.11 Discussion of results

The word error rates obtained for the best models show a relative improvement over the Nuance grammar of 40%. The most interesting result is that the simplest of our models, modelling the same language as the Nuance grammar, gives such an important gain in performance that it lowers WER by 22%.

We used the Chi-square test of significance to statistically compare the results with the results of the Nuance grammar showing that the differences of WER of the statistical language models in comparison with the baseline are all significant on the p=0.05 significance level. However, the Chi-square test also points out that the difference of WER for in-grammar utterances of the Nuance grammar and the `MP3GFLM` is significant on the p=0.05 level. This means that all the SLMs significantly outperform the baseline i.e. the Nuance Grammar `MP3NuanceGr` on the evaluation test set (being mostly out-of-coverage) but that `MP3NuanceGr` outperforms the SLMs on in-grammar utterances. Although `MP3NuanceGr` performs significantly better than the SLMs on in-grammar utterances the difference is small compared to `MP3GFLM`. As we expect to encounter both in-grammar and out-of-coverage utterances in our system the `MP3GFLM` seems to be our best choice as it outperforms the baseline (`MP3NuanceGr`) overall.

However, as the reader may have noticed, the word error rates are quite high, which is partly due to a totally independent test set with some of out-of-vocabulary words (4-10% OOVs depending on the vocabulary used) indicating that domain language grammar writing is very subjective. The students have captured a quite different language for the same domain and functionality. This shows the risk of a hand-tailored domain grammar and the difficulty of predicting what users may say. In addition, a fair test of the model would be to measure concept error rate or more specifically dialogue move error rate (i.e. both 'yes' and 'yeah' correspond to the same dialogue move `answer(yes)`). A closer look at the `MP3GFLM` results give a hint that in many cases the transcription reference and the recognition hypothesis hold the same semantic content in the domain (e.g. confusing the Swedish prepositions 'i' (into) and 'till' (to) which are both used when referring to the playlist). It was manually estimated that 53% of the recognition hypotheses could be considered as correct in this way opposed to the 65% Sentence Error Rate (SER) that the automatic evaluation gave. This implies that the evaluation carried out is not strictly fair considering the possible task improvement.

The N-Best results indicate that it could be worth putting effort on re-ranking the N-Best lists as both WER and SER of the N-Best candidates are considerably lower. This could ideally give us a reduction in SER of 10% and, considering dialogue move error rate, perhaps even more. More or less advanced post-process methods have been used to analyse and decide on the best choice from the N-Best list. Several different re-ranking methods have been proposed that show how recognition rates can be improved by letting external processes do the top N ranking and not the recogniser [CR01, vNBKN99, QAM$^+$02]. However, the way

that seems most appealing is how [GL04] (as part of the TALK project) and [KW01] re-rank N-Best lists based on dialogue context achieving a considerable improvement in recognition performance. We are considering basing our re-ranking on the information held in the dialogue information state in GoDiS, on knowledge of what is going on in the graphical interface and on dialogue moves in the list that seem appropriate to the context. In this way we can take advantage of what the dialogue system knows about the current situation.

3.2 In-Car Tourist Information Domain

The so called "In-Car" domain is a tourist information task, where the driver of a car can ask for information about hotels, bars and restaurants in an invented town. This task was realised only in English. The aim of this investigation is to build the speech recognition component for a dialogue system with minimal effort from scratch. Both the acoustic models described in section 3.2.3 and the best language model described in this document are used in the Baseline System detailed in Deliverable 4.2 [LGS05] and [LGHS06].

3.2.1 Test Data Collection of "Example" Utterances

Although we want to minimise our efforts we still need a small corpus for training and testing. We asked 9 co-researchers who were familiar with the task to submit a set of 10 "example" interactions with the system and a number of more advanced dialogues. These user utterances were recorded by at least two people.

Table 3.4: *Test and training set of small collection of "example" interactions.*

	Training	Test
prompt sets	5	4
recorded users	16	9
male/female	10 / 6	7 / 2
native/non-native	12 / 4	5 / 4
sentences	1500	700
Words	7000	4000

The data was divided into a test set and a training set (see table 3.4). It was taken care that the sets did not overlap. The training set was mostly used as held out data for interpolation, selection of the best model and other purposes. The test set was used for all the test runs done in the in-car tourist information domain. We believe that in the absence of recordings of users interacting with a real system this kind of data is the best approximation to it.

3.2.2 Simple generation Grammar

A simple grammar was written as a HTK grammar in Extended Backus-Nauer Form (EBNF). Such a grammar can be converted into a HTK word network. Then it can be directly used as a recognition

network in ATK. Also the HTK-random sentence generator (HSGen) [YEK+02] can take it as an argument for corpus generation. It is very easy to translate this kind of HTK-Grammar into an EBNF grammar that can be read by GF. The task grammar was structured in the following way:

1. Task specific semantic concepts (prices, names for hotels, bars, ...)
2. General concepts (local relations, numbers, dates, ...)
3. Query predicates (Want, Find, Exists, Select, ...)
4. Basic phrases (Yes, No, DontMind, Grumble, ...)
5. List of sub-grammars for user answers to all system prompts.
6. Main Grammar.

This structure makes it easy to debug the grammar and re-use rules. In future experiments it would be easy to create system state depended corpora for language model training. An example for a sub-grammar for a user reaction to a system prompt would appear as follows:

```
SYSTEM PROMPT:
   [ask happy] "Are you happy with those options?"

USER GRAMMAR:
   $REPLY_ASK_HAPPY = $YES_NO [ $THANK ] [ "GOODBYE" ] |
                      $Q_EXIST "SOMETHING MORE CENTRAL" |
                      "ARE THERE ANY OTHER OPTIONS" |
                      [ $YES_NO ] ( "THAT'S" ( "WHAT I WANTED" | "IT" ) |
                                    "I'M DONE" |
                                    "I GO FOR IT" ) [ $THANK ];
```

In this formalism all words in quotation marks are strings and will be printed as they are written. Words that start with a "$" are variables that may be replaced by rules defined elsewhere in the grammar. A perpendicular line "|" separates alternative items. Parentheses are used to group items and square brackets denote optional items. The main purpose of this Grammar is to generate a training corpus for trigram language models. Therefore dependencies between items that are more than tree words away from each other can be neglected.

3.2.3 Acoustic models for recognition experiments

All experiments were carried out on the test set as specified in table 3.4, using the Application Toolkit for HTK [You04] for speech recognition. For the acoustics the WSJCAM0 word internal triphone models distributed with ATK were used.

These models were adapted to an in-domain development set using Maximum A-Priori (MAP) adaptation and a HLDA transform (heteroscedastic linear discriminant analysis plus tertiary derivatives) was added to the system. The adaptation set included all user utterances of the SACTI data collection (see table 3.6) and the training set as specified in table 3.4.

Table 3.5: *WSJCAM0 Models were trained on text read by British English speakers.*

	WSJCAM0
users	92
sentences	7900
Words	130k

Table 3.6: *User turns of the SACTI Wizard Of Oz data collection. Used for adaptation of acoustic data.*

	SACTI WOZ
users	43
sentences	3000
Words	20k

The acoustic models were fixed throughout all following experiments in the tourist information domain; only the language models were changed. The acoustic models are identical with those used in Deliverable 4.2 [LGS05].

3.2.4 Grammar Networks vs. Statistical Language Models

We compared the performance of a grammar network with a statistical language model. In the first experiment a simple generation grammar as explained in section 3.2.2 was compiled into a recognition network and used as a language model in the speech recogniser.

In a second recognition experiment the HTK-random sentence generator (HSGen) [YEK+02] was used to generate a corpus. This tool walks through the network from left to right. On each branching point it makes a random decision on which path to follow. With this method it is possible to generate corpora of different size and see which is best suited for language model training. It is of course not guaranteed that every sentence of a corpus has been generated, but statistical n-gram language models are good in generalising over unseen events. The best results were obtained with a corpus of 30k randomly generated sentences.

Grammar generated corpora have slightly different properties as natural language data. This can cause problems for some smoothing techniques that use counts-of-counts statistics to estimate unseen events. Usually the number of events that occur 1 to 7 times are of practical importance. If a grammar does not produce n-gram events with low numbers of occurrence, Good-Turing, Katz discounting and Kneser-Ney smoothing are problematic.

In the literature modified Kneser-Ney smoothing is regarded as one of the best smoothing methods [CG99]. We therefore used it where robust counts-of-counts statistics were available. Otherwise Witten-Bell smoothing was used, which does not rely on count-of-counts statistics at all.

Results for different sizes of training data are displayed in Figure 3.1. For small grammar generated corpora the WER is relatively high. Adding more training data decreases the WER until it reaches the best value. From this point on more data does not help but rather deteriorates the model and the WER

Figure 3.1: *Recognition results for different numbers of grammar generated sentences of the simple tourist information grammar.*

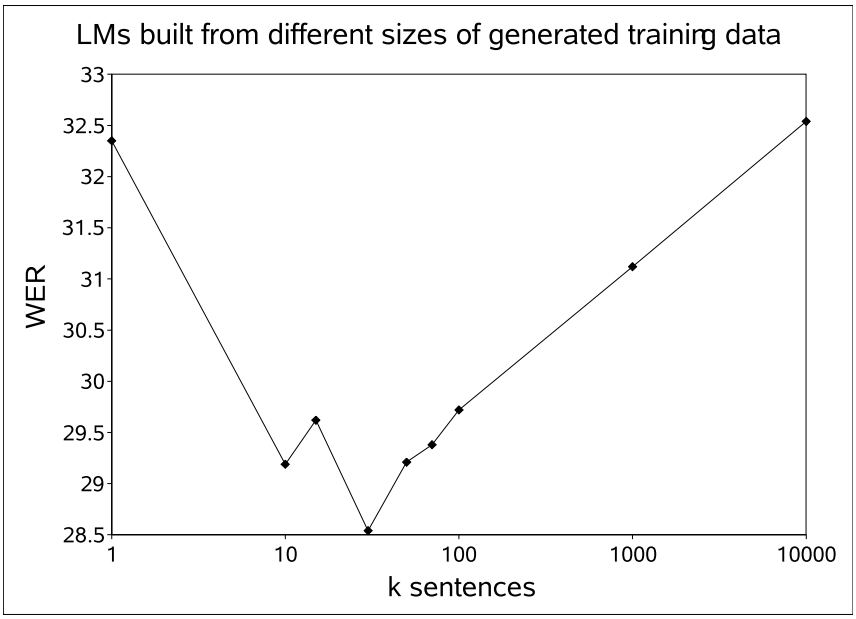


Table 3.7: *Word error rates (WER) of a grammar network and SLMs trained on grammar generated corpora and a Wizard Of Oz corpus. WERs of combined SLMS.*

Language Model	WER
Grammar network	40.4
SLM: 30k grammar corpus	28.5
SLM: SACTI WOZ corpus	28.7
SLM from pooled corpora: SACTI and 30k grammar sentences	22.2
interpolated SLMs: SACTI SLM and grammar SLM (10M sent.)	21.2
interpolated SLMs: FISHER SLM and grammar SLM (10M sent.)	22.7

increases again. The value given in table 3.7 is the result for the best language model which was trained on 30k grammar generated sentences. All results for statistical language models are considerably better than those for the grammar network.

3.2.5 In-Domain Language model

In the TALK project a Wizard Of Oz corpus in the Tourist information domain was collected (SACTI). It contains human-human dialogues. The users were given a map and asked to perform a number of tasks such as finding a hotel within a particular price range, or finding a restaurant of a particular type.

The major part of the dialogue was recorded in a ”simulated automated speech recognition (ASR) channel” [SWY04] [WY04] and a small portion was recorded as direct conversation. Half of the corpus consists

Table 3.8: *Contents of the SACTI-1 and SACTI-2 corpora used for language modelling.*

		Turns	words
SACTI-1	User, speech only, direct conversation	543	5054
	User, speech only, simulated ASR	2796	27197
	Wizard, speech only both	2737	53751
SACTI-2	User, speech and interactive map	2567	21255
	Wizard, speech and interactive map	2488	39440
$\sum$		11122	146697

of speech only dialogues in the other half an interactive map interface could be used along with speech. Table 3.8 summarises this briefly.

We expected this corpus to be quite representative for the task and built a language model from the transcriptions of the recordings. The class trigram model[2] was trained on both wizard and user turns. In table 3.7 we can see that the recognition result for the model built from this corpus is almost identical with that for the grammar generated model. It does not give the amount of improvement we expected. Although this corpus was recorded for the task, the language used in it is quite different from the language in the test set.

3.2.6 Combining Grammar and In-Domain Language models

The obvious next step is to combine the in-domain Wizard Of Oz corpus and grammar generated corpora and see if they complement each other. We investigated two ways of combining both corpora:

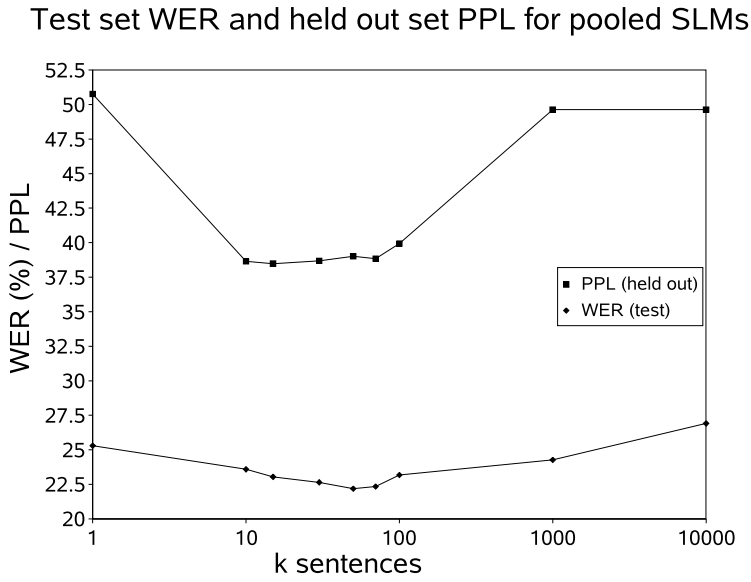
1. pool both corpora together and train one language model from the combined corpus.
2. build language models from each corpus and interpolate both language models.

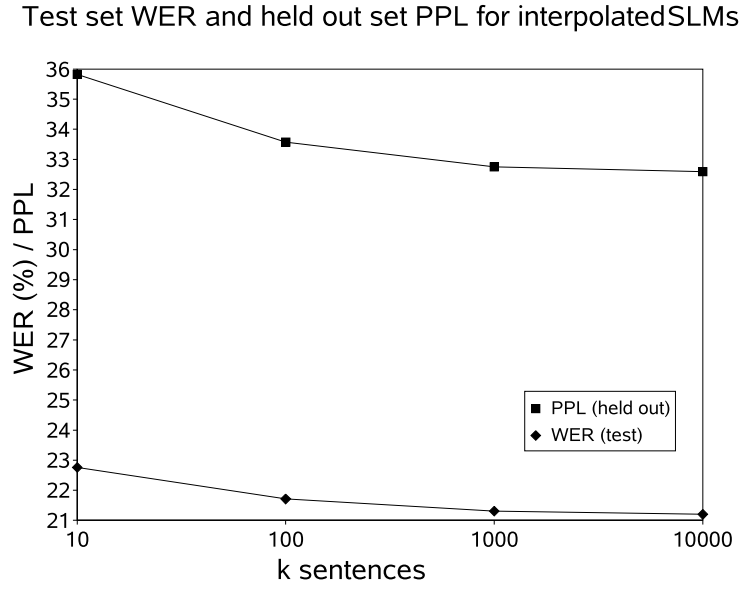
For both options a held out set is necessary, to estimate the weight that each corpus should contribute to the combined model. To evaluate the first technique it is only necessary to successively add larger numbers of grammar generated sentences to the WOZ corpus. In the top plot of figure 3.2 the perplexity calculated on a held out set and the WER of the test set are displayed for different numbers of generated sentences. Both lines first go down, reach a minimum and go up again. Although the shape of both curves is roughly the same the held out perplexity curve is not a tremendously good predictor for the minimum of the error rate on the test set. Instead of the actual minimum of 22.2 (for 50000 added grammar sentences) the WER corresponding to the perplexity minimum would have been 23.1 at 15000 sentences added from the grammar. In table 3.7 the actual minimum is reported.

The second technique involves building an SLM from the Wizard Of Oz data and then interpolating this SLM with SLMs trained on various grammar generated corpora. We used linear interpolation as supported in the SRI Language Modelling Toolkit [Sto02].

[2] A class model was used, because not all slot/value pairs that are possible in the dialogue system showed up in the corpus (E.g. no user asked for a single room.). A class model can smooth out effects like this.

Figure 3.2: *Recognition results (WER) on the test set and perplexity (PPL) on the held out set for pooled (top chart) and interpolated (bottom) SLMs, trained on a Wizard Of Oz corpus and on different numbers of grammar generated sentences of the simple tourist information grammar.*





$$P_{int} = (1-\lambda)P_{GrammarCorpus} + \lambda P_{WOZCorpus} \tag{3.3}$$

The interpolation weights were calculated using the CMU-Cambridge Statistical Language Modelling Toolkit [CR97]. For all data points the weight of the WOZ SLM was calculated to $\lambda = 0.49$. This means that both models contribute to the same degree to the interpolated model. The best result was obtained for an SLM built on 10M sentences (see table 3.7). As can be seen on the bottom chart of figure 3.2 the perplexity on the held out set is a good predictor for the test set WER. Both curves decrease as the grammar based SLM gets trained on a larger and larger corpus. We stopped at 10M sentences as generating a corpus of 100M sentences would have taken about a week. Our results show that linear interpolation of SLMs trained on different corpora leads to better results than merely adding both corpora together and training a SLM on the combined corpus. Linear interpolation is also more reliable when perplexity values calculated on a held out set are used to predict the best model combination for a recognition experiment on the test set.

3.2.7 Interpolating Language Models derived from a Grammar and a Standard Corpus

Given that it is rather expensive and tedious to collect a corpus of Wizard Of Oz recordings, we thought it might be an interesting strategy to start with language models that are built from a grammar generated corpus and interpolate them with models trained on a standard speech corpus such as the FISHER corpus [LDC05]. The FISHER corpus contains transcriptions of conversations about different topics. The idea behind this is, that the grammar generated data will contribute in-domain n-grams and the general corpus will add colloquial phrases.

For reasons of comparison the vocabulary used to build the Fisher SLM contained all words of the grammar and all words of the Wizard Of Oz data collection. This means that n-grams that contain other words were not included in the model. Then the Fisher SLM was interpolated with SLMs trained on grammar generated corpora of different size. For almost all interpolated models the optimal weight for the Fisher SLM was $\lambda = 0.33$. In figure 3.3 the perplexity calculated on the held out set and the WER of the test set are displayed for different interpolated models.

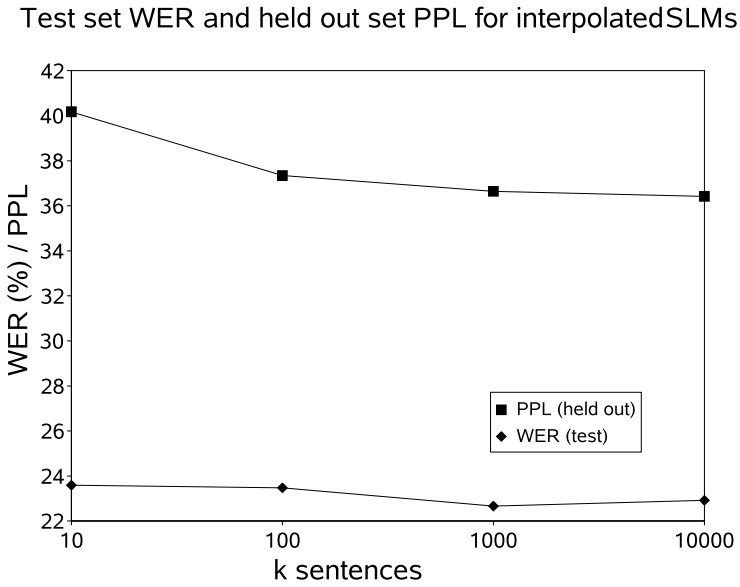
Both curves roughly have the same shape. The WER minimum is not at the same point as the perplexity minimum, but the difference between the absolute WER minimum of 22.7 and the value that would have been selected based on the perplexity minimum at the held out set is only 0.25.

3.2.8 Discussion of results

We compared speech recognition results using a recognition grammar with different kinds of statistical language models. All SLMs used in this section outperformed the recognition grammar network in terms of word error rates (WER) on the test set. We were very much surprised to what extent our statistical models outclassed the grammar network. Our best models nearly halved the WER.

A SLM trained on a corpus generated by a HTK grammar decreased WER by 29% relative compared to using the grammar directly. In a second series of experiments we interpolated a SLM trained on a grammar generated corpus with SLMs trained on two different corpora containing transcriptions of recorded speech. We used transcriptions of a Wizard Of Oz experiment, carried out in the domain of the dialogue system

Figure 3.3: *Recognition results (WER) on the test set and perplexity (PPL) on the held out set for interpolated SLMs, trained on the Fisher corpus and on different numbers of grammar generated sentences of the simple tourist information grammar.*



and the FISHER corpus consisting of a large collection of dialogues about different topics. In both cases we got substantial relative improvements of the WER: 26% for the WOZ corpus and 21% for the FISHER corpus.

It is quite surprising that combining a grammar trained SLM with an in domain Wizard Of Oz corpus was only a little better than combining it with a standard corpus. We think that the high costs involved in a Wizard Of Oz experiment are prohibitive given the relatively small gain in performance.

This supports the already earlier proposed strategy to start with language models that are built from a grammar generated corpus and interpolate them with SLMs trained on a standard speech data base. Once the dialogue system works one can do a data collection with the actual system and get the best data possible to retrain all components.

We were really astonished to what extend our SLMs outperformed the recognition grammar. Given these results we think it is highly promising for future work to put some effort in developing techniques that replace interpretation grammars or semantic parsers by components using the statistical approach.

Chapter 4

Selecting domain-relevant data

Usually the selection of domain relevant data for training a language model is carried out rather intuitively. If in-domain training data is not available, existing corpora are used which were collected in a similar domain or speaking style as the target domain. It is often the case that a big corpus exists that has relevant and irrelevant parts. These corpora are not always well documented, such that it is difficult, time consuming and sometimes impossible to select relevant parts from irrelevant parts.

A first selection of domain relevant parts of a corpus is simply done by defining the task vocabulary. All n-grams that contain words which are not in the vocabulary will not be included in the language model anyway. Further refinement would then involve selecting from all remaining "in vocabulary" n-grams those which are typical for the target domain.

In recent years the world wide web was identified as a possible source for training data for language models. It is practically impossible to use the whole of the Internet as a training corpus and therefore crucial to find ways of retrieving only relevant websites from the Internet [BM98, NOH+05, VAR99, SGN05, ZR01]. A number of different filtering techniques has been investigated to use large collections of written language or transcribed speech more effectively in language models [Wee04, ZSH00, CSW05, CWS+04].

4.1 Sentence selection based on in-domain vocabulary

Rosenfeld [Ros00b] argues that a little more domain corpus is always better than a lot more training data outside the domain. This has lead to the idea of extracting domain relevant data from bigger corpora. This section describes a very simple way of selecting domain relevant data from the Swedish news paper corpus described in section 3.1.3.

4.1.1 Extracting domain relevant data

To create a domain relevant corpus for the Swedish MP3 application sentences with domain related words were extracted from the Swedish newspaper corpus. We started by creating a domain relevant vocabulary taking the existing MP3 application vocabulary and adding missing domain related words (e.g. music, mp3-player, songs etc.). The resulting vocabulary was a vocabulary without highly frequent words such as functional words and pronouns (we also excluded ambiguous words e.g. 'låt" (Eng. 'song' or 'let')) i.e.

it only consisted of typical domain words. We used this domain vocabulary to extract all sentences where these domain words occurred from the Swedish newspaper corpus. The corpus we obtained consisted of about 15 million words i.e. 4% of the larger news corpus. We used this domain relevant part of the news corpus to create a new language model, DomNewsLM, thought to be more relevant to the domain than the NewsLM model.

By interpolating the MP3GFLM model described earlier with this new model we get a mixed model to test for domain relevance. The vocabulary used for this mixed model is the one described in 3.1 where the most frequent words in the GSLC corpus, the most frequent ones in the newspaper corpus and the vocabulary used for extracting domain data was added to the original MP3 grammar vocabulary. We are thereby using the same vocabulary as for the MixNewsMP3GF.

$$MixDomNewsMP3GF = \lambda * MP3GFLM + (1-\lambda) * DomNewsLM \tag{4.1}$$

4.1.2 Recognition results for domain adapted models

Testing domain relevance in this case means testing if perplexity decreases and recognition performance improves both in comparison with the NewsLM and the MP3GFLM. Table 4.1 shows the perplexity for the different SLMs.

Table 4.1: *Perplexity for the different LMs.*

LM	Perplexity
MP3GFLM	587
NewsLM	386
DomNewsLM	321
MixNewsMP3GF	78
MixDomNewsMP3GF	75

We can see a slight decrease in perplexity for our new domain adapted model in comparison with the NewsLM model and an important difference in perplexity for the mixed models in comparison with the others.

In 4.2 we can find the recognition word error rates for the models we are comparing.

Table 4.2: *Word error rates(WER) for the recording test set*

LM	WER(NBest)
MP3GFLM	37.11 (29.48)
NewsLM	61.62 (49.53)
DomNewsLM	45.03 (31.58)
MixNewsMP3GF	38.00 (27.37)
MixDomNewsMP3GF	34.07 (22.07)

Despite just a slight decrease in perplexity we get an important decrease in WER (27 % relative improvement) for our domain adapted model, DomNewsLM, in comparison with the NewsLM model. However, the

domain adapted model does not outclass the handcrafted MP3GFLM model which seems better suited to the domain. It is first by using the mixed model MixDomNewsMP3GF that we find a model better suited for the domain. This shows that our simple approach of extracting domain relevant data from a bigger corpus and integrate this data into our model can improve our original model. However, just adding any corpora and integrating this with our model does not necessarily mean an improvement, e.g. the MixNewsMP3GF model performs worse than the simple MP3GFLM model.

The GSLC corpus although of very small size but consisting of transcribed spoken language seemed more suitable for the domain giving a small improvement when integrated with our MP3GFLM. This leads us to think that if the GSLC corpus would have been larger our simple way of extracting domain relevant data would perhaps have given us an even better model. The GSLC corpus is as said based on activity types (see [All99]) and consists of several different texts collected from different activities such as church sermons, auctions, task-oriented dialogues or meetings. We have carried out some similar experiments with the GSLC corpus by choosing the most appropriate activities and creating a language model from these. In this case 'appropriate' means activities giving low perplexities when we tested our baseline model MP3GFLM on each activity text. Unfortunately, the GSLC corpus is very small and the domain related corpus from GSLC get too small to get any reliable language model. However, although the recognition performance did not improve significantly by using just the most essential part of the GSLC corpus nor did it degrade the performance. This may indicate that we at least managed to eliminate some superfluous information.

4.2 Sentence selection using perplexity filtering

In this case study we want to investigate the promises and limitations of perplexity filtering for improving language models for dialogue systems. We will use the simple in-car grammar to generate a seed corpus. Since we have the FISHER corpus and the Wizard Of Oz corpus available we will combine them and use the combined corpus as a large corpus from which we will select sentences. In combining these two corpora, we make sure that we have relevant and irrelevant sentences in the corpus to chose from. For reasons of comparability we use the same vocabulary, the same held out set and the same test set as in our previous experiments in section 3.2. The following procedure was applied:

- build a language model from a seed corpus LM_{Seed}
- build a language model from the large corpus LM_{Large}
- calculate relative perplexity $PP_{Rel} = PP_{LM_{Seed}}/PP_{LM_{Large}}$ for each sentence and sort corpus according to PP_{Rel}.
- select n lines with lowest PP_{Rel} and build a language model $LM_{Selected}$ from them. Do the same with the remaining lines.
- interpolate LM_{Seed}, $LM_{Selected}$ and $LM_{NotSelected}$ using the training part of the "invented dialogues" as detailed in table 3.4 to derive λs.
  $LM_{Filt} = (1-\lambda_{Sel}-\lambda_{NotSel})LM_{Seed} + \lambda_{Sel}LM_{Selected} + \lambda_{NotSel}LM_{NotSelected}$

We used 10M sentences generated by the simple in-car tourist information grammar as a seed corpus and increased the number of selected sentences n successively to obtain the perplexity graph shown in figure 4.1.

Table 4.3: *Speech recognition results on the test set.*

Language model	WER
ppl-filtering: (420000 sent.)	21.1
Large corpus LM interpolated with Seed LM	22.4
SACTI LM and Fisher LM interpolated with Seed LM	19.9

After a noisy initial segment[1] the perplexity stabilises at a certain level, goes down to a minimum and slowly increases again, as a growing number of sentences are selected. The first row of table 4.3 shows the test set WER obtained for the model that produced the best perplexity score on the held out set. This model is about 1% absolute better than interpolating the seed corpus directly with the large corpus (Fisher and SACTI pooled). Since we composed the large corpus from two smaller corpora, we can build $LM_{Selected}$ from the Wizard Of Oz corpus and $LM_{NotSelected}$ from the Fisher corpus. This model is about 1% better than the one based on perplexity filtering.

4.3 Discussion of results

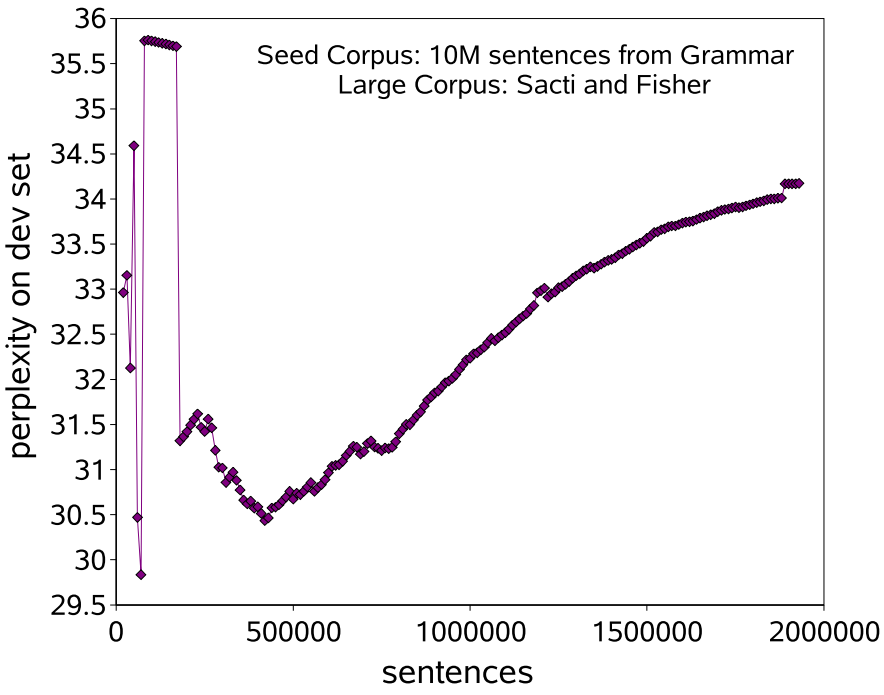
Both investigations using different techniques show that selecting domain relevant data is useful when applied to large, diverse corpora. In both experiments interpolation of a grammar generated baseline SLM with a SLM built on selected sentences results in a good WER improvement.

It comes as no surprise that probably the most important factor influencing the degree of improvement to be gained by data selection from large corpora is the nature of the corpus from which these sentences are selected. If all parts of the corpus are of a similar domain relevance, then the gain will be small. Only if there are parts of different domain relevance, improvement is possible.

The results in this section are encouraging to continue work on selecting domain relevant data. The experiments on perplexity filtering show that, there is room for improvement of current techniques. Future work could be directed towards improving techniques presented in this paper. Another issue could be to investigate the influence of corpus properties on the success of such techniques.

[1]The first part of the graph is very noisy, because basically all the one-word sentences like “yes”, “no”, “ok”, ... get very high scores. This leads to quite unnatural count-of-count statistics such that the selected LMs are effectively broken in this area.

Figure 4.1: *Perplexity of interpolated SLM calculated on the held out set (dev set).*



Chapter 5

Generating dialogue move specific SLMs

We have also experimented with dialogue move specific language models by using GF to generate all utterances that are specific to certain dialogue moves from our interpretation grammar. In this way we can produce models that are sensitive to the context but also, by interpolating these more restricted models with the general GF language model, not restrict what the users can say but take into account that certain utterances should be more probable in a specific dialogue context. Context-sensitive models and specifically grammars for different contexts have been explored earlier [BDG+97, WPI99, LG04] as well as in this project (see TALK deliverable D4.1 and [Lem04]), but generating a corpus for such language models artificially from an interpretation grammar by choosing which moves to combine seems to be a new direction. Dialogue move specific language models are models where utterances corresponding to certain moves are more salient (e.g. a model where all ways of answering yes and no are more plausible). Our first experiments seem promising but the dialogue move specific test sets are too small to draw any conclusions. We created three different models that we tested on parts of the test set described in section 3.1.

5.1 Moves in the MP3 Domain

Although the number of dialogue moves in our system is quite small the possibility of combining these in the same turn makes possible classes of move sets per turn, encountered in dialogue logs, reasonably large.

We looked at some of the automatically generated logs from interactions with the MP3 player application (DJGoDiS). There were 40 different move combinations associated with turns. We distinguish the following user moves in GoDiS: `greetings`, `requests`, `answers`, `ask moves`, `quit move`, `help move`, `yes and no answers` and `ICMs` (see [Lar02]). Examples of user moves are shown in table 5.1.

5.2 Dialogue move specific language models

To decide which models to create we took a look at common combinations of moves from automatically generated logs from interactions with the MP3 player. It was very common to give either the name of a song or a group or a combination of these in the same dialogue situations. We therefore created a

Table 5.1: *Dialogue moves used in the domain*

Dialogue move	Utterance example
greet	Hi!
quit	Bye!
help	Help.
answer(song(dancing queen))	Dancing Queen
answer(station(rant radio))	Rant Radio
answer(index(3))	number three
answer(group(Abba))	Abba
ask(X^current_song(X)))	what song is this
request(pause)	pause the music
request(clear)	clear the playlist
request(next_song)	i want to listen to the next song
answer(yes)	yeah
icm:acc*pos	OK

language model (`AnsGroupSongLM`) based on the `answer` moves for song and group and combinations of these by extracting all utterances corresponding to these moves from our GF grammar. This corresponds to utterances such as:

```
Dancing Queen
Abba
Dancing Queen by Abba
Abba with Dancing Queen
```

Another language model we created for our tests was a model including all requests the user does to control the mp3 player such as lowering the volume, pausing the music, skipping to next song etc. The utterances making up these requests were generated from the GF grammar. It should be noted that there exist a lot of other requests in the domain that are not included in this model as they correspond to other dialogue situations such as altering the playlist. We called this model the `RequestLM`.

A final model was a model with all yes and no answers which has been a common context-specific model to test (`MixYNLM`).

The MP3 player is a difficult test scenario as the dialogues are so direct and shallow, however these three models seem to be useful in the current setting as they are quite predictable. More models could of course be considered based on other moves and move combinations but to be able to show that dialogue move specific models can be useful and improve recognition performance we think it will be sufficient with these three models.

We interpolated each of these specific models with the general MP3 model (`MP3GFLM`) to get the mixed and less restrictive models we are looking for. We tested several weights and found an optimal weight of 85 for the mixed models. The tests presented in the following section compare the specific models, with the general one and the mixed ones which are the dialogue moves specific models we are considering.

5.3 Results

We tested all three models on test sets suited for each model that included utterances that corresponded to the moves in the model. It should be mentioned that this implies that the test sets may include utterances not covered by the GF grammar although considered belonging to some of the moves (e.g. a different wording for the same move). We tested the general model (the MP3GFLM from 3.1), the highly specific models and the mixed models on the same test set and compared the figures. We also selected a test set that only included in-coverage utterances.

Table 5.2: *Word error rates(WER) for the request model on move test set*

LM	WER(NBest)
RequestLM	43.35 (38.56)
GeneralLM	37.50 (30.05)
MixReqConLM	33.78 (24.40)

Table 5.3: *Word error rates(WER) for the request model on in-coverage test set*

LM	WER(NBest)
RequestLM	8.11 (0.00)
GeneralLM	13.51 (8.11)
MixReqConLM	9.91 (2.70)

Table 5.4: *Word error rates(WER) for the group and song model on move test set*

LM	WER(NBest)
AnsGroupSongLM	1.92 (1.10)
GeneralLM	2.47 (1.92)
MixGroupSongLM	2.19 (1.10)

5.4 Discussion of results

Context-specific language models or grammars have shown important recognition performance gain in earlier work([Lem04, BDG+97, KW01, GWS05]) and our first results show that our dialogue move specific models would also give an important gain in recognition performance. The improvement varies dependent on the dialogue moves we are modelling giving us about 10% relative improvement for the request and answer song and group models while as much as 27% for the yes and no model. The generation method we use makes it easier to obtain context-specific LMs and assures that we have at least the same coverage as the original interpretation grammar. Collection of corpora for context-specific language models has the drawback that sparse data turns into an even bigger problem and writing grammars for every state or move is tedious work.

Table 5.5: *Word error rates(WER) for the yn model on move test set*

LM	WER(NBest)
GeneralLM	66.18 (51.47)
MixYNLM	48.35 (32.35)

Table 5.6: *Word error rates(WER) for the yn model on in-coverage test set*

LM	WER(NBest)
GeneralLM	59.52 (38.10)
MixYNLM	30.95 (9.52)

Ideally we would like to have our dialogue move specific models generated and interpolated on the fly. However, another solution would be to have certain models chosen from the beginning and change between these during dialogue interaction. In either case we need a way to be able to choose between them i.e. choose which model suits best the current information state. We have started tests with machine learning on dialogue move prediction based on the current information state which seems promising (inspired by the achievement of [GL04] when using information state features with machine learning for the task of reranking N-Best lists). However those experiments are out of the scope of this deliverable.

Chapter 6

Conclusion and future work

A first observation is that statistical language models give us much more robust recognition, as expected. Our best language models for the Swedish MP3 as well as for the English tourist information domain roughly halve the word error rate compared to the baseline i.e. the grammar network. However, this also implies a falling off in in-grammar performance. It is interesting that language models that are trained only on a corpus generated by the grammar although being more robust and giving a significant reduction in WER rate, have only a marginally degraded performance on in-grammar sentences compared to using the grammar network directly. These simple models seem promising to use in a first version of the system with the possibility of improving it when logs from system interactions have been collected. In addition, the vocabulary of these models is in sync with our GF interpretation grammars. The results are comparable with those obtained by [BJ03] using random generation to produce a language model from an interpretation grammar.

Half of the improvement through using language models comes from models trained on grammar generated corpora and the other half comes from interpolating with models trained on corpora of conversational speech. Although when dealing with grammar generated corpora, perplexity measures do not reliably translate into error rates

Interpolating grammar generated SLMs with speech corpora consistently improves the word error rate. The closer to the domain of the speech corpus the better the WER improvement. This explains why general newspaper corpora are not so well suited. It seems apparent from the tests that the quality of the data is more important than the quantity. This makes extraction of domain data from larger corpora an important issue and increases the interest in generating artificial corpora. Both experiments on domain data selection demonstrate that sentence selection with a relative perplexity criterion or based on in domain vocabulary can improve recognition results. But our experiments for the English in-car domain also clearly showed that there is still considerable room for improvement and future research.

To summarise, we seem to have found a good way of compromising between the ease of grammar writing and the robustness of statistical language models in the first stage of dialogue system development. In this way we can use the knowledge and intuition we have about the domain, include it in our first language model and get a more robust behaviour than with a grammar. From this starting point we can then collect more data with our first prototype of the system to improve our language model further.

Inspired by the success of statistical language models trained on a combination of grammar generated corpora and standard speech corpora UCAM plans to take up some of the methods described in this deliverable and adapt them to the training of statistical semantic parsers. Semantic decoding is the missing

link between the speech recognition component and the dialogue manager, providing the latter with the semantic concepts of a user's utterance (see also status report T1.4s2).

UGOT are planning future experiments with GF-generated SLMs for the other applications in the UGOT home scenario. In this way, we will have SLMs working also for these domains instead of speech recognition grammars. We will also do further work on dialogue move specific models and we are considering implementing this approach in the GoDiS system enabling GoDiS with a language model switching approach. We hope to report on this in the near future.

Bibliography

[All99] J. Allwood. The Swedish Spoken Language Corpus at Göteborg University. In *in Fonetik 99, Gothenburg Papers in Theoretical Linguistics 81, Dept. of Linguistics*, Dept. of Linguistics, University of Göteborg, 1999.

[BDG+97] P. Baggia, M. Danieli, E. Gerbino, L. M. Moisa, and C. Popovici. Contextual information and specific language models for spoken language understanding. In *Proceedings of SPECOM'97*, Cluj-Napoca, Romania, 1997.

[BJ03] S. Bangalore and M. Johnston. Balancing data-driven and rule-based approaches in the context of a multimodal conversational system. In *ASRU: Automatic Speech Recognition and Understanding*, 2003.

[BM98] Adam Berger and Robert Miller. Just-in-time language modelling. In *Proceedings of the ICASSP*, 1998.

[CG99] Stanly F. Chen and Joshua T. Goodman. An empirical study of smoothing techniques for language modeling. *computer Speech and Language*, 13:359–397, 1999.

[CR97] P.R. Clarkson and R. Rosenfeld. Statistical language modeling using the CMU-Cambridge toolkit. In *Proceedings ESCA Eurospeech*, 1997. http://mi.eng.cam.ac.uk/ prc14/toolkit.html.

[CR01] A. Chotimongkol and A. I. Rudnicky. N-best speech hypotheses reordering using linear regression. In *Proceedings of Eurospeech 2001*, Aalborg, Denmark, 2001.

[CSW05] G. Chung, S. Seneff, and C. Wang. Automatic induction of language model data for a spoken dialogue system. In *Proceedings of SIGDIAL*, Lisbon, Portugal, 2005.

[CWS+04] G. Chung, C. Wang, S. Seneff, M. Tang, and E. Filisko. Combining linguistic knowledge and acoustic information in automatic pronunciation lexicon generation. In *Proceedings of ICSLP*, Jeju Island, Korea, 2004.

[FLK01] E. Fosler-Lussier and H.-K. J. Kuo. Using semantic class information for rapid development of language models within ASR dialogue systems. In *Proceedings of the International Conference on Acoustic, Speech, and Signal Processing (ICASSP)*, Salt Lake City, Utah, 2001.

[GL04] M. Gabsdil and O. Lemon. Combining acoustic and pragmatic features to predict recognition performance in spoken dialogue systems. In *Proceedings of ACL*, Barcelona, 2004.

[GLR02] G. Gorrell, I. Lewin, and M. Rayner. Adding intelligent help to mixed initiative spoken dialogue systems. In *Proceedings of the ICSLP*, 2002.

[GWS05] Alexander Gruenstein, Chao Wang, and Stephanie Seneff. Context-sensitive statistical language modeling. In *Proceedings of Interspeech*, 2005.

[HAH01] X. Huang, A. Acero, and H-W Hon. *Spoken Language Processing: A guide to theory, algorithm and system development*. Prentice Hall, 2001.

[JDB98] D. Janiszek, R. De Mori, and F. Bechet. Data augmentation and language model adaptation. Technical report, University of Avignon, 84911 Avignon Cedex 9 - France, 1998.

[JM80] F. Jelinek and R. Mercer. Interpolated estimation of markov source parameters from sparse data. In *In Pattern Recognition in Practice, E. S. Gelsema and L. N. Kanal*, North Holland, Amsterdam, 1980.

[Jon06] Rebecca Jonson. To appear: Generating statistical language models from interpretation grammars in dialogue systems. In *"Proceedings of 11th Conference of the European Association of Computational Linguistics*, Trento, Italy, 2006.

[KGR+01] S. Knight, G. Gorrell, M. Rayner, D. Milward, R. Koeling, and I. Lewin. Comparing grammar-based and robust approaches to speech understanding: A case study. In *Proceedings of Eurospeech*, 2001.

[KW01] Hacioglu K. and Ward W. Dialog-context dependent language modeling combining n-grams and stochastic context-free grammars. In *Proceedings of ICASSP*, Salt Lake City, Utah, 2001.

[Lar02] S. Larsson. Issue-based dialogue management. phd thesis, 2002.

[LDC05] Fisher corpus. LDC Catalog, 2005. http://www.ldc.upenn.edu/Catalog.

[Lem04] O. Lemon. Context-sensitive speech recognition in ISU dialogue systems: results for the grammar switching approach. In *Proceedings of CATALOG , 8th Workshop on the Semantics and Pragmatics of Dialogue*, Barcelona, 2004.

[LG04] Oliver Lemon and Alexander Gruenstein. Multithreaded context for robust conversational interfaces: context-sensitive speech recognition and interpretation of corrective fragments. *ACM Transactions on Computer-Human Interaction (ACM TOCHI)*, 11(3):241– 267, 2004.

[LGHS06] Oliver Lemon, Kallirroi Georgila, James Henderson, and Matthew Stuttle. An ISU dialogue system exhibiting reinforcement learning of dialogue policies: generic slot-filling in the TALK in-car system. In *Proceedings of EACL*, page to appear, 2006.

[LGS05] Oliver Lemon, Kallirroi Georgila, and Matthew Stuttle. D4.2: Showcase exhibiting Reinforcement Learning for dialogue strategies in the in-car domain. Technical report, TALK Project, 2005.

[NOH+05] Tim Ng, Mari Ostendorf, Mei-Yuh Hwang, Ivan Bulyko, Manhung Siu, and Xin Lei. Web-data augmented language model for mandarin speech recognition. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Philadelphia, US, 2005. http://ssli.ee.washington.edu/projects/ears/WebData/web_data_collection.html.

[Nua05] Nuance Communications, as of May 2005. http://www.nuance.com.

[PSB01] SV. Pakhomov, M. Schonwetter, and J. Bachenko. Generating training data for medical dictations. In *Proceedings of the NAACL*, 2001.

[QAM+02] J. F. Quesada, J. G. Amores, P. Manchón, G. Peréz, S. Knight, D. Milward, and J. Thomas. Possibilities for enhancing speech recognition by consulting information states. In *Deliverable D2.3*, 2002.

[Ran05] A. Ranta. Grammatical framework homepage, 2005. http://www.cs.chalmers.se/∼aarne/GF.

[RHJ+00] M. Rayner, B. A. Hockey, F. James, E. Owen Bratt, S. Goldwater, and J.M. Gawron. Compiling language models from a linguistically motivated unification grammar. In *Proceedings of the COLING*, 2000.

[RLBE00] A. Raux, B. Langner, A. Black, and M. Eskenazi. Let's go: Improving spoken dialog systems for the elderly and non-natives. In *Proceedings of the Eurospeech*, Geneva, Switzerland, 2000.

[Ros00a] Ronald Rosenfeld. Incorporating linguistic structure into statistical language models. In *In Philosophical Transactions of the Royal Society of London A*, 2000.

[Ros00b] Ronald Rosenfeld. Two decades of statistical language modeling: Where do we go from here? In *Proceedings of the IEEE*, 2000.

[SFLK+02] R. Solsona, E. Fosler-Lussier, H. J. Kuo, A. Potamianos, and I. Zitouni. Adaptive language models for spoken dialogue systems. In *Proceedings of the 2002 International Conference on Acoustic Speech and Signal Processing (ICASSP-2002)*, Orlando, Florida, USA, 2002.

[SGN05] A. Sethy, P. G. Georgiou, and S. Narayanan. Building topic specific language models from webdata using competetive models. In *In Proceedings of the Eurospeech*, 2005.

[Sto02] Andreas Stolcke. SRILM - An extensible language modeling toolkit. In *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado, September 2002. http://www.speech.sri.com/projects/srilm/.

[SWY04] Matthew Stuttle, Jason D. Williams, and Steve Young. Framework for dialogue data collection with a simulated ASR channel. In *Proceedings of the ICSLP*, Jeju, South Korea, 2004.

[VAR99] D. Vaufreydaz, M. Akbar, and J. Rouillard. Internet documents: A rich source for spoken language modeling. In *Proceedings of the ASRU Conference*, pages 277–280, Keystone, USA, 1999.

[vNBKN99] G. van Noord, G. Bouma, R. Koeling, and M. Nederhof. Robust grammatical analysis for spoken dialogue systems. In *Journal of Natural Language Engineering*, 1999.

[Wee04] Chze Ling Wee. Web data for language modelling of conversational telephone speech. Master's thesis, Cambridge University Engineering Dept, Machine Intelligence Laboratory, Trumpington Street, Cambridge, CB2 1PZ, United Kingdom, 2004.

[WPI99] H. Wright, M. Poesio, and S. Isard. Using high level dialogue information for dialogue act recognition using prosodic features. In *In DIAPRO-1999*, 1999.

[WY04] Jason D. Williams and Steve Young. Characterizing task-oriented dialog using a simulated ASR channel. In *Proceedings of the ICSLP*, Jeju, South Korea, 2004.

[XR00] W. Xu and A. Rudnicky. Language modeling for dialog system. In *Proceedings of ICSLP 2000*, Beijing, China, 2000.

[YEK+02] Steve Young, Gunnar Evermann, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book, Version 3.2*. Machine Intelligence Laboratory, Cambridge University Engineering Dept, Trumpington Street, Cambridge, CB2 1PZ, December 2002. http://www.htk.eng.cam.ac.uk.

[You04] Steve Young. *ATK: An Application Toolkit for HTK, Version 1.4.1*. Machine Intelligence Laboratory, Cambridge University Engineering Dept, Trumpington Street, Cambridge, CB2 1PZ, July 2004. http://htk.eng.cam.ac.uk/develop/atk.shtml.

[ZR01] Xiaojin Zhu and Ronald Rosenfeld. Improving trigram language modeling with the world wide web. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2001.

[ZSH00] Imed Zitouni, Kamel Smaïli, and Jean-Paul Haton. Beyond the conventional statistical language models: the variable-length sequences approach. In *Proceedings of the Sixth International Conference on Spoken Language Processing*, Beijing, China, 2000.