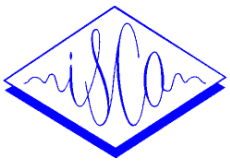

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

DYNAMIC SELECTION OF LANGUAGE MODELS IN A DIALOGUE SYSTEM

Y. Estève, F. Béchet, R. de Mori
LIA - University of Avignon - France
{yannick.esteve,frederic.bechet,renato.demori}@lia.univ-avignon.fr

ABSTRACT

This paper describes a method for building statistical Language Models (LMs) dedicated to specific dialogue situations. The architecture of the speech recognition system proposed uses several LMs. The first stage of this system, consists of producing a word-lattice from a given sentence uttered by a speaker. A general LM calculates a sentence-hypothesis. Then, in a second stage, the system chooses a specialized LM according to the word-lattice and the previous hypothesis. Another decoding process is performed using this specialized LM in order to produce a new sentence- hypothesis. Finally, a decision-module processes these two hypotheses in order to assign three confidence levels to the sentence-hypothesis produced. These confidence levels can be used by the dialogue manager in order to improve the dialogue, by asking a confirmation to the speaker when a sentence is labeled ambiguous.

This research is supported by France Telecom's R&D under the contract 971B427.

1. INTRODUCTION

Interactive vocal servers are one of the main application of speech recognition techniques. These servers are usually dedicated to process a specific task within a limited and well identified semantic domain. The limited size of the vocabulary needed by such application results in good performance from the speech recognition module and makes the use of these servers efficient.

Language Models (LMs) used in such speech recognizers are usually n-gram models. They are trained on text corpora made of dialogue transcriptions between a human and a machine for the same application area.

Thanks to the small size of the vocabulary used, good results in perplexity can be obtained even with small training corpora. Different kind of LMs, based on n-grams models, have been proposed in [6] and [3] in order to define classes of recognition units adapted to the particularities of speech recognition for vocal servers.

These particularities correspond to various dialogue-situations which are constant whatever the application targeted by the server is. A careful examination of the sentences uttered by users of a vocal server allows us to classify them into several categories which are representative of a specific dialogue-situation. For example, the variability of the sentences uttered by users to answer direct questions is very limited. To model these *key-sentences*, LMs integrating larger contexts than the usual 2-gram or 3-gram have been proposed in [1].

In order to take into account these various dialogue-situations, we propose in this article a method which calculates, in addition of a general LM, several sub-LMs dedicated to process some specific dialogue-situations.

These specific LMs have two main justifications:

1. they can model more precisely some regularities in the interventions of the user;
2. they can identify the kind of sentence which has been uttered (confirmation, negation, general question, ...). These information can increase the robustness of the dialogue manager, even when the speech recognition module fails.

2. DESCRIPTION OF THE APPLICATION

This work is based on a corpus of dialogue transcriptions uttered by several speakers, the AGS corpus from France Telecom R&D [5]. This corpus contains phone-dialogues between users and two vocal servers dedicated to process weather broadcast and job hunting.

This corpus contains:

- The AGS training-corpus, made of 9842 sentences (which have been uttered by several speakers and transcribed manually). The vocabulary contains 823 different words.
- A test-set of 393 sentences associated to 393 word-lattices produced by France Telecom R&D's speech recognition system (the lattices contain acoustic scores for each word).

It is important to note that, in the AGS corpus, some sentences have a very high frequency of utterance (10 sentences represent 20% of all the occurrences). These sentences are usually user's answers to a direct question asked by the dialogue system.

A manual study of the AGS corpus shows the low variety of sentence patterns represented in it. According to different criteria, it is possible to split the corpus into sub-corpora which are representative of a specific dialogue-

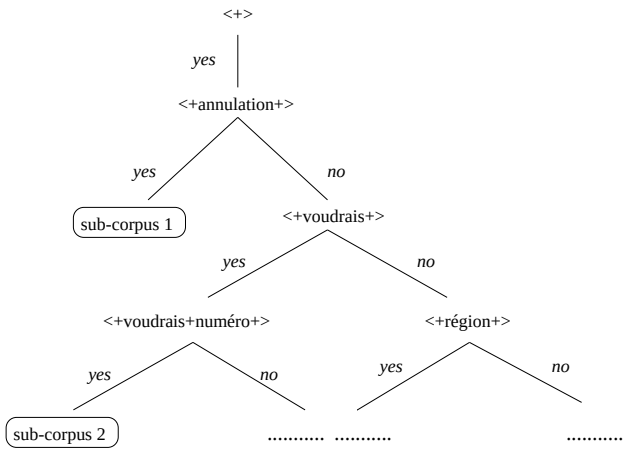

Figure 1: Semantic Classification Tree

situation. For example, we can group together the sentences related to queries on the same semantic domain (weather server, job-hunting server), or sharing the same syntactic pattern (for example, the sentences beginning with *Je voudrais le numéro de . . . /I would like the number of . . .*). Moreover, direct answers made by the user to the queries of the system (*oui*/*yes*, *non*/*no*, *annulation*/*cancel*, ...) can be easily detected.

According to the criteria used, the size and the kind of sub-corpora will vary a lot. We have chosen a strategy based on a particular kind of decision tree, called Semantic Classification Tree (SCT) [7]. After the training process of the tree, a sub-corpus is attached to each node in a hierarchical way: the whole corpus at the root level and the most specific sub-corpora in the leaves.

3. SPLITTING THE TRAINING CORPUS

The SCTs automatically build regular expressions and each one can split the training corpus into two sub-corpora (the sentences which match the regular expression, and those that don't) (see Figure 1). For the growing process, we need a set of training samples, a set of questions, and a criterion to choose the best question for each node. We will quickly present these three parameters.

3.1. Set of samples

The samples used to build the SCT are the sentences of the AGS corpus. These sentences have been tagged by means of a Part-Of-Speech tagger developed at the LIA [8]. Then, each word of a given sentence is replaced by its lemma in order to increase the generalization power of the learning process.

3.2. Set of questions

In the SCTs, questions are generated with a lexicon and a set of three symbols: <,>,+. The symbols <,> represent the begin and the end of a sentence. The symbol + is a non-empty sequence of words.

During the growing process of the SCT, each node of the tree is associated with a regular expression called the *Known Structure* (*KS*) and a set of samples containing all the sentences which satisfy this regular expression. At the beginning of the growing process, the root of the tree is associated to the *KS* $< + >$ and to the entire training corpus. A *KS* also records the position of the last item that was introduced in it.

The *KS* of a node and the set L composed of the lexical entries of the lexicon will give rise to several new regular expressions by replacing in *KS* a gap with elements of L. More precisely:

- each element i of L produces four different patterns: $\{i\}$,$\{+i\}$,$\{i+\}$,$\{+i+\}$
- each of the generated patterns replaces in *KS* the gap situated respectively at the right and at the left of the element of the last item introduced.

With this method, a given *KS* generates a maximum of $4 \times |L| \times 2$ regular expressions. A 2K word lexicon will produce for each *KS*, $4 \times 2000 \times 2 = 16K$ different new regular expression.

Each regular expression splits the set of samples associated to the node in two: the set of the samples that match the regular expression and the set of those that don't. A regular expression is therefore seen as a yes-no question.

3.3. Selection criterion

To select a question among all the possible ones, we have chosen to use the criterion of perplexity-reduction between the LM trained on a node's corpus and the ones trained on its children. After having split the AGS corpus into a training corpus and a development corpus, we process as follow:

For a given node and a given question, we calculate four sub-corpora:

1. A_{yes}: sentences of the training corpus which match the KS,
2. A_{no}: sentences of the training corpus which don't match the KS,
3. D_{yes}: sentences of the development corpus which match the KS,
4. D_{no}: sentences of the development corpus which don't match the KS.

Then, we train three 2-gram LMs: $M_{A_{yes}}$,$M_{A_{no}}$,$M_{A_{yes}+A_{no}}$ on the training corpora: A_{yes},A_{no},A_{yes+no} and we calculate the following perplexities (with a function PP which compute the perplexity):

- $P_{yes} = PP(M_{A_{yes}}, D_{yes})$
- $P_{no} = PP(M_{A_{no}}, D_{no})$

The question chosen is the one which minimize the value of $P_{yes} + P_{no}$ with respect to the following constraint: $P_{yes} + P_{no} < 2 * (P_{ref} + \varepsilon)$ (where ε controls the expanding of the tree).

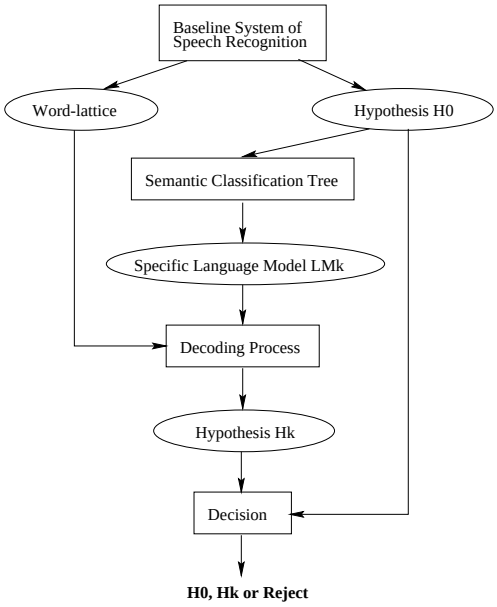

Figure 2: System Architecture

The construction of the tree is stopped when the size of the sub-corpora become too small or if we can't find a question which contents the previous constraint.

4. TRAINING SPECIFIC LANGUAGE MODELS

To each node of the SCT obtained, we can associate a bigram LM trained on all the sentences which match the KS of this node. Unfortunately, due to the small size of the AGS corpus, most of the sub-corpora obtained are too small to train a bigram model which won't be over-trained.

To avoid this problem, we have decided to use a LM adaptation technique presented in [4]. In this case, the general model is the one trained on the entire training corpus, and the adaptation corpus is the one trained on the sub-corpus of a node.

5. SYSTEM ARCHITECTURE

During the recognition process, the choice of a specific LM is tightly linked to the state of the dialogue. Unfortunately, this information isn't available in the AGS corpus. In order to select a LM among all those calculated previously, we developed a speech recognition system which works in two steps: a first sentence-hypothesis, called H_0, is calculated by means of a general LM, called LM_0. Then, this hypothesis is used to select a specific LM, called LM_k according to the tree presented in the previous section.

The architecture of our system is described in Figure 2. The hypothesis H_0 is extracted from a word lattice calculated by the France Telecom R&D speech recognition system. Once LM_k is chosen, a second decoding process is performed using this LM to produce the hypothesis H_k.

5.1. Selection of the specific LM

To choose LM_k, we parse the SCT built previously with the hypothesis H_0. The parsing process is straightforward, it consists on traversing the tree, starting at the root until a leaf is reached. For each node N visited, if H_0 matches the regular expression of N, the next node to visit is the daughter of N labeled *yes*, otherwise, it is the daughter labeled *no*.

When a leaf is reached, this process stopped and the LM attached to the leaf becomes LM_k. We did some experiments in order to test the robustness of this method in relation with word-errors in H_0. The results showed that this method chooses the correct specific LM associated to the sentence uttered by the speaker with a good accuracy (about 90%).

Once LM_k is chosen, a new decoding process using it is performed in order to produce the second hypothesis H_k. These two hypotheses model different information: if the uttered sentence is very close to *key-sentences* related to specific dialogue-situations seen in the training corpus, H_k will be more relevant. On the other hand, if the sentence represents a general request from the user, LM_k can't generalize enough to process efficiently the word-lattice and H_0 will be closer to the sentence uttered.

At this step of the process, a decision-phase is necessary in order to select one of the hypotheses or reject both if the confidence measures on the hypotheses are too low.

5.2. Decision module

The decision-module is based on rules which compare the features (acoustic scores,bigram scores, syntactic structure, number of words, differences, specific language model used, ...) of the two hypotheses H_0 and H_k. These rules are automatically obtained by means of a decision tree. For the growing process of this tree (using a standard algorithm), a development corpus have been made : it contains, for a set of sentences from the AGS corpus, both hypotheses H_0 and H_k produced by our system and a label which can be either ACCEPT_h_0, ACCEPT_h_k or REJECT_BOTH, according to the word-error rate of both hypotheses.

The questions used to split the corpus at each node are related to various sources of information like the acoustic and linguistic scores, the LM_k chosen, the syntactic pattern of the hypotheses, the length of the hypotheses or the number of words in common. The splitting criteria chosen is the Gini impurity criteria proposed in [2].

During the recognition process, this tree is used when the two hypotheses H_0 and H_k are different: the tree is parsed with all the features attached to H_0 and H_k and the decision taken corresponds to the label of the leaf reached at the end of this parsing process.

5.3. Confidence levels

Having hypotheses based on different sources of information allows us to attach a confidence measure to the solution proposed by the system. We have defined three confidence levels:

1. Level 1: this corresponds to the highest confidence level. When $H_0 = H_k$ we believe that the hypothe-

	Baseline system	LIA system without reject	LIA system with rejects
w.e.r	30.84%	30.74%	28.34%
s.e.r	52.67%	52.41%	49.16%
Accepted Sentences	393	393	358
Rejected	0%	0%	8.9%

Table 1: Comparisons between the baseline system and the proposed system

sis produced is relevant enough because two different decoding process have produced the same solution.

2. Level 2: this is the middle confidence level. This level indicates that the hypotheses H_0 and H_1 are different. The decision module proposed one of them as the solution (tagged with a confidence level of 2), the other one is suppressed.
3. Level 3: This corresponds to the rejection-level. The decision-module chooses to reject both hypotheses by lack of confidence in their different features. In this case, the recognition system won't propose a solution and the dialogue-manager will ask more information to the user.

6. EXPERIMENTS

Experiments were carried out using the 393 word-lattices hypotheses produced by the France Telecom R&D's speech recognition system. The training corpus is the AGS corpus which contains 9842 sentences for a total of 49610 words (the vocabulary contains 823 entries). AGS was used to build the SCT and the LMs attached to each node.

Because of the small number of word-lattices, we couldn't divide this set into two sub-sets (one for the training of the decision-module, one for the tests). So we decided to use the *leave-one-out* method, which consists in training the decision-module on 392 word-lattices and making the test on the word-lattice left over. This have been done for each word-lattice.

Comparisons between the baseline system (France Telecom R&D's system) and the proposed system are summarized in Table 1 (*w.e.r* stands for *word error rate*, *s.e.r* stands for *sentence error rate*, and *Rejected* represents the word-lattices rejected by our system only, the baseline system doesn't offer this possibility).

The new system has rejected 8.9% word-lattices (tagged with a confidence level 3 by the decision-module). The reduction in word error rate and sentence error rate comes mainly from these rejects.

It is important to note that the confidence level given to each solution calculated by our system is truly significant: Table 2 shows the results in term of word error rate and sentence error rate according to the confidence level given to the sentence (*%sentence* stands for the percentage of sentences tagged with a given level).

	%sentence	w.e.r	s.e.r
level 1	33.84	15.61	16.54
level 2	57.25	31.79	68.44
level 3 (reject)	8.9	59.23	85.71

Table 2: Performances according to the confidence level

7. CONCLUSION

We propose a method which splits a training corpus into sub-corpora in order to train specific language models. Then, we show how to use these specific language models to improve a speech recognition baseline system. Our method allows us to give to the final sentence-hypothesis three confidence levels. Incorrect hypotheses can be detected and rejected: this can be very useful for the dialogue manager in order to ask the user a confirmation instead of driving the dialogue in a wrong direction.

The experiments show that this method obtains interesting results: detecting and rejecting wrong sentences and choosing the best hypothesis allow us to reduce word and sentence error rates. Moreover, experiments show the reliability of the confidence levels proposed.

Future work concerns the ability of identifying correct substrings in a sentence hypothesis by means of several LMs which model different dialogue situations. Giving a confidence measure, not only to the entire sentence, but also to items within a sentence would allow us to increase the robustness of the dialogue process.

REFERENCES

[1] Nasr A., Estève Y., Béchet F., Spriet T., and De Mori R. A language model combining n-grams and stochastic finite state automata. In *Eurospeech*, Budapest, 1999.

[2] L Breiman, J Friedman, R Ohlsen, and C Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[3] Beaujard C. and Jardino M. Un modèle de langage de langage mixte basé sur la similarité des mots dans un système de reconnaissance de la parole. In *JEP*, 1998.

[4] Janiszek D., , De Mori R., Béchet F., Matrouf D., and Mokbel C. New language model adaptation algorithm based on the definition of cardinal distance. In *ESCA Workshop on Interactive Dialogue in Multi-Modal Systems*, Kloster-Irsee, Germany, 1999.

[5] Sadek D., Ferrieux A., Cozannet A., Bretier P., Panaget F., and Simonin J. Effective human-computer cooperative spoken dialogue : the AGS demonstrator. In *ICSLP*, Philadelphia, 1996.

[6] Damnati G. Modèles de langage et classification automatique pour la reconnaissance de la parole continue dans un contexte de dialogue oral homme-machine. In *Thèse de l'université d'Avignon et des Pays du Vaucluse*, 2000.

[7] Kuhn R. and De Mori R. The application of semantic classification trees to natural language understanding. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(449-460), 1995.

[8] Spriet T. and El-Bèze M. Introduction of rules into a stochastic approach for language modelling. In Keith Ponting, editor, *Computational Models of Speech Pattern Processing, NATO ASI Series F*, volume 169, pages 350–355, 1998.