Bootstrapping Language Models for Dialogue Systems

Karl Weilhammer, Matthew N. Stuttle and Steve Young

Cambridge University Engineering Department,
Trumpington Street, CB2 1PZ, Cambridge, UK
kw278@eng.cam.ac.uk, mns25@eng.cam.ac.uk, sjy@eng.cam.ac.uk

Abstract

We report results on rapidly building language models for dialogue systems. Our base line is a recogniser using a grammar network. We show that we can almost halve the word error rate (WER) by combining language models generated from a simple task grammar with a standard speech corpus and data collected from the web using a sentence selection algorithm based on relative perplexity. This model compares very well to a language model using "in-domain" data from a Wizard Of Oz (WOZ) collection. We strongly advocate the use of statistical language models (SLMs) in speech recognisers for dialogue systems and show that costly WOZ data collections are not necessary to build SLMs.
Index Terms: dialogue systems, speech recognition, language models, grammar.

1. Introduction

Poor speech recognition performance is often described as the single most important factor prohibiting the wider use of spoken dialogue systems. In this investigation we will show how error rates can be reduced with minimal effort using very simple techniques. The domain of our dialogue system is a tourist information task, in which the user can ask for information about hotels, bars and restaurants in an invented town. The task language is English.

Many dialogue systems use recognition grammars instead of statistical language models (SLMs) [1]. We will compare this standard approach with using SLMs that are trained from an artificial corpus which was generated by a recognition grammar [2, 3, 4, 5, 6]. Collecting "in-domain" data under a Wizard Of Oz (WOZ) paradigm or a similar collection framework is regarded as the best way to get good training material for language models. We compare our grammar generated SLMs with SLMs trained on such an "in-domain" corpus. In a third series of experiments we interpolate our grammar generated SLMs with "in-domain" data and a standard speech corpus. Finally we describe a sentence selection algorithm and apply it to a standard speech corpus and a corpus that was collected from the Internet.

2. Experimental setup

2.1. 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. The data was divided into a test set and a training set (see table 1). 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 tourist information domain.

Table 1: *"Example" interactions of training and test set.*

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

2.2. Generation and recognition grammar

A simple HTK grammar was written consisting of around 80 rules in extended Backus-Naur Form (EBNF). The grammar was used in two ways: Firstly it was converted into a word network (16996 nodes and 40776 transitions) to be used in the recogniser. Secondly a corpus of random sentences was generated from the grammar. The task grammar was structured in the following way:

1. Task specific semantic concepts (prices, hotel names ...)
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 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. The grammar is overgenerating, allowing some utterances which are not proper English sentences, or which do not make sense semantically.

2.3. Acoustic models

All experiments were carried out on the test set as specified in table 1, using the trigram decoder in the Application Toolkit for HTK (ATK) [7] for speech recognition. For the acoustic models the WSJCAM0 word internal triphone models[1] distributed with ATK were used. These models were adapted to a 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[2] (see sect. 3.2) and the training set as specified in table 1. The acoustic models used

[1] 92 speakers of British English, 7900 read sentences, 130k words
[2] 43 users, 3000 sentences, 20k words

throughout all following experiments in the tourist information domain were fixed; only the language models were changed.

3. Experiments

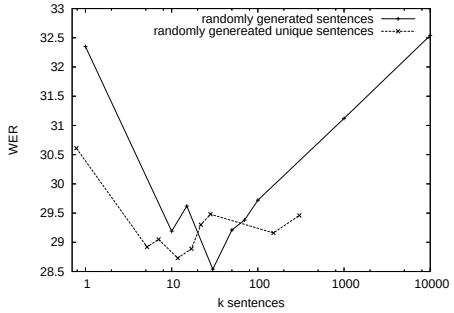
In this section we train language models on corpora that were generated by grammars. These artificial corpora have slightly different properties compared to natural language data. This can cause problems for some smoothing techniques that use counts-of-counts statistics to estimate unseen events. Usually 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 [8]. 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.

3.1. Grammar networks vs. statistical language models

In the first experiment we compared the performance of a grammar network with a SLM. A simple generation grammar (as explained in section 2.2) was compiled into a recognition network and used as a language model in the speech recogniser. This system yielded an error rate of 40.4%.

In a second recognition experiment the HTK-random sentence generator [9] 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 sizes and see which is best suited for language model training. Not every sentence of a corpus will be generated, but statistical n-gram language models are good in generalising over unseen events.

Figure 1: *Recognition results for different numbers of grammar generated sentences of the simple tourist information grammar.*



Word error rate (WER) results for different sizes of training data are displayed in Figure 1. The best result of 28.5 was obtained with a corpus of 30k randomly generated sentences. For small grammar generated corpora the WER is relatively high. Adding more training data decreases the WER until it reaches a minimum. From this point on more data does not help but rather deteriorates the model and the WER increases again. The reason for this could be that the random sentence generator overweights common n-grams by repeating short sentences more often than long sentences. To fix this problem we removed all duplicate sentences from the corpora, such that they contained only unique sentences. Figure 1 shows that for language models generated from these corpora the WERs are less sensitive to the size of the corpus. With a WER of 28.7 the best value in this experiment is close to that of the previous experiment.

Table 2: *Word error rates (WER) of a grammar network and SLMs trained on grammar generated corpora and a WOZ corpus. WERs of combined SLMs.*

Language model	WER
Grammar network	40.4
SLM: 30k grammar corpus	28.5
SLM: SACTI WOZ corpus	28.7
SACTI SLM and grammar SLM (10M sent.)	21.2
Fisher SLM and grammar SLM (10M sent.)	22.7

All results for SLMs are considerably better than those for the grammar network. The best model gives a relative improvement of 29% over using the grammar.

Reading the graphs in figure 1 from the practical view point suggests that a grammar written for the purpose of corpus generation, can be much less precise than a grammar intended to be used as a recognition network. The grammar developer can save a lot of time and effort. Trigram SLMs only capture dependencies that stretch over three words and smooth out many of the unseen word combinations, whereas a recognition grammar must cover all possible sentences explicitly.

3.2. "In-domain" language model

In the TALK project a WOZ corpus in the Tourist information domain (SACTI) was collected. It contains human-human dialogues (11122 turns, 147k words) in which the users were given a map and asked to perform a number of tasks. The major part of the dialogues was recorded in a "simulated automated speech recognition (ASR) channel" [10] [11] and a small portion was recorded as direct conversation. Half of the corpus consists of speech only dialogues. In the other half an interactive map interface could also be used along with speech.

We expected this corpus to be quite representative for our task and built a language model from the transcriptions of the recordings. The class trigram model[3] was trained on both wizard and user turns. In table 2 we can see that the WER of 28.7% for this model is almost identical with that for the grammar generated model.

The problem with this corpus is that it was recorded well before the dialogue system was defined that was assumed for the collection of the test set. We believe, this is a realistic situation, since it is very likely that during the development the prompt set or the strategy used by the dialogue system may change several times.

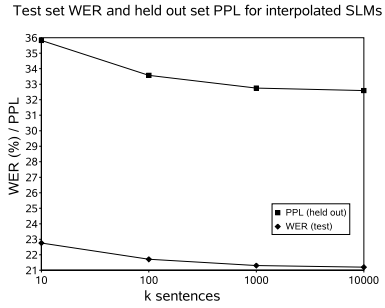
Given the near-identical WER, it is much faster and cheaper to adapt a simple grammar than to record and transcribe in-domain data.

3.3. Combining grammar and "in-domain" language models

The obvious next step is to combine the "in-domain" WOZ corpus and our grammar generated corpora to see if they complement each other. We used linear interpolation as supported in the SRI

[3] A class model was used, because not all slot/value pairs that are possible in the dialogue system appeared in the corpus (E.g. no user asked for

Figure 2: *Recognition results (WER) on the test set and perplexity (PPL) on the held out set for interpolated SLMs, trained on a WOZ corpus and on different numbers of grammar generated sentences of the simple tourist information grammar.*



Language Modelling Toolkit [12]. The interpolation weights were calculated using the CMU-Cambridge Statistical Language Modelling Toolkit [13]. The outcome of this calculation was that for all data points the optimal weights of the WOZ SLM was roughly $\lambda = 0.5$. This means that both models contribute to the same degree to the interpolated model. The best result was obtained for an SLM built from the largest corpus containing 10M sentences (see table 2). As can be seen on the chart of figure 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. It seems that interpolation of the grammar corpus with a corpus of natural speech straightens out most of the artifacts caused by the random generator.

3.4. Interpolating language models derived from a grammar and a standard corpus

Given that it is rather expensive and tedious to collect a corpus of WOZ recordings, a lower cost strategy would be 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 [14]. 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 from the WOZ data collection. This means that n-grams that contain other words were not included in the model. This Fisher SLM was interpolated with SLMs trained on grammar generated corpora of different sizes. Optimal weights were calculated for each interpolated model separately. Almost all optimal weights for the Fisher SLM were around $\lambda = 0.33$. The perplexity curves calculated on the held out set and the WER of the test set are almost identical to the graphs in Figure 2 of the previous section. Here 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%. Table 2 contains the value that would have been selected using the held out set perplexity as a decision criterion.

a single room.).

3.5. Sentence selection using perplexity filtering

We collected a corpus of 2.4 M words from the Internet using the web data collection tools of the university of Washington [15]. We segmented it into 188,909 sentences. This corpus contains text that was returned by a search engine when presented with trigrams as search queries. In this context as well as with the diverse Fisher corpus it might be rewarding to extract all relevant sentences of the corpus and give them a higher weight than the remaining sentences. We built a Seed SLM from our 10M grammar generated sentences and executed sentence selection on the Fisher corpus. In a second round we used the interpolated Grammar-Fisher SLM as a seed SLM and executed sentence selection on the web data. For the sentence selection, we used the following algorithm [16]:

- build a language model from a seed corpus LM_{Seed}
- build a language model from the large corpus LM_{Large}
- calculate $PP_{Rel} = PP_{LM_{Seed}}/PP_{LM_{Large}}$ 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 1 to derive λs.

$$LM_{Filt} = (1 - \lambda_{Sel} - \lambda_{NotSel})LM_{Seed} + \\ + \lambda_{Sel}LM_{Selected} + \lambda_{NotSel}LM_{NotSelected}$$

Table 3: *Speech recognition results on the test set.*

Language model	WER
grammar LM interpolated with Fisher and Web LM	22.0
2 rounds of ppl-filtering	21.5

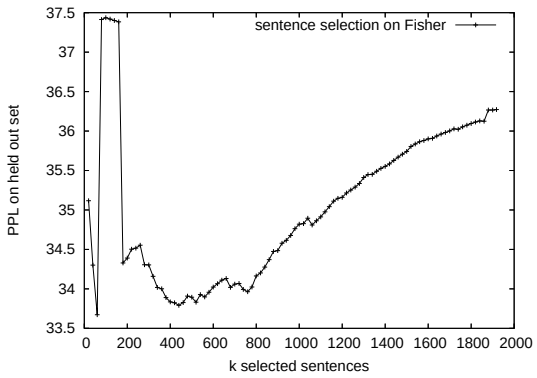
The first row of table 3 shows the test set WER obtained interpolating SLMs built from the 10M sentences generated by the grammar, the Fisher corpus and the web corpus. Interpolating with the web corpus gave us a relative improvement of 3%. Applying perplexity filtering to both the Fisher corpus and the web data improved the test set WER by another 2% relative. The graph in figure 3 shows the change in perplexity as the number of selected sentences increases. After a noisy initial segment[4] the perplexity stabilises at a certain level. It goes down to a minimum and slowly increases again, as a growing number of sentences are selected. The improvements that can be gained by sentence selection are higher in inhomogeneous corpora. As a number of SLMs built from different corpora are interpolated with the Seed LM, the little improvements gained by applying sentence selection to each of them can add up to a considerable decrease in WER.

4. Discussion of results

We compared speech recognition results using a recognition grammar with different kinds of statistical language models (SLMs).

[4]The first part of the graph is very noisy, as 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 3: *Perplexity of interpolated SLM calculated on the held out set.*



All SLMs used in this paper outperformed the recognition grammar network in terms of word error rates (WER) on the test set. Our best models nearly halved the WER.

A SLM trained on a corpus of 30k sentences generated by a grammar decreased WER by 29% relative compared to using the grammar directly as a recognition network. Our results suggest that a grammar that is written for the purpose of corpus generation, can be much less precise than a grammar intended to be used as a recognition network. In a second experiment we compared the results of the grammar generated SLMs with a SLM trained on an "in-domain" corpus consisting of transcriptions of Wizard Of Oz (WOZ) experiments. Both Models had similar performance. In a third series of experiments we interpolated our grammar generated SLMs with SLMs trained on a standard spontaneous speech corpus consisting of a large collection of dialogues about different topics. We gained a further relative improvement in WER of 21%. However, the absolute performance was not quite as good as that achieved by interpolating with the Wizard Of Oz SACTI data (22.7 vs 21.2). Finally we introduced a sentence selection algorithm and applied it to a spontaneous speech corpus and a corpus collected from the web. Adding these final refinements resulted in a performance very close to that obtained using the WOZ data but without the cost (21.5 vs 21.2).

Overall, we conclude that by using synthetically generated corpora interpolated with general corpora of real world data, effective language models can be built for boot-strapping a spoken dialogue system without recourse to expensive WOZ data collections. However, to get the best performance, the real world data needs to be filtered by a scheme such as the perplexity-based method described here.

5. Acknowledgements

This paper was supported by the EU Framework 6 TALK Project (507802). I would like to thank Rebecca Jonson for many helpful discussions.

6. References

[1] 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.

[2] 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.

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

[4] 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 ICASSP*, Salt Lake City, Utah, 2001.

[5] Stephanie Seneff, Chao Wang, and Timothy J. Hazen, "Automatic induction of n-gram language models from a natural language grammar," in *Proceedings of the Eurospeech*, Geneva, Switzerland, 2003.

[6] Rebecca Jonson, "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.

[7] Steve Young, *ATK: An Application Toolkit for HTK, Version 1.4.1*, Cambridge University Engineering Dept, July 2004, http://htk.eng.cam.ac.uk/develop/atk.shtml.

[8] Stanly F. Chen and Joshua T. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech and Language*, vol. 13, pp. 359–397, 1999.

[9] 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*, Cambridge University Engineering Dept, December 2002, http://www.htk.eng.cam.ac.uk.

[10] 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.

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

[12] Andreas Stolcke, "SRILM - An extensible language modeling toolkit," in *Proceedings of the ICSLP*, Denver, Colorado, September 2002, http://www.speech.sri.com/projects/srilm/.

[13] 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.

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

[15] Tim Ng, Mari Ostendorf, Mei-Yuh Hwang, Ivan Bulyko, Manhung Siu, and Xin Lei, "Web-data augmented language model for mandarin speech recognition," in *Proceedings of the ICASSP*, Philadelphia, US, 2005, http://ssli.ee.washington.edu/projects/ears/WebData/web_data_collection.html.

[16] Chze Ling Wee, "Web data for language modelling of conversational telephone speech," M.S. thesis, Cambridge University Engineering Dept, 2004.