Language Modelling and Spoken Dialogue Systems - the ARISE experience

P. Baggia[1], A. Kellner[2], G. Pérennou[3], C. Popovici[4], J. Sturm[5], F. Wessel[6]
[1]CSELT, I-10148 Torino, Italy, [2]Philips Research Lab., Aachen, Germany,
[3]IRIT, Toulouse, France, [4]ICI, Bucuresti, Romania,
[5]Univ. of Nijmegen, Nijmegen, The Netherlands, [6]RWTH Aachen, Aachen, Germany
baggia@cselt.it

ABSTRACT

The aim of this paper is to describe the experiences gained in the field of language modelling during the LE-3 ARISE (Automatic Railway Information Systems for Europe) project. All of the different techniques presented in this paper are related to the field of Spoken Dialogue Systems, and they cope with the issues of limited amount of training material and the exploitation of the constraints available in a dialogue system. The results obtained may be useful for the future development of similar applications.

Keywords: language modelling, spoken dialogue system, speech recognition

1. INTRODUCTION

This paper describes many techniques devoted to a better adaptation of the language modelling component of a Spoken Dialogue System (SDS). In the ARISE project [3], four prototypes of train timetable information SDS were developed, for three different languages. The Dutch (ARISE-NL) and one of the French systems (ARISE-FR1) were based on Philips technology, the other French system (ARISE-FR2) on LIMSI technology and the Italian one (ARISE-IT) on CSELT technology. For each system, the size of the vocabulary and of the training material are described.
A technique for robust understanding, based on conceptual segments [8] is presented. It is able to solve phrase and word ambiguity, highly relevant for the French language. Then, many techniques have been developed to increase the robustness of the language models (LMs) by using the constraints available in a dialogue strategy [4,10,13], such as interpolation of LMs, automatic classification of dialogues states, and the use of grammar-driven smoothing. Moreover, another constraint faced by the SDS developers is the limited amount of data for LM training. A common trouble is to use the transcriptions of data acquired in the field or data from similar projects[1], and to progressively add new material. The use of written queries was also exploited, as an approximation of the linguistic interactions of a SDS, and an on-line update of the LMs.

[1] The LIMSI system was bootstrapped on the data acquired in Mask project.

2. INPUT DATA

2.1 Vocabulary

For all the systems, the vocabulary was class-based according to the semantically relevant word classes, e.g. city and station names, weekdays, numbers, etc. This implies that the LMs were class-based too, so that the probability of each word in a class was equally distributed.
Besides the semantically based word classes, CSELT further classified the remaining words by using a clustering algorithm described in [6]. The improvements obtained not only reduced the WER, see [9], but even the LMs size was decreased, (from 359 to 120 classes). Philips used the information contained in the language understanding grammar to automatically cluster the vocabulary words [5]. Some non-terminal symbols of the grammar can be expanded to many different terminal symbols (single words), for example city names or numbers. These words may then be grouped into one class. All other words form a class on their own. This approach automatically generates classes for words, which are expected to appear in the same word context.
Table 1 shows the vocabulary size and the number of railway station for each system.

System	Words	Station names
ARISE-FR1	1,293	600
ARISE-FR2	1,794	500
ARISE-IT	3,475	2,533
ARISE-NL	1044	500

Table 1 – Vocabulary size

2.2 Training and Test Corpora

Table 2 shows the size of the training and test corpora used during the development of the LMs. The size is given in the number of dialogues and sentences.

System	Dialogues		Sentences	
	Train	Test	Train	Test
ARISE-FR1	-	-	9,520	-
ARISE-FR2	6,122	130	72,392	1,515
ARISE-IT	1,683	226	15,575	2,040
ARISE-NL	7,756	453	73,402	4,330

Table 2 - Training and test data

The evaluation of the ARISE-FR1 was done only at the dialogue level, by analysing the Dialogue Success and Failure rates.

3. CONCEPTUAL SEGMENTS

Conceptual segment (CS) modelling provides an alternative approach to robust automatic understanding. This technique was tested by IRIT in the ARISE-FR1 system. It is based on the hypothesis that each sentence $W=w_1 \ldots w_N$ is emitted by a two level Markov source.

At the first level $S=S_1 \ldots S_K$ is generated. Each S_i is a sub-Markov source which emits CSs, that are word sequences W_i. It corresponds to a special twofold interpretation domain where illocutionary and referential values are considered. For example:

W = No from Paris tomorrow morning (1)

may be emitted by

S_1 = NO-source (W_1=*no*),
S_2 = DEPARTURE-CITY-source (W_2=*from Paris*)
S_3 = DEPARTURE-DATE-TIME or
S'_3= ARRIVAL-DATE-TIME (W_3=*tomorrow morning*)

In this case the specific role of S_1 is confined to the generation of an illocutionary value (the negation), S_2 generates a referential value concerning the departure town, and S_3 (or S'_3) generates two referential values (date and time). Each sub-source may have itself several levels. So the overall model[2] is recursive and formally equivalent to a *stochastic ATN*.

One of the advantages is the possibility to use a common model for segments like TIME, CITY… in S_2, S_3… This grammar is trained on spontaneous dialogue corpora representative of the application.

In languages such as French, where the recognition of the semantic cases requires very often a contextual analysis, the CS appears as an efficient method to face word and phrase ambiguity problems. Let us consider a simple example: the French word *vers* has two main meanings in the context of timetable applications: *vers*$_1$*="approximatively"* and *vers*$_2$*= "to"*.

Consider the two following examples

W = Non de Paris <u>demain vers midi</u> (2)
"no from Paris tomorrow about twelve o'clock"
W = Non de Paris <u>vers Toulouse</u> (3)
"no from Paris to Toulouse"

Using the CS model trained on a representative corpus the stochastic decoding will recognise *<u>demain vers midi</u>* as emitted by S_3 where *vers* has the value *vers*$_1$ and *<u>vers Toulouse</u>* as emitted by S_4 = ARRIVAL-CITY-source where *vers* has the value *vers*$_2$.

Most of the ambiguities which require a context larger than the CS can be solved by the first level Markov model, that is for example in (2): the probability of the sequence $S_1S_2S_3$ will be greater than the one of $S_1S_2S'_3$ (given a representative training corpus).

[2] The Philips dialogue system, used for the ARISE-FR1 system, represents the recursive conceptual model by a *CF-stochastic and attributed grammar* [1].

The model is particularly appropriate for designing robust automatic understanding modules: out of domain segments can be taken into account by a filler source or useless words can be detected and ignored.

4. DIALOGUE-STATE DEPENDENT LANGUAGE MODELS

To take advantage of the dynamic behaviour of a spoken interaction, the dialogue-state was considered by building a set of dialogue-state dependent LMs [4,9,13]. Therefore the training corpus was split according to each dialogue state, where the dialogue-state is derived from the system prompt. At run-time, depending on the point in the dialogue, the corresponding LM can be activated for the recognition of the next utterance. However, due to the large number of possible dialogue states, the creation of a different LM for each of them can cause some troubles, as some of them are very rarely observed even in a large corpus. Moreover, there are groups of dialogue states that lead to very similar user answers. To overcome this problem, different techniques were investigated, and they are described in the following Sections. The experimental results are given in terms of perplexity (PP), Word Error Rate (WER), Concept Error Rate[3] (CER).

4.1 Interpolated LMs

RWTH proposed to use a linear interpolation of all dialogue-state dependent LMs and a global LM for each dialogue state [13]. In doing so, supplementary information contained in the different language models can be exploited. The main problem with this model is the large number of interpolation weights. In order to train the interpolation weights, the Expectation-Maximization-Algorithm was used to minimise the Leaving-One-Out perplexity on the LM training corpus.

ARISE-NL	PP	WER(del/ins)	CER(del/ins)
Baseline	11.8	14.0 (2.0/2.6)	14.7 (2.3/5.1)
All dial-states	9.3	13.2 (1.8/2.5)	14.0 (2.1/5.6)
Generalisation	9.2	13.2 (1.9/2.5)	14.0 (2.0/5.6)

Table 3 - LM results for the ARISE-NL

In order to compare the interpolated model with the generalisation of dialogue states, an automatic text-clustering algorithm was used to merge the training corpora of several dialogue states until a sufficient amount of training material for each LM is obtained. The clustering algorithm, which was used, works favourably well in terms of reducing the perplexity. The optimal number of clusters, on the other hand, cannot be determined automatically. Ideally, the clustering algorithm should choose that number of clusters, which minimises the perplexity on new, previously unseen

[3] The Concept Error Rate (CER) metric, see [2], is similar to WER, but at the parsing level. CER accounts for insertions/deletions/substitutions in a string of concepts, instead of words. A concept is a semantic attribute-value pair, e.g. <arrival_station = Bonn>.

data. In order to simulate unseen data the clustering algorithm was extended [14]. Instead of minimising the log-likelihood of the cluster dependent unigram models, now the log-likelihood of the cluster dependent Leaving-One-Out unigram models is minimised. With the Leaving-One-Out unigram models the algorithm merges dialogue-states until no corpus is moved any more and the Leaving-One-Out perplexity on the training data is minimised.

The best results were achieved using a combination of these methods. Although the linear interpolation of the generalised language models with the global language model has not improved the WER, a significantly smaller number of language models was interpolated and the computing time was thus reduced. For these three combined models RWTH also measured the CER. With the combined model the CER was reduced by 5% relative, from 14.7% with a dialogue-state independent language model to 14.0% with our best dialogue-state dependent model.

4.2 Auto-classification of dialogues states

For the Italian system, the first approach tested was to manually cluster the dialogue states on the basis of declarative rules [9]. The results obtained are very interesting: 27% reduction of PP, 7% of WER and 13% of CER, see Table 4 (*Baseline* and *Manual Cls* Rows). Then, an automatic approach for clustering dialogue states was also investigated [11]. The method was based on the mutual information between two clusters of dialogue-states, and it even suggests an appropriate number of dialogue-state clusters (73 dialogue states were mapped into 7 dialogue-state dependent LMs). By using these techniques a full automatic procedure to create the dialogue-state dependent LM was developed. Table 4 shows that the *Autom. Cls* even slightly improve the performance of the system.

ARISE-it	PP	WER(del/ins)	CER(del/ins)
Baseline	21.7	26.2 (7.6/7.5)	24.8 (8.9/13.2)
Manual Cls	15.7	24.4 (7.2/6.9)	21.5 (7.9/10.9)
Autom. Cls.	15.6	24.3 (7.0/7.0)	21.5 (8.0/10.8)
Robust LMs	15.8	24.2 (7.1/6.7)	21.3 (7.9/10.7)

Table 4 - LM results for the ARISE-IT

4.3 Use of additional knowledge

On the ARISE-NL corpus, a test was done for modelling the answers to the first system prompt: "*from where to where do you want to travel?*". The experiment showed that an adaptation of grammar, lexicon and LM, improve the performance with 15.9% on the sentences which mainly contain station names. However, it reduced the performance of the sentences that contain also date and time expressions. The conclusion is that over-specialised LM, used in a mixed-initiative system, can penalise the more informative sentences.

For these reasons in the ARISE-IT, the robustness of the dialogue-state dependent LMs was increased, by the use of simple grammars to reinforce the more informative sentences. The generation grammars are designed to produce linguistically correct sentences, which are appropriate for one or more dialogue states. A sort of grammar-driven back-off smoothing was realised [10]. Using this technique, the advantages of the dialogue-state dependent LMs were maintained, but even the more complex sentences, which are involved in a mixed-initiative dialogue strategy, were improved.

The results in Table 4 (*Robust LMs* compared with *Baseline*) shows a global reduction of 27% PP, 8% of WER, and 15% of CER. The improvement of this method is overall relevant when the amount of training material is small.

5. BOOTSTRAP LANGUAGE MODELS

An important issue in SDSs is how to create an initial LM for a new application when no or only little training data is available.

Philips exploited the task-specific information contained in the language understanding grammar in order to create the bootstrap LM. This was done by automatically increasing the amount of training data for a new application [5]. The grammar covers typical formulations for the application and thus implicitly contains information about expected user utterances. Monte-Carlo methods are used to randomly choose rules at the branching points of the grammar and thereby create random sentences, which are covered by the grammar. This results in an artificial corpus from which an N-gram language model can be trained. If a small amount of training material is available this can be used to obtain weights for the different rules which leads to a more realistic corpus. Still, the grammar only covers the meaningful parts of the user input and does not model the meaningless filler phrases.

Therefore, the language model trained on the artificial corpus and the class-based language model should be combined in a hierarchical way: into a fill-up model. If an N-gram was not seen in the top-level model, its likelihood is derived from the model on the next level, which in turn may fall back to lower levels.

Train Sentences	Word	Classes	Gramm.	Gramm. & Class
0	21.5	21.3	16.9	16.5
100	20.3	17.4	16.0	15.8
1,000	17.9	15.4	15.1	14.5

Table 5 - Comparison of initial LMs (WER)

As can be seen from the WER results reported in Table 5, the class-based LM performs better than the usual word bigram model, but the model derived from the grammar gives a much better initial performance. Integrating the two approaches into a fill-up model combines the strengths of the two models and leads to the best results.

After bootstrapping with such an initial model, online adaptation of the stochastic grammar and the recognizer LM helps to continuously improve the performance of the system. One problem in unsupervised adaptation are misrecognitions that may lead to a deterioration of the system performance because of error reinforcement. In [12] some methods to avoid this effect were examined.

An approach that turned out to be quite efficient is to use multiple sentence hypotheses from an N-best list as adaptation material. For that, each hypothesis *i* gets a weight Ω_i derived from its a-posteriori likelihood such that the weights add up to 1. Every sentence now contributes to the adaptation material according to its weight Ω_i. The idea behind this approach is that well-understood parts of a sentence will occur in most of the hypotheses of an N-best list, whereas for misrecognitions there will usually be several alternatives. Thus, the effect of a recognition error is distributed over several competing hypotheses and does not result in strong error reinforcement.

CSELT used, for the training of a bootstrap LM, all the written sentences generated during the development phase of a dialogue system [7]. A LM trained on that material does not reach the performance of the one trained on acquired data, but the development material already has the dialogue dependency properties (necessary for the creation of dialogue-step dependent LMs) and suggest a quite good frequency distribution of users answers. Another advantage in using the written sentence corpora is that they do not need extra-time to be obtained. The performance of this bootstrap LMs gives quite good results and may be improved by adding the acquired material.

6. FINAL REMARKS

Although there were different languages (Dutch, French, and Italian), vocabulary sizes and used techniques, in the ARISE project, the following common guidelines could be distinguished:

- the classification of vocabulary words,
- the exploitation of dialogue states for the generation of dialogue-state dependent LMs,
- the clustering of dialogue states, with minor differences in the techniques used (Mutual Information vs. Leaving-One-Out),
- the use of generation grammars both for increasing the robustness of the LMs and for creating a bootstrap LM

Using these techniques, the improvements obtained in the different systems are comparable: a relevant reduction of perplexity (about 25%) and a reduction of recognition and understanding error rate (about 10%). All of these techniques may be useful for the developers of task-oriented dialogue systems.

4. REFERENCES

[1] Aust, H., M. Oerder, F. Seide, and V. Steinbiss (1995). The Philips automatic train timetable information system. *Speech Communication*, 17 (3-4) pp. 249-262.

[2] Boros, M., W. Eckert, F. Gallwitz, G. Görz, G. Hanrieder, and H. Niemann (1996). Towards Understanding Spontaneous Speech: Word Accuracy vs. Concept Accuracy. *Proceedings of ICSLP'98*, vol. 6, Banff, Canada, vol. 2, pp. 1009-1012.

[3] den Os, E., L. Boves, L. Lamel, and P. Baggia (1999). Overview of the ARISE Project. *In this Proceedings*.

[4] Drenth, E.W., and B. Rueber (1997). Context-dependent probability adaptation in speech understanding. *Computer Speech and Language*, 11(3), pp. 225-252.

[5] Kellner, A., (1998). Initial language models for spoken dialogue systems. *Proceedings of ICASSP'98*, Seattle,WA, pp. 185-188.

[6] Moisa, L., and E. Giachin (1995). Automatic Clustering of Words for Probabilistic Language Models. *Proceedings of EUROSPEECH-95*, Madrid, vol. 2, pp. 1249-1253.

[7] Moisa L., P. Baggia, C. Popovici (1998). Language Modelling in EasyDial. *Proceedings of IVTTA'98*, Torino, Italy, pp. 179-184.

[8] Pérennou, G, M. de Calmès, A. Lavelle, and R. Tronel (1998). Un systeme de dialogue oral spontane pour l'acces telephonique aux informations d'horaire de train, *Proceedings of Nimes 98*.

[9] Popovici, C., and P. Baggia (1997). Specialized Language Models using Dialogue Predictions. *Proceedings of ICASSP'97*, München, vol. 2, pp. 815-818.

[10] Popovici, C., and P. Baggia (1997). Language Modelling for Task-Oriented Domains. *Proceedings of the EUROSPEECH'97*, Rhodes, Greece, pp. 1459-1462.

[11] Popovici, C., P. Baggia, P. Laface, L. Moisa (1998). Automatic Classification of Dialogue Context for Dialogue Prediction. *Proceedings of ICSLP'98*, Sydney, vol. 2, pp. 397-400.

[12] Souvignier, B., and A. Kellner (1998). Online adaptation for language models in spoken dialogue systems. *Proceedings of ICSLP'98*, vol. 6, Sydney, Australia, pp. 2323-2326.

[13] Wessel, F., and A. Baader (1999). Robust Dialogue-State Dependent Language Modeling Using Leaving-One-Out. *Proceedings of ICASSP'99*, Phoenix, AZ, vol. II, pp. 741-744.

[14] Wessel, F., A. Baader, and H. Ney (1999). A Comparison of Dialogue-State Dependent Language Models. To appear in *Proceedings of ECSA Workshop on Interactive Dialogue in Multi-Modal Systems*, Irsee, Germany.