Efficient integration of automated speech recognition in the framework of dialogue-based vocal systems

Alex Trutnev

December 10th, 2004
Abstract

In this work, we propose different strategies for efficiently integrating an automated speech recognition module in the framework of a dialogue-based vocal system. The aim is the study of different ways leading to the improvement of the quality and robustness of the recognition.

We first concentrate on the choice of the type of acoustic models that should be used for the speech recognition. Our goal is to evaluate the hypothesis that hybrid acoustic models, in which estimation of frame-based phoneme probabilities is made through artificial neural networks, provide performance results similar to the “classical” Hidden-Markov models using Multi-Gaussian estimations, while being more robust in generalization across tasks. We experimentally show that, due to the size of the parameter space to be explored, it is not always practically possible to achieve a performance comparable to the one of Multi-Gaussian models, and that in fact hybrid models often lead to worse recognition performance.

In a second part, we focus on one of the main limitations of state-of-the-art speech recognition: the inadequacy of the one-best approach to yield a hypothesis corresponding to the right transcription. For that, we explore the solution consisting in producing, during acoustic decoding, a word lattice containing a very large number of hypotheses, that is then filtered by a syntactic analyzer using more sophisticated syntactic models, such as stochastic context-free grammars. The goal of this approach is to yield syntactically correct hypotheses for further processing. More precisely, we study the approach consisting in dynamically tuning the relative importance of the acoustic and language models, resulting in the increase of the lexical and syntactic variability in the word lattice. We identify and experimentally quantify two important drawbacks for this approach: its high computational cost and the impossibility to guarantee that, in practice, the correct solution is indeed present in the lattice.

Finally, we study the problem of the inadequacy of the use of generic linguistic resources (language models and phonetic lexica) to yield robust and efficient recognition results. In this context, we explore the solution consisting in the integration of dynamic phonetic and language models controlled by an associated dialogue model. In this approach, restricted lexicon and language models dependent on the context of the dialogue are used in place of the complete ones. We first experimentally verify that this approach indeed yields a significant increase in speech recognition performance, and we then focus on the problem of producing, for a given application, the adequate dialogue model that can efficiently integrate the speech recognition module. In this perspective, we propose an enhancement of the used dialogue
model prototyping methodology by integrating speech recognition error simulation within the Wizard-of-Oz dialogue simulation. We show that such an approach enables a more complete prototyping of the dialogue model that guarantees a better adequacy of the resulting dialogue model to the targeted vocal application.
Résumé

Ce travail de recherche porte sur l’analyse de différentes stratégies d’intégration efficace d’un module de reconnaissance de la parole au sein d’un système vocal à base de dialogue. L’objectif poursuivi est l’étude de différentes manières de réaliser une telle intégration dans le but d’augmenter la qualité et la robustesse de la reconnaissance.

Nous avons tout d’abord porté notre attention sur le choix des modèles acoustiques nécessaires pour la reconnaissance de la parole. Notre but était d’évaluer l’hypothèse que les modèles acoustiques hybrides, dans lesquels l’estimation des probabilités des phonèmes est faite à l’aide de réseaux de neurones artificiels, mènent à des résultats similaires à ceux des modèles multi-gaussiens “classiques”, tout en étant plus robustes. Nous avons expérimentalement montré qu’en raison de la taille de l’espace de paramètres à explorer, il n’est pas toujours possible d’atteindre des performances comparables à celles des modèles multi-gaussiens, et que, de ce fait, les modèles hybrides mènent souvent à des performances dégradées.

Dans la seconde partie, nous nous sommes concentrés sur l’une des limitations majeures de la reconnaissance de la parole moderne : l’inadéquation de l’approche “one-best” pour fournir l’hypothèse correspondant à la transcription exacte. Dans cette perspective, nous avons exploré la solution qui consiste à produire, pendant le décodage acoustique, un treillis de mots contenant un nombre très élevé d’hypothèses, et à filtrer ensuite ce treillis à l’aide d’une analyse syntaxique à base de modèles syntaxiques plus sophistiqués, comme, par exemple, les grammaires hors-contexte probabilistes. Cette solution a pour objectif de produire des hypothèses syntaxiquement correctes pour un traitement ultérieur. Plus précisément, nous avons étudié l’approche consistant à ajuster dynamiquement les paramètres contrôlant les importances relatives du modèle acoustique et du modèle de langage de façon à produire une grande variabilité lexicale et syntaxique dans le treillis de mots résultant. Nous avons identifié et expérimentalement quantifié deux inconvénients de cette approche : son coût computationnel élevé et l’impossibilité de garantir, dans la pratique, la présence de la solution correcte dans le treillis.

Enfin, nous avons étudié le problème de l’inadéquation de l’utilisation de ressources linguistiques (modèles de langage et lexiques phonétiques) génériques pour mener à une reconnaissance de la parole robuste et efficace. Dans ce contexte, nous avons exploré la solution consistant à utiliser des ressources dynamiques contrôlées par un modèle de dialogue associé. Dans cette approche, un lexique phonétique et un modèle de langage restreints dépendants du contexte du dialogue sont utilisés à la place des modèles com-
plets. Nous avons tout d’abord expérimentalement vérifié que cette solution mène effectivement à l’augmentation des performances de reconnaissance, puis, nous nous sommes intéressés à la production, pour une tâche donnée, d’un modèle de dialogue adapté pouvant intégrer le module de la reconnaissance de la parole. Dans ce but, nous avons proposé et expérimentalement validé une approche permettant l’extension de la méthodologie de prototypage de modèle de dialogue utilisée, par l’intégration d’une simulation de la reconnaissance de la parole dans le cadre de la simulation “Wizard-of-Oz” du dialogue. Nous avons montré qu’une telle approche permet un prototypage plus complet du modèle de dialogue, ce qui, du fait de la prise en compte des caractéristiques du module de reconnaissance de la parole, contribue à garantir meilleure adéquation du modèle de dialogue résultant à l’application vocale visée.
Acknowledgements

I would like to thank all the people I have known during this research and who supported me throughout this work. But first of all, I would like to express my deepest gratitude to Dr. Martin Rajman, my supervisor, who brought me to work on this topic and was always available to discuss any problem. His patience is unlimited! I’m extremely grateful to Dr. Jean-Cédric Chappellier for his open minded and constructive advices.

I would like also to thank the members of my jury, Prof. N. Castell Ariño, Prof. G. Coray, and Prof. I. Kopeček for the important comments they provided me with about my work and for the interesting discussion that I had with them during the defense.

Many thanks to my colleagues Antoine, Romaric, Florian, Christian and others who not only taught me the pleasures of thinking and working, but also the ones of surfing, biking and so on.

And, of course, all my gratitude to my family, and especially my mother, without whom...
Acronyms

All the acronyms used in this dissertation are listed below in alphabetical order.

AM            Acoustic model
ANN           Artificial Neural Network
ASR           Automatic Speech Recognition
CFG           Context-Free Grammar
CMU           The Carnegie Mellon University
CPU           Central Processing Unit
DM            Dialogue Model
DTMF          Dual Tone Multi-Frequency
GDN           Generic Dialogue Node
HMM           Hidden Markov Model
HTK           Hidden Markov model ToolKit
ISDN          Integrated Services Digital Network
LA            Lattice Accuracy
LER           Lattice Error Rate
LM            Language Model
MFCC          Mel-Frequency Cepstral Coefficients
MLP           Multi-Layer Perceptron
NIST          National Institute of Standards and Technology
PIN           Personal Identification Number
PLP           Perceptual Linear Prediction
PoS           Part-of-Speech
RASTA         Relative Spectra Perceptual Linear Prediction
RDPM          Rapid Dialogue Prototyping Methodology
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>Sentence Accuracy</td>
</tr>
<tr>
<td>SAMPA</td>
<td>Speech Assessment Methods Phonetic Alphabet</td>
</tr>
<tr>
<td>SCFG</td>
<td>Stochastic Context-Free Grammar</td>
</tr>
<tr>
<td>SER</td>
<td>Sentence Error Rate</td>
</tr>
<tr>
<td>SFP</td>
<td>Swiss French Polyphone database</td>
</tr>
<tr>
<td>SRE</td>
<td>Speech Recognition Engine</td>
</tr>
<tr>
<td>STRUT</td>
<td>Speech Training and Recognition Unified Tool</td>
</tr>
<tr>
<td>TTS</td>
<td>Text-To-Speech</td>
</tr>
<tr>
<td>WA</td>
<td>Word Accuracy</td>
</tr>
<tr>
<td>WER</td>
<td>Word Error Rate</td>
</tr>
<tr>
<td>WoZ</td>
<td>Wizard-of-Oz experiment</td>
</tr>
<tr>
<td>WSG</td>
<td>Word Spotting Grammar</td>
</tr>
</tbody>
</table>
# Contents

1 Introduction  
2 State-of-the-art  
  2.1 Introduction  
  2.2 Overview of the applicative framework  
  2.3 Speech Recognition  
    2.3.1 Signal processing: features extraction  
    2.3.2 Acoustic models  
    2.3.3 Language Modeling  
    2.3.4 Out-of-Vocabulary words  
    2.3.5 Acoustic decoding  
      2.3.5.1 Viterbi algorithm  
      2.3.5.2 Beam Search  
      2.3.5.3 Stack Decoding  
      2.3.5.4 Multi-Pass Search  
      2.3.5.5 Forward-Backward Search  
  2.4 Speech synthesis: Text-to-Speech  
  2.5 Dialogue modeling  
    2.5.1 Rapid dialogue prototyping  
      2.5.1.1 Producing the task model  
      2.5.1.2 Deriving the initial dialogue model  
      2.5.1.3 The Wizard-of-Oz experiment  
      2.5.1.4 Internal and external field tests  
  2.6 Evaluation procedure and evaluation metrics  
  2.7 Conclusion  
3 Syntactic and pragmatic coupling  
  3.1 Introduction  
  3.2 Syntactic coupling  
    3.2.1 The One-Best paradigm  
    3.2.2 Tight integration with syntactic information  
    3.2.3 Sequential integration with syntactic information: the sequential N-Best paradigm  

19  
23  
24  
24  
25  
26  
27  
28  
29  
29  
30  
30  
31  
31  
31  
32  
35  
35  
35  
36  
36  
38  
39  
40  
42  
43  
44  
44  
45  
45  
46
CONTENTS

3.2.3.1 N-Best search algorithm .................................. 47
3.2.3.2 Word lattice ............................................. 48
3.2.3.3 Parsing the lattice ...................................... 49
3.2.4 Enhanced N-Best paradigm ................................ 50
3.2.4.1 Tuning the parameters of the search algorithm ..... 50
3.2.4.2 Coupling scheme ....................................... 52
3.3 Dialogue integration ........................................... 53
3.3.1 Dynamic lexicon and language model .................. 53
3.3.1.1 Adaptation of linguistic resources ................. 54
3.3.2 Speech recognition simulation ........................... 55
3.3.2.1 Local approach ...................................... 55
3.3.2.2 Confusion matrix ...................................... 57
3.4 Speech recognition weaknesses .............................. 59
3.4.1 Implementing the dialogue manager .................... 59
3.4.2 (Limited) mixed-initiative interaction .................. 59
3.4.3 Avoiding the repetitions .................................. 60
3.4.4 Dealing with the dialogue repairs ...................... 60
3.4.5 Minimization of the duration of dialogues ........... 61
3.4.6 Feedback about the state of the system ............... 64
3.4.7 Dealing with mismatches (conflictual informations) .. 65
3.5 Conclusion ................................................... 65

4 Experimental validation ........................................ 67
4.1 Introduction .................................................. 68
4.2 The used databases .......................................... 69
4.3 The used software ........................................... 70
4.3.1 Speech Recognition Engines ........................... 70
4.3.2 Language modeling ...................................... 71
4.4 Acoustic hybrid models ..................................... 71
4.5 Acoustic Multi-Gaussian models ............................ 72
4.6 Phonetic lexicon ............................................ 73
4.7 Language models ............................................. 73
4.8 Evaluations of speech recognition module ............... 74
4.8.1 Global evaluation framework ............................ 74
4.8.2 Acoustic models .......................................... 74
4.8.3 Phonetic lexicon ......................................... 75
4.8.4 Language models ........................................ 75
4.8.5 Speech Recognition Engines ........................... 77
4.9 Syntactic coupling ............................................ 78
4.9.1 Evaluation framework .................................... 78
4.9.2 One-Best approach: results of the evaluations ...... 82
4.9.3 N-Best: results of the evaluations ...................... 84
4.9.4 Enhanced N-Best: results of the evaluations .......... 88
4.9.5 Coupling speech recognition with a syntactic analyzer 93
## CONTENTS

4.10 Simulation of speech recognition .......................... 94  
4.11 Evaluations of vocal systems: InfoVox system .......... 95  
4.12 Conclusion ..................................................... 96

5 Conclusion .......................................................... 99

A One-Best train recognition: main experiments .......... 103

B One-Best train recognition: additional experiments .... 105

C Synoptic table of the obtained results ................. 107
# List of Tables

3.1 One-word variations of \( N \)-Best paradigm .......................... 50
3.2 Example of enhanced \( N \)-Best results ................................. 52

4.1 Sizes of training and test data for hybrid acoustic models .......... 73
4.2 Training and Cross-validation scores for hybrid acoustic models 73
4.3 Evaluation of the AMs using SFP digits ............................... 74
4.4 Conditions for linguistic resources evaluation .......................... 76
4.5 Evaluation of the SREs on the InfoVox data ............................ 77
4.6 STRUT parameters values used for the evaluations ................. 78
4.7 HTK parameters values for the assumption on the word lattice size .......................................................... 79
4.8 HTK parameters values for the assumption on the parameter space .......................................................... 79
4.9 HTK parameters values for the additional experiments ............ 79
4.10 Test data used during the preliminary experiments ................. 80
4.11 Sizes of train and test data .................................................. 81
4.12 One-Best recognition: preliminary experiments ..................... 82
4.13 One-Best train recognition: main experiments ....................... 83
4.14 One-Best test recognition: main experiments ........................ 83
4.15 One-Best train recognition: additional experiments ............... 84
4.16 One-Best test recognition: additional experiments ................. 84
4.17 \( N \)-best recognition: preliminary experiments .................... 85
4.18 \( N \)-best recognition: additional experiments ....................... 88
4.19 Enhanced \( N \)-best recognition: additional experiments .......... 92
4.20 \( N \)-Best and Enhanced \( N \)-best: additional experiments .......... 92
4.21 Experimental set-up for experiments on sequential coupling ....... 93
4.22 Recognition performances with and without syntactic coupling 93
4.23 Evaluation of the speech recognition simulation ................... 95
# List of Figures

2.1   Typical system architecture of a dialogue-based vocal system  
      (InfoVox system) ............................................ 24
2.2   Overview of the speech recognition process ........................ 26

3.1   Coupling speech recognition with syntactic analyzer: N-Best  
      sequential approach ......................................... 47
3.2   Example of word lattice ....................................... 48
3.3   Example of real word lattice ................................... 49
3.4   Coupling speech recognition with syntactic analyzer: enhanced  
      N-Best approach ............................................... 52
3.5   Scheme of dynamic coupling .................................... 54

4.1   Scheme of training of a language model with the CMU Toolkit 71
4.2   Recognition performance with a phonetic lexicon of varying size 75
4.3   Evaluation with LMs and CFGs .................................. 76
4.4   N-Best approach: recognition performances ........................ 86
4.5   N-Best approach: average computational cost (in CPU-seconds) 86
4.6   N-Best approach: average size of word lattices (in Kilobytes) 87
4.7   N-Best approach: an average number of paths in word lattices  
      (in log-scale) .................................................... 87
4.8   N-Best and Enhanced N-Best approaches: assumption on the  
      word lattice size (word analysis) ............................... 89
4.9   N-Best and Enhanced N-Best approaches: assumption on the  
      word lattice size (lattice analysis) ............................ 89
4.10  N-Best and Enhanced N-Best approaches: average computational  
      cost (in CPU sec) ................................................. 90
4.11  N-Best and Enhanced N-Best approach: average size of word  
      lattices (in Kilobytes) ........................................... 91
4.12  N-Best and Enhanced N-Best approaches: average number of  
      hypotheses in word lattices (in log-scale) .................... 91
4.13  Enhanced N-Best approach: assumption on the parameter space 92
4.14  Experimental set-up for the evaluation of speech recognition 94
Chapter 1

Introduction

Automated speech recognition and its applications in different domains have known significant increase in popularity during the past decades: high performance algorithms and systems are available [50]; and speech recognition performance continues to increase. Substantial progress has been made in the core technologies, leading to the lowering of the barriers to speaker independence, continuous speech, and large vocabularies. The transition from laboratory demonstrations to commercial deployment has already begun [32] and, with steady improvements in speech recognition performance, systems are now being deployed within telephone and cellular networks in many countries. Speech input capabilities are emerging and provide functions like voice dialing (e.g. “Call home”), call routing (e.g. “I would like to make a collect call”), simple data entry (e.g. entering a credit card number), and preparation of structured documents (e.g. “radiology reports”). Several factors have contributed to this rapid progress: substantial progress in acoustic modeling, e.g. through multi-stream approaches; generalization and continuous improvement of the powerful Hidden Markov Model [31]; better language models and more powerful algorithms allowing their integration in state-of-the-art speech recognition systems; production of large speech corpora for system development. Establishment of standards for performance evaluation and advances in computer technology have also indirectly influenced speech recognition.

However, even though much progress has been made, automated speech recognition remains a difficult problem, largely because of the many sources of variability associated with the signal to process, and speech recognition performance still remains relatively poor, especially in the case of the integration of speech recognition in complex speech understanding systems.

The main goal of this research work is to develop and assess new strategies for combining state-of-the-art speech recognition models and advanced linguistic and pragmatic models into efficient speech understanding systems,
in the perspective of improving vocal dialogue-based systems.

Currently, the coupling of continuous speech recognition systems (including very simple language model) and advanced language understanding systems (including stochastic or deterministic finite state automata implementing regular and context-free grammars) is typically made through the \( N \)-best paradigm. Within the general framework of this paradigm, the speech recognizer uses minimal syntactic constraints to first generate a word lattice or a list of transcriptions that are then filtered by more complex linguistic modules.

However, many of the existing implementations of the \( N \)-best paradigm are mainly designed to reduce the overall word error rate and are not necessarily appropriate (i.e. optimal) for a post-processing that includes higher level knowledge sources typically used in language understanding systems. For example, many of the different hypotheses produced by usual speech recognizers often simply correspond to one-word variations of each other. Moreover, the variations often concern short words that typically do not carry much semantic information. This means that no reliable syntactic or semantic models can be developed to enhance the performance of speech recognition systems in case of weak speech recognition.

One of the main problems occurring during the implementation of vocal systems is the management of the interaction with the user. This is often done through dialogue modeling.

A large variety of methods are available for the implementation of dialogue models. These methods range from the simplest ones - graphs and finite state models - to more sophisticated one, such as discourse grammars and plan-based approaches. More recent approaches, e.g. collaborative [45], or agent-based methods [1, 64], attempt to capture the generic properties of the dialogue, and thus make it less task-dependent.

Different problems can be cited concerning the dialogue modeling: (i) natural language inputs in dialogue systems can be extremely ambiguous, and become even more complicated once dialogue is more than a single question followed by an answer from the user; (ii) more complex dialogue models imply more difficulties in their implementations, sometimes leading to tasks extremely costly at the computational level.

In chapter 2, we present an overview of state-of-the-art models and techniques used in the domains of speech recognition and natural language processing (syntactic analysis and dialogue modeling). The general framework of vocal dialogue-based systems is also discussed.

Chapter 3 concentrates on the study of the main limitations of the models presented in the chapter 2, and proposes different solutions. In particular, we
discuss different ways of coupling a speech recognizer with advanced syntactic modules. The goal is to overcome the limitations of statistical language models typically used in the speech recognizers. We also propose several enhancements of the traditional methodology for dialogue prototyping with the perspective of increasing speech recognition performance at the dialogue level.

In chapter 4 are discussed the results of the experimental validation of the solutions proposed in chapter 3. The evaluation of the coupling with syntactic and pragmatic modules aimed at measuring the feasibility of the proposed solutions and their impact on the speech recognition performance. All the evaluation tests were performed with real data, and several experiments were carried out to make different comparative evaluations of state-of-the-art speech recognition models and popular speech recognition engines.

Finally, conclusions and some directions for future work are provided in chapter 5.
Chapter 2

Speech recognition and dialogue modeling

Abstract

This chapter presents the general framework of dialogue-based vocal systems. State-of-the-art models, data structures and algorithms used for automated speech recognition and natural language processing in such systems are also introduced.

The speech recognition module corresponds to one of the key components of any interactive vocal system. In order to accomplish its role, i.e. to produce the orthographic transcription of user vocal utterances, a speech recognition module implements a complex process. This process includes signal processing and features extraction modules, aiming at representing the acquired acoustic signal in some formal way, such as estimated phonemes probability distributions. The acoustic decoding module then produces one or several hypotheses for the required transcription. The main knowledge sources used by a speech recognizer are the acoustic model, the phonetic lexicon and the language model.

Once it is produced, the transcription generated by a speech recognizer is usually further processed by additional modules. For example, in the case of a semantic analyzer, the purpose of the processing is to extract information out of the transcription. In cases where several transcription hypotheses are produced, the use of a syntactic analyzer as a front-end for the semantic module is necessary. Its role is to select one hypothesis (or, at least, to reduce the potentially large number of produced hypotheses) by using some more advanced linguistic model, such as a stochastic context free grammar.

The information extracted from the transcription(s) is then passed to a pragmatic module. A dialogue manager might be used at this stage. Based on the history of the dialogue and the information extracted from the transcriptions, the dialogue manager eventually decides upon the next action to
perform.

2.1 Introduction

One frequent aim for dialogue-based vocal systems is to provide customers with specific information. It can be in the context of a flight reservation, or correspond to a search in a database for information about some person, about restaurants, etc. To respond to this objective, the state-of-the-art architecture of a vocal system corresponds to a combination of a speech recognition module, providing orthographic transcriptions of spoken utterances, a semantic analyzer, aiming at “understanding” the transcriptions, and a pragmatic module, responsible for the management of the dialogue. Interfaces between the system and the required application-dependent databases also needs be considered.

2.2 Overview of the applicative framework

The overview of the typical architecture of a dialogue-based vocal system is given in figure 2.1, illustrating the InfoVox system [66].

Figure 2.1: Typical system architecture of a dialogue-based vocal system (InfoVox system)

The general algorithm of functioning of the system is composed of the following actions:

- acoustic signal acquisition: it can be done using from different sources,
2.3. SPEECH RECOGNITION

- e.g. a cellular or a stationary telephone, a microphone, etc. Recorded data can also be used;

- speech recognition: it is a complex process, including features extraction, phonemes probability estimation, and search (or decoding) algorithm. The most popular output of the speech recognizer is a sequence of words, or a word lattice;

- the semantic module is essentially responsible for the “understanding” of what was recognized. In the case of InfoVox system, the module is implemented as constraints extractor, whose output is a special form filled with the keywords found in the recognized utterance;

- a pragmatic module operates on the results received from the semantic module. In most cases, the pragmatic module is implemented as a dialogue module that makes decisions concerning the strategy of interaction with the human interlocutors. For its decisions, the pragmatic module needs to consult a database with the information of specific nature;

- interactions between the system and human interlocutors are something very important. Speech generation is used for this purpose. It can be implemented as text-to-speech or as prerecorded audio messages.

The detailed discussion of the main components of this scheme is given in the following sections.

2.3 Speech Recognition

State-of-the-art architecture for automatic speech recognition is mostly software architecture which generates a sequence of word hypotheses from an acoustic signal. The most popular algorithms implemented in this architecture is based on statistical methods: for a vector \( y_i \) of acoustic features computed using the acoustic signal (section 2.3.1), the speech speech recognition system generates a word sequence \( W \) with a search process (section 2.3.5) based on the rule:

\[
W = \arg \max_W p(y_i|W)p(W)
\]

where \( W \) corresponds to the candidate having maximum a-posteriori probability, \( p(y_i|W) \) is estimated by acoustic model (see 2.3.2), and \( p(W) \) is estimated by language model (see 2.3.3).

However, although the automatic speech recognition has advanced greatly since its inception, the systems still perform poorly on many tasks, particularly on spontaneous conversational speech. This is mainly due to:
1. acoustic models, which do not capture enough the speech variability. In a 6 weeks workshop at Johns Hopkins University, it was shown that even providing the recognizer with the “right” grammar (i.e., trigrams estimated on the test set), the performance of the systems significantly degraded (from 90% to around 50-60% recognition rates 4.8) when going from read speech to natural speech;

2. language models which do not model the long term dependencies and grammatical constraints of natural speech. This was also shown at the same workshop by providing the recognizer with the correct (cheating) local probabilities (i.e., phone sequence).

We will present in the following sections the state-of-the-art ideas and methodologies used in speech recognition. Its overview is provided in figure 2.2.

![Diagram of speech recognition process]

Figure 2.2: Overview of the speech recognition process

### 2.3.1 Signal processing: features extraction

The representation of the acoustic input, i.e. the nature of the low level signal processing that has to be performed on the acoustic input in order to facilitate the speech recognition task, is an extremely important research question. The role of signal processing is to transform the input acoustic signal to some mathematical description that can be used by further models for processing. During the signal processing, the acoustic signal, coded in some specific format (e.g., A-law or wav format) is treated with some algorithm of features extraction resulting in a set of characteristics describing the signal.

A key assumption in stochastic speech processing is that the speech signal is stationary over short intervals of time. Thus, the signal is divided typically into 10 - 20 msec frames using an overlapping window approach in which each
window accounts for 20 - 30 msec of the signal. Because of the stationarity, the obtained signal frames can then be treated by numerous techniques of speech analysis.

Different computational models of feature extraction are developed. The most popular techniques used for speech processing today are mel-frequency cepstral coefficients (MFCCs) [21] and relative spectra perceptual linear prediction (RASTA) [29, 30]. New models also appear. They are based on multi-stream speech recognition and, as a particular case of this approach, on multi-band processing. Recently, it has been shown, that this approach could be generalized to deal properly with multi-resolution representation to capture acoustic feature dependence at different time scales.

The short-, and medium term feature extraction is used directly in the HMM based recognition module, while compatible long-term feature extraction provides prosodic patterns, that is particularly important at the syntactic level and could generate more relevant N-best hypotheses, discussed in 3.2. Prosodic patterns can be energetic profile, pause structure, and other intonation and durational features. The incorporation of prosodic information considerably reduces the number of parse trees in the syntactic modules and thus decreases the overall search complexity. Although quite a large number of papers have reported the usefulness of prosodic features in speech recognition, the reliability of these features is still not high enough.

The resulted set of characteristics, or feature vectors, is used, for instance, by the acoustic model to estimate the distributions of phone-based likelihoods.

2.3.2 Acoustic models

All modern continuous speech recognition systems are based on the theory of hidden Markov models (HMM) [36]. An HMM is a double stochastic model, in which the generation of the underlying phonemic string and the frame-by-frame surface acoustic realizations are both represented probabilistically as Markov processes in which the parameters can be trained to give optimal performance. Artificial neural networks, such as Multi-Layer Perceptrons, are also used to estimate the frame based scores, yielding to systems referred to as “hybrid HMM/ANN” systems [10].

Hybrid systems are an important trend in speech recognition. The results cited in the literature and obtained in the framework of several international evaluation tests, have shown their capacity to yield competitive performance with additional advantages in terms of CPU and memory requirements.

Hybrid systems, among several other properties, are particularly well suited to:

1. deal with contextual acoustic inputs, without requiring independence assumptions;
2. directly estimate posterior probabilities (as opposed to likelihoods with standard MultiGaussian models, i.e. discriminative vs generative learning), which are better suited (and have been shown to yield better performance) to recognize utterances in terms of phonemes sequences;

3. estimate confidence levels.

A general trend in HMM acoustic modeling now is to include in the models new dependencies on different factors such as phonetic context, speaking rate, “hidden modes”, or longer term dependencies such as a syllable structure and a prosody contour. Sometimes, this however requires significant modifications to the underlying HMM approach. Along these lines, one of the most promising enabling technology is the “multi-stream” approach.

2.3.3 Language Modeling

The role of language models used by speech recognition systems is to filter out all the possible syntactically correct intermediate solutions produced during the acoustic decoding. The necessity of language models resides in the fact that word recognition is not sufficiently reliable to enable multi-word utterances to be recognized with acceptable accuracy without the additional use of a language model [67].

There are two broad classes of language models: the N-gram based on statistical models [5, 38, 59, 12], flexible and adaptable, but which suffers from considerable over-generation (diminishing accuracy), and the stochastic phrase-structure grammar model [44, 36, 37], relatively inflexible and difficult to adapt (hence prone to under-generation), but providing better performance for those sentences that match the grammar (for which a syntactic interpretation is also given).

The standard N-gram based statistical language models suffer from two main shortcomings:

- the questionable validity of the underlying hypothesis of limited scope of probabilistic dependencies;
- the sparse data problem due to the number of parameters that need to be estimated ($V^N$, where $V$ is the size of the vocabulary used).

To respond to the first problem, the solution is to integrate higher level linguistic information in the speech recognition process. New linguistic models are able to simultaneously take the specificities of the output produced by the acoustic modules (word- or phone-lattices) into account. This solution is the subject of discussion in section 3.2.

A possible solution for the sparse data problem is to enhance N-gram estimates with estimates of probabilities of some N-classes such as N-lemmas
2.3. SPEECH RECOGNITION

(sequences of lemmas), or N-PoS (sequences of Parts-of-Speech). Since the number of distinct N-classes (N-lemmas or N-PoS) is smaller than the one of N-grams, better training (i.e. parameter estimation) of the corresponding statistical models can be achieved. However, even if better estimates can be obtained for N-lemmas and N-PoS, when used alone, they usually give higher values of perplexity\cite{40, 22}, that has strong positive impact on the performances of the recognition system.

Several alternatives of class models can be considered:

- multi-layered language models integrating in a unique set of transition and emission probabilities the stochastic information on word and class transitions coming from probability distributions separately trained on the annotated data;

- hybrid language models operating on sequences (simultaneously) made of both words and classes and directly trained on hybrid data.

2.3.4 Out-of-Vocabulary words

For applications such as an automated processing of phonebook queries, the processing of Out-Of-Vocabulary words is of central concern because the size of the lexicon associated with the application is extremely large (essentially due to the presence of a huge amount of proper nouns).

The research track to tackle this problem relies on the use of the already mentioned class models within the acoustic decoder. The general idea is to associate the considered large set of words (such as proper nouns) with a specific class. Transition properties will then be recorded in the language model in a generic way through the class occurrences, while the acoustic properties of the considered word set will be recorded in the phonetic lexicon through the association of the (unique) class entry with a whole phonetic model (instead of a simple phonetic sequence), eventually automatically derived from the available phonetic transcriptions of the annotated class members.

Traditional one step recognition has to be modified. Two steps are needed:

1. a first recognition phase provides word and class occurrence boundaries. For the portions of the signal corresponding to class occurrences, a second recognition phase is carried out to provide (eventually weighted) phone-lattices;

2. the produced phone-lattices are filtered out with finite-state spelling checking techniques using the whole class content as a reference lexicon.

2.3.5 Acoustic decoding

Also often referred to as search algorithms.
In the following sections we provide a generic overview of different search strategies prevalent in speech recognition.

2.3.5.1 Viterbi algorithm

Viterbi search and its variant forms belong to a class of breadth-first search techniques. Here all hypotheses are pursued in parallel and gradually pruned away as the correct hypothesis emerges with the maximum score. In this case, the recognition system can be treated as a recursive transition network composed of the states of HMMs in which any state can be reached from any other state.

Viterbi search is time-synchronous, i.e. at any stage all partial hypotheses generated during the search terminate at the same point in time. Since these hypotheses correspond to the same portion of the utterance, they can be directly compared with each other. However, a complete Viterbi search is impractical for even moderate-sized tasks because of the large size of the state space. A Viterbi beam search is used to reduce the search space.

2.3.5.2 Beam Search

In Viterbi beam search only the hypotheses whose likelihood falls within a fixed radius, or beam, of the most likely hypothesis are considered for further growth [17, 39]. The best beam size can be determined empirically or adaptively. The advantage of the dynamic beam heuristics is that it allows the search to consider many good hypotheses in the absence of a clearly dominant solution. Conversely, in case of a clear best hypothesis few others need to be maintained. The main problem with the Viterbi beam search is that the state-level information cannot be merged readily to reduce the number of required computations. Many variations of Viterbi beam search have been proposed to improve its performance. For instance, different beam widths can be applied at different levels in the search hierarchy [2], and each of these can be adjusted independently based on the number of active paths at that level. In another modification, a tighter pruning beam can be applied to the paths at initial frames of data to limit the extent of hypothesis generation.

In very large vocabulary tasks, a tree-structured network is used to represent the search space in which the states corresponding to phones that are common to different words are shared by different hypotheses. This approach uses the fact that the uncertainty about the identity of the word is much higher at its beginning than at the end. Therefore, more computations are required at the beginning of a word than toward its end.
2.3. **SPEECH RECOGNITION**

2.3.5.3 **Stack Decoding**

The stack decoding algorithm is similar to the A* search popularly used in artificial intelligence. It is a depth-first technique in which the most promising hypothesis is pursued until the end of the speech data is reached.

Stack decoding algorithm requires an evaluation function to compare hypotheses of different lengths. Since the score of a path progressively decreases with time (it is a product of probabilities), the search process is biased to always prefer shorter hypotheses. This problem is overcome by normalizing the score of a path based on the number of frames of data it spans. However, the A* stack decoder suffers from problems of speed, size, accuracy and robustness for large vocabulary spontaneous speech applications.

2.3.5.4 **Multi-Pass Search**

A multi-pass search algorithm employs a coarse-to-fine strategy to decoding. In this approach, computationally inexpensive acoustic models are initially used to produce a list of likely word hypotheses. These hypotheses are later refined using more detailed and computationally demanding models. The first search pass (often called a fast match) produces either an N-best list of possible word sequences or a word graph (or lattice) as its output.

For example, an initial search pass can be performed using word-internal context-dependent phones with a bi-gram language model to generate a list of candidate hypotheses. Then, in a second pass of decoding, a trigram language model, which treats common three-word sequences, can be used with cross-word context dependent phones. The resulting two-pass search will give performance comparable to a single-pass Viterbi search, but often require less computational resources. An important emerging variant of the stack decoding technique is envelope search.

2.3.5.5 **Forward-Backward Search**

Forward-backward search algorithms use an approximate time-synchronous search in the forward direction to facilitate a more complex and expensive search in the backward direction. This generally results in speeding up the search process on the backward pass as the number of hypotheses to be explored is greatly reduced by the forward search.

A simplified acoustic or language model is used to perform a fast and efficient forward-pass search in which the scores of all partial hypotheses that fall above a pruning beam width are stored at every state. Then a normal within-word beam search is performed in the backward direction to generate a list of the N-best hypotheses. The backward search yields a high score on a hypothesis only if there also exists a good forward path leading to a word-ending at that instant of time.
Similar to the Baum-Welch algorithm used for training acoustic models, the total path score at each state of the HMM at time is obtained by combining the scores on the forward and backward passes. The forward pass score of a partial path represents the joint probability of observing the input feature sequence over time instants through, and being in a state at the time. Similarly, the backward pass score denotes the joint probability of a path that accounts for the observed features from time until the final frame of input data, emerging from a state at time. Thus the total path score is given by combining the scores of the forward and backward paths that meet in the same state.

The N-best word sequences obtained using this procedure are rescoring using more sophisticated acoustic and language models to obtain the best sentence hypothesis.

Forward-backward search algorithms have greatly facilitated real-time handling of large-scale speech recognition tasks. The backward pass search is fast enough to be performed without any perceptible delay after the forward search. The forward pass can be made extremely suboptimal and efficient, as the forward path scores do not need to be very accurate relative to the path scores obtained in the backward pass.

2.4 Speech synthesis: Text-to-Speech

Although the primary goal of speech synthesis is to serve within the interfaces of vocal systems, it can also be used for the generation of acoustic data in the framework of speech recognition simulation (section 3.3.2), and particularly in the module concerning estimation of confusion matrices (section 3.3.2.2). For this reason, we also provide here a brief overview of text-to-speech techniques.

Speech synthesis is the process which transforms a string of phonetic and prosodic symbols into a speech signal. The quality of the result is a function of the quality of the string, as well as of the quality of the generation process itself.

Usually, two quality criteria are proposed. The first one is intelligibility, which can be measured by taking into account several kinds of units (phonemes, syllables, words, phrases). The second one, more difficult to define, is often labeled as pleasantness or naturalness. Most of the present text-to-speech (TTS) systems produce an acceptable level of intelligibility, but the naturalness dimension, the ability to control expressibility, speech style and pseudo-speaker identity are still poorly mastered.

**Input to speech synthesis** The input string to the speech generation component is basically a phonemic string resulting from the grapheme to phoneme converter. It is usually enriched with a series of prosodic marks
denoting the accents and pauses. With few exceptions, the phoneme set of a
given language is well defined; thus, the symbols are not ambiguous. How-
ever the transcript may represent either a sequence of abstract linguistic
units (phonemes) or a sequence of acoustic-phonetic units (phones or tran-
sitional segments). In the former case, also called phonological or normative
transcript, it may be necessary to apply some transformations to obtain the
acoustical transcript.

Analogously, the prosodic symbols must be processed differently accord-
ing to their abstraction level. However, the problem is more difficult, because
there is no general agreement in the phonetic community on a set of prosodic
marks that would have a universal value, even within the framework of a
given language.

**Prosody generation** Usually only the accents and pauses, deduced from
the text, are transcribed in the most abstract form of the prosodic string.
But this abstract form has to be transformed into a flow of parameters in
order to control the synthesizer. The parameters to be computed include
the fundamental frequency ($F_0$), the duration of each speech segment, its
intensity. A melodic (or intonation) model and a duration model are needed
to implement the prosodic structure computed by the text processing com-
ponent of the speech synthesizer.

$F_0$ evolution, often considered the main support of prosody, depends
on phonetic, lexical, syntactic and pragmatic factors. Depending on the
language under study, the melodic model is built on different levels, generally
the word level (word accent) and the sentence or phrase level (phrase accent).
The aim of the melodic model is to compute $F_0$ curves. Three major types
of melodic models are currently in use for $F_0$ generation. The first type of
melodic model is production-oriented. It aims at representing the commands
governing $F_0$ generation. This type of model associates melodic commands
with word and phrase accents. The melodic command is either an impulse
or a step signal. The $F_0$ contour is obtained as the response of a smoothing
filter to these word and phrase commands. The second type of melodic
model is based on perception research. Synthetic $F_0$ contours are derived
from stylized natural $F_0$ contours. At the synthesis stage, the $F_0$ curves are
obtained by concatenation of melodic movements: $F_0$ rises, $F_0$ falls, and flat
movements. Automatic procedures for pitch contour stylization have been
developed. In the last type of melodic model, $F_0$ curves are implemented as
a set of target values, linked by interpolation functions.

The phonemic durations result from multifold considerations. They are
in part determined from the mechanical functioning of the synthesizer when
the latter is of articulatory nature, or from the duration of the prerecorded
segments in the case of concatenative synthesis. One part is related to the
accent. Another part, reflecting the linguistic function of the word in the
sentence, is usually related to the syntactic structure. Finally, the last part is related to the situation and pseudo-speaker's characteristics (speech rate, dialect, stress, etc.).

Two or three levels of rules are generally present in durational models. The first level represents co-intrinsic duration variations (i.e., the modification of segment durations that are due to their neighbors). The second level is the phrase level: modification of durations that are due to prosodic phrasing.

One of the most difficult problems in state-of-the-art speech synthesis is prosodic modeling. A large number of problems come from text analysis, but there is also room for improvement in both melodic and durational models. In natural speech the prosodic parameters interact in a way that is still unknown, in order to supply the listener with prosodic information while keeping the feeling of fluency. Understanding the interplay of these parameters is one of the hottest topics for research on speech synthesis today. For prosodic generation, a move from rule-based modeling to statistical modeling is noticeable, as in many areas of speech and language technology.

**Speech signal generation** The last step for speech synthesis is generation of the acoustic signal, according to the segmental and prosodic parameters defined at earlier stages of processing.

Speech signal generators (the synthesizers) can be classified into three categories:

1. articulatory synthesizers;
2. formant synthesizers;
3. concatenative synthesizers.

Articulatory synthesizers are physical models based on the detailed description of the physiology of speech production and on the physics of sound generation in the vocal apparatus. Typical parameters are the position and kinematics of articulators. Then the sound radiated at the mouth is computed according to equations of physics. This type of synthesizer is rather far from applications and marketing because of its cost in terms of computation and the underlying theoretical and practical problems still unsolved.

Formant synthesis is a descriptive, acoustic-phonetic approach to synthesis. Speech generation is performed by modeling the main acoustic features of the speech signal. The basic acoustic model is the source/filter model. The filter, described by a small set of formants, represents articulation in speech. It models speech spectra that are representative of the position and movements of articulators. The source represents phonation. It models the global flow or noise excitation signals. Both source and filter are controlled by a set of phonetic rules (typically several hundred). High-quality, rule-based
formant synthesizers, including multilingual systems, have been marketed for many years.

Concatenative synthesis is based on speech signal processing of natural speech databases. The segmental database is built to reflect the major phonological features of a language. For instance, its set of phonemes is described in terms of diphone units, representing the phoneme-to-phoneme junctures. Non-uniform units are also used (diphones, syllables, words, etc.). The synthesizer concatenates coded speech segments, and performs some signal processing to smooth unit transitions and to match predefined prosodic schemes. Direct pitch synchronous waveform processing is one of the most simple and popular concatenation synthesis algorithms.

2.5 Dialogue modeling

The need for a dialogue component in a system for human-machine interaction arises for several reasons. Often the user does not express his requirement with a single sentence, because that would be impractical; assistance is then expected from the system, so that the interaction may naturally flow in the course of several dialogue turns. Moreover, a dialogue manager should take care of identifying, and recovering from, speech recognition and understanding errors.

The studies on human-machine dialogue have followed two main theoretical guidelines traced by research on human-human dialogue. *Discourse analysis*, developed from studies on speech acts, views dialogue as a rational cooperation and assumes that the speakers’ utterances be well-formed sentences. *Conversational analysis*, on the other hand, studies dialogue as a social interaction in which phenomena such as disfluencies, abrupt shift of focus, etc., have to be considered. Both theories have contributed to the design of human-machine dialogue systems.

In the following sections are presented the methodology for rapid dialogue prototyping (RDPM) introduced in [53, 51].

2.5.1 Rapid dialogue prototyping

The general idea of the methodology is to produce, for any given application, a quickly deployable dialogue-driven interface and to improve this interface through an iterative process based on Wizard-of-Oz experiments, i.e. dialogue simulations.

More precisely, the dialogue model is a finite-state model that can be quite easily and systematically derived from a relational representation of the application itself, hereafter called the *task model*. More precisely, the RPDM consists of five main consecutive steps, namely: (1) *producing a task model* for the targeted application; (2) automatically *deriving an initial dialogue model (and the associated dialogue-driven interface)* from the produced
task model; (3) using the generated interface to carry out Wizard-of-Oz experiments (i.e. dialogue simulations) to improve the initial dialogue model; (4) carrying out an internal field test to further refine the dialogue model (reformulation of system messages, improved feedback, etc.), and to validate the evaluation procedure (coherence, understandability); and (5) carrying out an external field test to evaluate the final dialogue model according to the evaluation procedure defined during the internal field test. All five steps will be presented in more detail in the next sections.

2.5.1.1 Producing the task model

In the RDPM, an application is seen as a set of functions the user can invoke through the dialogue-driven interface to perform the various functionalities provided by the application. In this perspective, an application is modeled as a set of relational tables, where the rows correspond the possible functions (also called "solutions" or "targets") and the columns are the attributes needed to uniquely identify each of the functions and to invoke it.

In other words, the values of the attributes in a row of the solution table (also referred to as canonical values) correspond to the values of the arguments of the function, the call of which results in the fulfillment of the corresponding application functionality. For example, in the InfoVox restaurant application [66], the task model is reduced to a single generic function select_restaurant( Type_of_food, Location, Opening_time, Opening_day, Price_range), the attributes of which identify the five selection features available for the restaurant search. Therefore, the task model of the InfoVox project is simply a table with five columns: Type (i.e. type of food), Location, Time (i.e. opening times), Day (i.e. opening days), and Price (i.e. price range), the rows of which are the various value combinations corresponding to existing restaurants.

2.5.1.2 Deriving the initial dialogue model

A dialogue model is defined as a set of interconnected Generic Dialogue Nodes (hereafter often referred to as GDNs) [8], where each of the dialogue nodes is associated with one of the attributes (also called "slots" or "fields" hereafter) in the solution table. For any given slot, the role of the associated GDN is to perform the simple interaction with the user required to obtain a valid value for the associated attribute. The processing of the GDNs (i.e. the actual interaction with the user according to the specification of the GDNs) is performed by a specific module called the local dialogue manager.

Generic Dialogue Nodes To deal with the various attributes appearing in the solution table defining the task model, we consider three main types
2.5. DIALOGUE MODELING

of GDNs:

1. Simple GDNs (also called static GDNs) associated with static fields, i.e. fields the values of which do not change in time, or change only very slowly; for example the price range for a given restaurant;

2. List processing GDNs (also called dynamic GDNs) associated with dynamic fields, i.e. fields the values of which quickly change in time; for example the types of food in a selected restaurant;

3. Internal GDNs that are used to perform the interactions that are required by various special functions implemented in the dialogue manager (e.g. confirm a selected solution, start/reset the dialogue, etc.).

To realize the interaction for which it is responsible, each GDN contains two main types of components: prompts, that are messages uttered by the GDN during the interaction, and grammars, serving to make the connection between the surface forms appearing in the natural language utterances of the users and the canonical values used in the task model.

Local Dialogue Flow Management Strategy Each GDN is able to locally process five types of generic situations: (1) OK: the user answers the question in an acceptable way; (2) Request for Repetition: the user asks for the repetition of the last system prompt; (3) Request for Help: the user does not know how to answer the question and asks for more explanation; (4) NoInput: the user produces no utterance; and (5) NoMatch: the user answers but nothing useful can be extracted from the produced utterance.

In the case of the OK situation, control is simply handed back to the global dialogue manager which applies the global dialogue management strategy for the activation of the next GDN. In the other four situations, control remains at the GDN level. In these "problematic" cases, there is, therefore, a need for repairing the dialogue and the system operates in the following way: (a) Request for Repetition: the current GDN is reactivated and its main prompt is played if it is the first request for repetition, otherwise the reformulation prompt is played \(^1\); (b) Request for Help: the GDN is reactivated and the associated help prompt is played instead of the main prompt; and (c) NoInput/NoMatch: the current GDN is reactivated and the NoInput/NoMatch prompt is concatenated at the beginning of the main prompt.

Notice that, in all cases, there is an upper limit to the number of consecutive times that a given GDN can be activated. If this limit is exceeded,

\(^1\)During the InfoVox experiments, this strategy appeared to be inadequate in some situations (such as a request for repetition following a request for help) and is currently being modified.
control is handed back to the global dialogue manager with an appropriate error message.

Global Dialogue Flow Management Strategy  The Global Dialogue Flow Management consists of several complementary strategies:

- a branching strategy (also called branching logic hereafter) defining the next GDN to be activated;
- a dialogue dead-end management strategy to deal with dialogue situations where no solution corresponds to the request expressed by the user;
- a confirmation strategy to provide the system with validation possibilities for the values acquired during the interaction;
- a dialogue termination strategy to define when the interaction with the user should be terminated (i.e. a solution proposed); and
- a strategy to deal with incoherences (i.e. situations where there are at least two incompatible values provided by the user).

All these strategies are encoded in the global dialogue manager in an application-independent manner.

2.5.1.3 The Wizard-of-Oz experiment

A Wizard-of-Oz experiment [25] (hereafter called a WoZ experiment) can be defined as a simulation of a human-machine interaction, during which a user is exposed to a system he/she believes to be fully automatic, while a hidden human operator (the wizard) is manually operating (at least) some of the system functionalities that have not yet been fully implemented (sometimes, no implementation at all has been done at the WoZ stage and the experiment then corresponds to a complete simulation) [20, 26].

Within such a setup, the main goal of a WoZ experiment is to provide "realistic" experimental data about the "true" behavior of the members of some targeted user group when faced with an automated system for a given application. To this end, the experimental data are gathered (the experiments are often recorded and/or filmed) and analyzed by the system designers, in order to obtain valuable insights to guide subsequent modeling and implementation decisions [47].

The underlying hypothesis is that it is easier (quicker, cheaper, etc.) to setup and modify a manually operated simulation than to specify, develop, and modify a real implementation of a system. While this hypothesis is undoubtedly very often true, it is, however, important to notice that a WoZ
experiment is by no means a cheap operation. It is not an easy task to make users convincingly believe that they are faced with a machine when the simulation is in fact operated by a human. All clues that could reveal the presence or intervention of the wizard must be thoroughly removed. Thus, actions need to be taken to physically hide the wizard during the interaction and an interaction interface, even simplified, usually needs to be developed to this end. Further, the wizard has to undergo a specific training so that he/she can consistently behave in a manner that can convincingly be believed as the one the user is expecting from a machine (no sophisticated inferences, no emotional reactions, no apparent tiredness, etc.). In addition, it can be quite difficult to guarantee that the behavior of the simulated system will remain uniform over time (the wizard can be in better shape one day than the other, he/she might not consistently remember the provided instructions to operate the simulation over a longer period of time, etc.) [42].

A WoZ experiment can significantly improve the design of an interactive system (e.g. the design of the user interface [11]). Indeed, the results of the experiments are not only used for a first evaluation of the adequacy of the initial decisions (architecture, functions, interface, etc.) of the designers for the targeted system; in addition, the experimental data gathered during the experiment can also serve, if thoroughly recorded, as initial and strongly relevant training data for the system.

2.5.1.4 Internal and external field tests

The evaluation is proposed to be carried out in the form of a satisfaction questionnaire, submitted to a sample of potential users after they had interacted with the prototype, on the basis of a set of predefined scenarios involving specific contexts for a targeted application.

More precisely, for each of the contacted users, the following four step evaluation procedure is proposed:

1. first, the user is provided with a short description of the project and of the evaluation process (with the help of an instruction sheet that has to be systematically read to the user); in particular, the user should be aware of the fact that his/her interaction with the system can be recorded;

2. in the second step, the user is put in a more precise interaction context; to this end, a scenario (i.e. a short story describing a concrete situation where an information about restaurants is required²) has

²The following short story is an example of a scenario for a restaurant application: "For the celebration of the 20th wedding anniversary of your uncle Jim, you would like to
to be selected from a set of different scenarios defined to cover various settings for the attributes describing the targeted application, and read to the user;

3. during the third step, the user is connected directly to the system through a conference call including the supervisor of the field test; in this way, the supervisor is able to monitor the experiment during the system-user interaction, and to intervene if necessary (when for example the user is blocked in some stage of the interaction and is likely to hang up, or when the system crashes);

4. in the fourth step, once the user completes his/her interaction (i.e. either some results are proposed by the system or the user refuses to continue), he/she is submitted the satisfaction questionnaire.

The measured average duration for the above mentioned four steps can vary, and were in the InfoVox project in the range of three minutes for the first step (project and evaluation description), three minutes for the second step (scenario description), five to ten minutes for the third step (interaction with the prototype), and around ten minutes for the last step (satisfaction questionnaire). The overall duration of one evaluation was, therefore, in the range of 20 to 25 minutes, which was considered acceptable by most users.

2.6 Evaluation procedure and evaluation metrics

To assess the performance of a speech recognition system, a suitable evaluation metric has to be defined. This metric is also used in the development and optimization process of the system.

The evaluation is performed on a special set of utterances that have not been used in the training of the acoustic and language models. Such unseen data is used because the system normally performs much better on data that has also been used in the training process thus the estimate of the performance would be overly optimistic.

A simple evaluation metric is the Sentence Error Rate (SER, also called string error rate) which is calculated by comparing the hypothesis string generated by the decoder to the reference string and scoring the whole sentence as wrong if they differ:

\[
SER = \frac{\text{Number of correctly recognized sentences}}{\text{Total number of test sentences}}
\]

make a reservation for 35 people in a restaurant that is opened on Sunday at noon. The event will take place in three weeks from now and should propose quite a fancy menu, as well as a special menu for children
This metric is rather coarse as any deviation of the hypothesis from the reference is counted as one error independent of how similar the two strings actually are.

The most commonly used metrics in speech recognition evaluation are the Word Accuracy (WA) and the Word Error Rate (WER), which measure the positive and negative performances of speech recognizer, or, concretely, measure the number of words that are equal to the reference words or differ between the hypothesis and the reference:

\[
WA = \frac{\text{Correct}}{\text{Reference}}
\]

\[
WER = \frac{\text{Substitution} + \text{Deletion} + \text{Insertion}}{\text{Reference}}
\]

where Correct is the number of words correctly recognized, Substitution is the number of substituted words, Deletion is the number of deleted words, Insertion is the number of inserted words, and Reference is the number of words in the reference.

The calculation of the WER, i.e. the definition of what word is substituted and what word is deleted, is not as trivial as for the SER, because the length of the hypothesis may differ from the reference. Therefore the WER is defined relative to an alignment of the two word sequences.

The optimal alignment is found by a dynamic programming procedure that minimizes the Levenshtein distance (the weighted sum of insertion, deletion and substitution errors) of the two word strings. The word error rate is then given as the number of errors divided by the number of reference words.

Due to an excessive number of insertion errors, the sum \( WA + WER \) can actually exceed 100%.

For most tasks the WER metric is much more appropriate than the SER. For example in a dictation system the subjective quality of the recognition depends on the number of words the user has to correct manually and not the number of sentences in which the errors are located. Still there are applications were the string error rate is the more relevant metric. An example of this is the recognition of digit strings (typically telephone numbers or PINs). Here the misrecognition of even one digit can be catastrophic and render the whole hypothesis string useless. The only way to recover from an error is to repeat the whole string (this is typical in systems where speech is the only input modality, such as voice-dialing applications).

It should be noted that the word error rate is also very approximative, as all word error are treated as equally important. In many domains there are errors which are much more grave than others. For example, in a machine translation system missing one article in a sentence is presumably relatively unimportant whereas misrecognising the main verb will probably be fatal for the whole translation.
2.7 Conclusion

In this chapter we introduced the general framework used for vocal dialogue-based systems. We described the main components of such systems, namely the speech recognition, the text-to-speech generation and the dialogue modeling, and presented several limitations related to the models used for speech recognition, such as, for instance, the problem of training the statistical language models with sparse training data or the limited scope of probabilistic dependencies used by such models.

In the next chapter we present different solutions to deal with these limitations, by integrating more advanced linguistic models. We also discuss the enhancement of the presented dialogue model prototyping methodology relying on the use of a model for the simulation of speech recognition errors.
Chapter 3

Coupling speech recognition with syntactic and dialogue models

Abstract

This chapter presents the integration of high-level linguistic and pragmatic information in the speech recognition process. The goal is to improve speech recognition performance and provide better integration of speech recognition modules in dialogue-based vocal systems. We first present a sequential approach to coupling speech recognizers and linguistic analyzers. This approach overcomes some of the major limitations of statistical language models by using more advanced syntactic models, such as stochastic context-free grammars, and by providing more lexical and syntactic variability in the word lattices submitted to further processing.

We then discuss several issues related to the coupling of the speech recognition module with the dialogue model. We first experimentally verify that the use of information about the current dialogue state can significantly improve speech recognition performance. The discussed coupling scheme consists in a dynamic use of speech recognition resources (lexicon and linguistic models) under the control of the dialogue, i.e. dependent of the current dialogue state. We also propose an enhanced method for training a robust dialogue model.

Finally, we discuss several pragmatic solutions for various problems occurring within dialogue-based vocal systems in the cases where the used speech recognition engines are not fully adequate for the specific targeted application or do not dispose of the data necessary to train reliable acoustic and linguistic models.

All the presented solutions are validated on real data. The obtained results are discussed in chapter 4.
3.1 Introduction

Modern speech recognition is a difficult task, especially when used in dialogue-based vocal applications. This is due to several factors including the necessity to take into account high acoustic variability of multi-locutor tasks and high lexical and syntactic variability of continuous unconstrained oral speech. The previous chapter presented different models designed to deal with these problems.

Considerable progress was made concerning the theory and practical implementations of all of these models, but a number of limitations are still present. For example, in the case of the hybrid acoustic models, we experienced that it is not always possible to train an adequate acoustic model for a specific application, even if the necessary resources are available. As to statistical language models, the short-term nature of the dependencies they use and the difficulties related with training with sparse data still remain one of their important drawbacks. This chapter presents different solutions consisting in enhancing speech recognition by its combination with higher-level linguistic and pragmatic models.

Integration of speech and linguistic models can potentially lead to substantial enhancement of the systems: on one hand, speech recognition can benefit from linguistic information to uncover the correct utterances; on the other hand, the linguistic analyzer can take advantage of all the information provided by the recognizer (multiple hypotheses, global acoustic characteristics, confidence scores, ...) to improve the quality of the analysis.

Several methodologies of coupling speech and linguistic models have already been investigated [54, 55], and the resulting models are often called hybrid models [34, 33]. Two main architectures that have been proposed are the tightly-coupled and sequential architecture [67]. Our work concentrates on the sequential approach, discussed in section 3.2.

An important work has also been done on the theory of rapid dialogue prototyping [51]. For not too complex tasks, it is now possible to produce reliable and functional dialogue models in a realistic way. In section 3.3, we propose an enhancement of the used methodology for rapid dialogue prototyping by integrating it with speech recognition simulation.

Finally, the interesting issue of whether it is possible, at the level of dialogue modeling, to deal with speech recognition errors, is discussed, and several pragmatic solutions are presented in section 3.4.

3.2 Coupling speech recognition with a syntactic analyzer

In this section we discuss the coupling of a speech recognizer with a syntactic analyzer. The use of advanced syntactic information produced by
models such as context-free grammars, leads to more robust and performant speech recognition. Such information is often syntactically more reliable in comparison to the statistical language models used in the search algorithms implemented in speech recognizers.

We first study different ways of coupling a speech recognizer with a syntactic analyzer, namely the tight and sequential coupling schemes. Then we propose an enhanced version of the sequential scheme.

### 3.2.1 The One-Best paradigm

In the framework of state-of-the-art HMM recognition systems, the search algorithm considers all possible hypotheses, evaluates their posterior probability using acoustic and language models, and then chooses one hypothesis with the highest probability. This approach is known as the One-Best approach, and it often uses, for its implementation, frame-synchronous decoding techniques integrating variants of the dynamic programming Viterbi algorithm.

Besides the evident advantage of the easy implementation of this approach in vocal applications, an important drawback of the One-Best approach is that it does not guarantee the correct solution to be found. Evaluations of speech recognition engines using this approach show that even if the performances at the level of words get better, performances at the level of sentences remain poor. This is actually an important issue, because no realistic posterior processing (semantic or pragmatic) is feasible, if solutions obtained with speech recognition are not reliable.

The explanation of this is mainly due to the fact that the used acoustic and language models and parameters of search algorithm suffer from the training problem: due to the unavailability of the necessary training data for most applications, training of models and parameters results in “local minima”, and the combination of such “locally” trained models and parameters leads to poor overall recognition performances.

The reinforcement of language models can be done through tight integration with more advanced syntactic information.

### 3.2.2 Tight integration with syntactic information

Tight integration is made through direct transfer of the syntactic information in the $N$-gram language models of the speech recognizer by approximation of the syntactic model. Various degrees of approximation can be used. The syntactic model can, for example, serve as a basis to derive "theoretical" values for the $N$-gram probabilities used in the acoustic decoder [63, 19]. These "theoretical" values can then be combined with the empirical ones resulting from training. A practical implementation can be done by generation, with
the syntactic model, a huge amount of “theoretical” sentences combined to
the empirical ones and used to estimate N-gram probabilities.

The advantage of tight integration approach is that it requires almost
no modification of the acoustic decoder. On the other hand, the integration
of long distance dependencies remains still very limited. However, it is an
open research question to determine, for concrete applications such as vo-
cal servers, to what extent this limitation effectively influences the actual
performance of the systems.

3.2.3 Sequential integration with syntactic information:
the sequential N-Best paradigm

An alternative to the One-Best approach is the sequential N-Best approach.
In this approach several hypotheses (usually the ones with the highest prob-
abilities) are kept and submitted for parsing to a syntactic analyzer. The
number of intermediate hypotheses kept during the decoding phase is con-
trolled by a parameter typically called beam. With the currently used tech-
niques, the beam parameter is the only practical way of controlling the size
of the produced word lattices: bigger values of beam will lead to bigger word
lattices (and, unfortunately, to higher computational cost and processing
time).

Notice that the value used for the beam parameter corresponds to the
N referred to in the N-Best expression. For example, a 3-best approach
corresponds to a tuning of the speech recognition process where N = 3
intermediate hypotheses are kept during the acoustic decoding. The final
number of hypotheses present in the produced lattice is always much bigger
(in the range of several millions or even much more) and can be computed
only aposteriori, i.e. after the production of the lattice.

The basic assumption of this approach is that, while all available knowl-
edge sources (acoustic, lexical, syntactic, semantic) contribute to improve
overall recognition accuracy, the influence of each of the sources vary greatly
(as measured by perplexity reduction or directly accuracy improvement) at
substantially varying computational cost. For example, a first-order sta-
tistical language model can reduce perplexity by at least a factor 10 with
little extra computation, while applying a complete natural language model
of syntax and semantics to all partial hypotheses typically requires heavy
computation for less perplexity reduction.

It is therefore advantageous to use a strategy in which the knowledge
sources are applied, one or two at a time, in a proper order to constrain the
search progressively: the most powerful knowledge sources are used first to
produce an intermediate scored list of possible hypotheses. The hypothesize
are then filtered and reordered on the basis of the information provided by
the remaining knowledge sources.

Several passes of recognition are required in the N-Best paradigm. Typ-
3.2. SYNTACTIC COUPLING

A first pass reduces the number of possible sentences from the very large number allowed by the language model to a short list of (say 100) alternative hypotheses. Then, these hypotheses are evaluated using the remaining knowledge sources. In sequential approach, the syntactic module is used as a filter operating on the output of the speech recognizer of several forms (word lattice, word graph, phone lattice, ...).

The advantage of the N-Best paradigm is that the techniques developed for language processing can be used almost without modification. The main drawback, true for all sequential implementations, is that the method loosens the advantage of jointly considering speech and language information from various levels which usually greatly reduce the perplexity of the recognition task and therefore the size of the state space that needs to be explored.

Figure 3.1 presents the N-Best sequential approach: for the input acoustic signal, speech recognizer produces a (eventually huge) number of hypotheses; hypotheses form a word "lattice" submitted to a syntactic analyzer using some syntactic model (on the figure, SCFG stays for Stochastic Context-Free Grammar, GTSG stays for Gibbsian Tree Sutitution Grammar [57, 56]); the result is an improved recognized utterance.

![Diagram](image)

Figure 3.1: Coupling speech recognition with syntactic analyzer: N-Best sequential approach

3.2.3.1 N-Best search algorithm

For the implementation of the search algorithm of N-Best approach, an exact breadth-first N-Best sentence algorithm is available [68] and can be used to produce the output word "lattice" with a complexity linear to N. However, as for large values of N, the required computation might become excessive, an exhaustive search cannot be done for any realistic problems. An
approximate version of the algorithm with a computational cost proportional to the number of (locally) kept partial hypotheses is also proposed [28].

### 3.2.3.2 Word lattice

A word lattice is a directed graph. It constitutes an efficient way to represent and store a huge number of word sequences. For example, the lattice containing 10 possible word sequences produced by a speech recognizer for the utterance “He is going there” is given in figure 3.2.

![Word lattice diagram](image)

**Figure 3.2: Example of word lattice**

Two components of word "lattice" should be considered:

- each node represents a point on the time line. Each node is the beginning or the end of a segment containing an acoustic signal corresponding to the pronunciation of a word. In the figure 3.2, six time instants are presented. The duration of the intervals between the nodes is not relevant, and only the fact that there are arcs that go from one time instant to another is important;

- each arc corresponds to the pronunciation of a word. Each arc contains a word spoken over a certain time. This time is the interval between the beginning and ending nodes. In the example, arcs correspond to words “he”, “there”, etc.

An example in figure 3.2 corresponds to a hypothetical case. In reality, word lattice contains hundreds, thousands, or even billions of possible solutions. Figure 3.3 shows real word lattice produced for the utterance “Un paradis terrestre possède un fleuve” (“A terrestrial paradise has a river”) extracted from the Swiss French Polyphone database [16] in the framework of ISIS project [6].

Additional information, such as acoustic and linguistic likelihood of each word, can be kept in a word lattice and used for syntactic processing. An acoustic likelihood of a word is estimated as a score of alignment of the HMM
models associated with the words on the signal segments. The linguistic likelihood corresponds to the N-grams.

3.2.3.3 Parsing the lattice

A syntactic analyzer is used to filter the word lattices produced by the speech recognizer to identify the correct solution(s) that is (are) submitted to semantic or pragmatic processing [15, 4, 58, 62].

An important constraint is required from the syntactic module: for applications such as vocal information servers, only methods with algorithmic complexity polynomial in the size of the inputs can be considered, as exponential methods would require excessive amounts of CPU time to process the incoming data. Such a constraint on the algorithmic complexity of the processing techniques imposes very severe limitations on the sophistication of the linguistic models that can be used, and most often, only finite state models (such as Markov Chains) or context-free models (such as stochastic context-free grammars) are used.

The standard polynomial-time context-free parsing algorithms that are most frequently used are essentially variations of the Earley [24] top-down parsing algorithms or of the CYK bottom-up parsing algorithm [18, 13, 14]. Several improvements have been proposed for both of the parsing approaches, such as stochastic versions or heuristic-based implementations (using a probabilistic version of the A* algorithm) for the Earley algorithm.

It is often a difficult task to select a unique sentence from the word lattice. Additional semantic information has then to be integrated to limit ambiguity. Several extensions, concerning the probabilistic model and/or the linguistic formalism used, can be cited (trigram models for the stochastic modeling of the syntactic rules [41], optimized first-order dependence models [67], Spoken Language Constraint Networks corresponding to extensions of Constraint Dependency Grammars [60, 36], more complex syntactic formalisms, such as unification-based grammars.
A simple way to integrate such formalisms into a speech recognition framework is to represent the unification-based grammar by a context-free approximation, which can be automatically derived with techniques such as the restriction algorithm.

### 3.2.4 Enhanced N-Best paradigm

An important problem with the usual N-Best search strategies is that many of the different hypotheses produced by the speech recognizer are simply made of one-word variations of the same single transcription and often concern short words that typically do not carry much information.

Consider for instance the following example, extracted from the Swiss French Polyphone database. For the spoken utterance “De la cuisine valaisanne s’il vous plait” (“valaisan cuisine please”), the obtained solutions are presented in table 3.1. The variations only concern the words in bold font in table.

<table>
<thead>
<tr>
<th>de la cuisine pas vous a les cent ce non pas</th>
</tr>
</thead>
<tbody>
<tr>
<td>de la cuisine pas vous a les cent ce non non</td>
</tr>
<tr>
<td>de la cuisine pas vous a les cent ce non que</td>
</tr>
</tbody>
</table>

Table 3.1: One-word variations of N-Best paradigm

The proposed enhancement consists in varying the parameters of the speech recognizer to increase the variability of the resulting solutions. In this case, integration can be considered as the problem of merging multiple sources of knowledge according to the formalism of mixture of experts [9] (instead of using Bayes’ rule), which can be expressed as follows:

\[
P(M|X, L) = w_A \cdot P(M|X) + w_L \cdot P(M|L)
\]

(3.1)

where \( M \) represents the hypothesized model (associated with the hypothesized word sequence), \( X \) the acoustic sequence, and \( L \) the knowledge about the language model. \( w_A \) and \( w_L \) respectively represent the relative reliability of the acoustic and language models. This formalization is equivalent to the scaling factor used in standard approaches to scale the acoustic and language model contribution [46]. Notice however that the scaling factor is optimized once for all the training set, while in the enhanced version, the \( w_A \) and \( w_L \) parameters are optimized dynamically during recognition.

#### 3.2.4.1 Tuning the parameters of the search algorithm

Consider the two following cases for the weighting factors \( w_A \) and \( w_L \):
3.2. SYNTACTIC COUPLING

- $w_A = 1, w_L = 0$: situation where all the confidence is given to the acoustic model. In this case, the speech recognizer will have the tendency to produce a word sequence composed of small words, phonetically close to the pronounced utterance;

- $w_A = 0, w_L = 1$: in this situation, the language model has the absolute confidence. For any input acoustic input, the speech recognizer always produces the same word sequence, the one that is the most probable from the linguistic point of view.

When varying the values of parameters, one can expect that “pieces” of correct solutions (partial solutions) will be identified with different values of the parameters. For example, for an acoustic segment of low quality, it would be preferable to constrain the speech recognizer to give more confidence to the language model, especially if this segment is preceded by a segment of good acoustic quality.

Several strategies can be used to achieve effective tuning for the values of the parameters, such as:

- based on the acoustic reliability of the obtained intermediated hypotheses. The parameters can then be estimated from the signal-to-noise ratio;

- based on offline/online supervised/unsupervised maximum likelihood adaptation of the parameters;

- based on the exploration of a large number of possible values for the parameters, by varying the values of the different tuning parameters available for the speech recognition, e.g. scaling factors, word insertion penalty factor.

In the proposed approach, we use the last strategy.

Consider the previously cited example ("De la cuisine valaisanne s’il vous plaît" ("valaisan cuisine please"). Running a speech recognizer with different parameters settings gives the results presented in table 3.2.

In this table, the produced hypotheses are grouped in two blocks corresponding to different settings of the speech recognizer (ASR setting 1 where the acoustic model is favored - $w_A = 0.75$, $w_L = 0.25$, and ASR setting 2 where the language model is dominant - $w_A = 0.25$, $w_L = 0.75$). Within each block, the lines then correspond to the $N$-Best hypotheses. In each of the hypotheses, the correct segments (partial solutions) are indicated in bold.

It can be seen from table 3.2, that the correct complete solution can be extracted from the produced hypotheses as the concatenation of two partial ones: "de la cuisine" produced when the recognizer is constrained by the acoustic model and "valaisanne s’il vous plaît" when constrained by the language model.
### 3.2.4.2 Coupling scheme

The scheme used for the enhanced sequential coupling of the speech recognizer run with different settings and the used syntactic analyzer is presented in figure 3.4. The important difference with respect to the initial N-Best approach is that the hypotheses are produced by varying the values of the parameters of the speech recognizer.

Notice that the production of the recognition hypotheses results from the sequential execution of several recognition passes, the results of which are then merged into a single lattice. For each pass, only the One-Best hypotheses or eventually the N-Best ones with a small value for N (typically, \(beam = 2 - 5\)) are produced.

![Diagram](image)

**Figure 3.4:** Coupling speech recognition with syntactic analyzer: enhanced N-Best approach

In the case of a real implementation, the multiple recognition passes would of course be run in parallel.

The proposed approaches have been experimentally evaluated, and the obtained results are discussed in section 4.9 (4.9.2 for the One-Best coupling,
4.9.3 for the N-Best coupling and 4.9.4 for the Enhanced N-Best approach).

In the next sections we discuss different ways of efficiently integrating speech recognition with the dialogue component of a vocal system.

3.3 Integrating speech recognition within a dialogue model

The use of pragmatic information about the situations occurring in the dialogue can lead to substantial improvements of speech recognition performance. This is done through the dynamic use, during acoustic decoding, of phonetic lexicon and language model associated to some particular dialogue situation.

We propose an enhancement of the methodology for dialogue prototyping by integrating speech recognition simulation. This enhancement leads to the production of more realistic dialogue model.

3.3.1 Dynamic lexicon and language model

In the framework of the methodology for rapid dialogue prototyping introduced in [51] and presented in section 2.5.1, each dialogue situation is characterized by limited amount of phonetic and linguistic informations when compared to the informations describing the whole application.

Therefore, it would be advantageous to use this clue in real application in view to improve speech recognition performances (section 4.8.3). For example, for the situations involving logical questions, when the lexicon is composed only of two words “yes” and “no”, and no language model is needed in this case, evaluations performed during the InfoVox project show $WER = 53.4\%$ for the complete phonetic lexicon and $WER = 8.1$ for the lexicon restricted to two words.

The figure 3.3.1 presents the scheme of dynamic coupling of speech recognition with a dialogue model.

Now, our following consideration concerns the fact that even if the choice of the right dynamic model is a relatively simple task, the use of such models within the systems implementing mixed-initiative dialogue strategies is dangerous: the users can provide more informations than expected by the models. And if we consider speech recognition performance of the state-of-the-art systems as weak (section 4.8), the necessity to deal with the speech recognition errors at the level of the dialogue modeling becomes obvious.

We propose to incorporate the model of erroneous, or “noised” speech recognition in the cycle of dialogue prototyping. This essentially concerns the validation and field-tests. The use of such a model is also practically useful to resolve several limitations concerning the implementation of Wizard-of-Oz experiments:
1. integration of real speech recognition engine. The main difficulty here is the coupling of the speech recognizer with the implemented system;

2. inaccessibility of data required to prepare models for the speech recognition engine, the aim of WoZ experiment being the acquisition of the data;

3. impossibility to control the performance of the speech recognition. Every speech recognizer tries to perform the best speech recognition it can. It is therefore impossible to “play” with different performance levels to evaluate the performances of the dialogue models.

3.3.1.1 Adaptation of linguistic resources

In the case of dynamic models, adaptation of the language model is necessary. It can be made by renormalization of values of the subset of N-grams extracted from the original model.

The problem we have here is that the probabilities (and therefore their average) are a priori dependent on the whole phonetic lexicon. In order to be able to reestimate a subset of N-grams, we need to make the following assumption: any modification of phonetic lexicon does not change the order of relations between the probabilities.

For we have a dynamic phonetic lexicon V extracted from the complete lexicon W, i.e. \( V \subset W \), the probabilities and their average of \( P_V() \) can
be derived on the basis of $P_W()$, due to the fact that any modification of phonetic lexicon (e.g. partitioning it in subsets) does not change the order of relations between the probabilities\footnote{Indeed, independently of the volume of used phonetic lexicon, speech recognizer produces the same acoustic scores for the same phonetic words}. Therefore

$$P_V(w_j|w_i) = \frac{P_W(w_j|w_i)}{\sum_{w_j \in V} P_W(w_j|w_i)}$$

### 3.3.2 Speech recognition simulation

We consider speech recognition as a stochastic process characterized by parameters such as acoustic and language models, and meta-parameters such as relative importance of acoustic and language models or the word insertion penalty.

Each parameter suffers from the training problem which is linked to the fact that the trained parameters achieve local minima: due to the unavailability of the necessary training data for most applications, the combination of these “locally” trained parameters results in overall recognition performances worth than those of human interlocutors, especially in the case of unconstrained continuous speech.

The evaluation of speech recognition is based on Word Accuracy (WA) and Word Error Rate (WER) scores [27, 48]. These scores measure the recognition performances of a given speech recognition engine (SRE) for a given application: WA tells the proportion of well recognized words, WER indicates the proportion of misrecognized words.

One possible idea of speech recognition simulation would be thus to directly operate on the WA and WER scores, as done in Boris project [23], but putting more theoretical sense in the choice of the words to be corrected. This idea is used as basis of the proposed approach, named Local approach.

The following section describes in details this approach.

#### 3.3.2.1 Local approach

In this approach, we model the speech recognition as a process performing four actions on each word: leave the word unchanged, replace the word, delete the word, insert new word. A probability is associated with each of the actions, estimated on the basis of our expectation of the overall WA and WER. The problem here is therefore to estimate the probabilities.

Let first define the following variables:

$$WA = \frac{1}{n} OK$$

and

$$WER = \frac{1}{n} (DEL + INS + SUBST)$$
where \( OK \) is the number of well recognized words, \( DEL \) is the number of deleted words, \( INS \) is the number of inserted words, \( SUBST \) is the number of substituted words, and \( n \) is the size of the input sequence:
\[
n = OK + SUBST + DEL
\]

The size of the output sequence is
\[
n' = OK + SUBST + INS
\]

Then, after the necessary transformations, we obtain the estimations of the probabilities:

\[
P_{ok} = \frac{WA}{WER + WA} \tag{3.2}
\]

\[
P_{ins} = 1 - \frac{1}{WER + WA} \tag{3.3}
\]

\[
P_{del} = 1 - \frac{\sigma}{WER + WA} \tag{3.4}
\]

\[
P_{subst} = 1 - P_{ok} - P_{ins} - P_{del} \tag{3.5}
\]

where \( WA \) and \( WER \) are average values balanced by the lengths \( n_i \) of the observed word sequences, \( \sigma \) is the average balanced expansion rate:
\[
WA = \frac{\sum n_i W_{Ai}}{\sum n_i}, \quad WER = \frac{\sum n_i W_{ERi}}{\sum n_i}, \quad \sigma = \frac{\sum n_i \sigma_i}{\sum n_i}
\]

and \( \sigma_i \) is the expansion rate:
\[
\sigma_i = \frac{n'_i}{n_i}
\]

Finally, the algorithm is the following:

1. initialize the values of \( P_{ok}, P_{ins}, P_{del} \) and \( P_{subst} \) on the basis of 3.2-3.5 (we suppose that \( WA, WER \) and \( \sigma \) are initial parameters);
2. put the first word in the sequence as the current word;
3. randomly choose one operation \( OK, INS, SUBST \) or \( DEL \) according to the probabilities;
4. if the selected operation is:

- **OK**: copy the current word to the output sequence; finish if the current word is the last word in the input sequence; put the following word as the current word;

- **SUBST**: substitute the current word by some word and copy it in the output sequence; the substitution is done either uniformly or on the basis of words confusion matrix (see 3.3.2.2); finish if the current word is the last word in the input sequence; put the following word as the current word;

- **DEL**: do nothing with the output sequence; finish if the current word is the last word in the input sequence; put the following word as the current word;

- **INS**: choose randomly some word and copy it in the output sequence;

5. return to 3

The proposed algorithm was validated, and the results are discussed in section 4.10.

### 3.3.2.2 Confusion matrix

In the words (resp. phonemes) confusion matrix, the cell \([i][j]\) corresponds to the probability of confusing the word (resp. phoneme) \(i\) with the word (resp. phoneme) \(j\), or, more precisely, it corresponds to the conditional probability \(P(i|j)\) that the word (phoneme) \(i\) is recognized when the word (phoneme) \(j\) has been pronounced. This probability measures the phonetic similarity between words (resp. phonemes).

We propose two approaches for estimation of the confusion matrix values.

**Empirical estimation** In this approach, the confusion between words is observed on audio recordings of the isolated words, and the probability \(P(i|j)\) is the average probability obtained during the recognition. The necessary audio recordings can correspond to some real data acquired during experiments, or to data synthesized with text-to-speech techniques.

It is more difficult to produce a reliable phonemes confusion matrix with the empirical approach, since in the case of real data, it requires the phonetic segmentation of the audio recordings, and in the case of text-to-speech, the synthesized data might not correspond to the desired coverage of the targeted application.
**Estimation by generation** This approach is used in the cases where no acoustic observations are available for the phonetic lexicon. The approach relies on a phonetic lexicon modeled with HMM models [49]: each word of the lexicon is a sequence of phonemes, each phoneme is a Multi-Gaussian left-to-right model.

Comparing to the empirical approach, the values of confusion matrices obtained with this approach are "theoretical", but the advantage of the approach by generation is that it can be used for any phonetic lexicon.

The functioning of the approach is the following:

- a set of observation sequences is generated for each phoneme;
- using the generated data, phonemes confusion matrix is produced using the empirical approach;
- words confusion matrix is produced using phonemes confusion matrix.

The data generation algorithm is taken from [35], and it is as follows:

1. for each model, set the state index $j = 1$ and the time index $t = 1$;

2. partition the unit interval proportionally to the mixture coefficient $c_{jm}$, with $1 \leq m \leq M$, where $M$ in our case is 24. Generate $x$, a random number uniform on $[0, 1]$. Select the mixture density, $l$, according to the subinterval in which $x$ falls;

3. decompose $[U_{jt}]$ into $Q \Lambda Q'$, where $Q$ is the matrix of eigenvectors of $U_{jt}$ and $\Lambda$ is the diagonal matrix of eigenvalues of $U_{jt}$;

4. generate a $K$-dimensional normal deviate, $y$, of zero mean and covariance $\Lambda$;

5. set the output observation $O_t = Q_y + \mu_{jt}$;

6. partition the unit interval proportionally to $a_{jk}$, $1 \leq k \leq N$. Generate $x$, a random deviate uniform on $[0, 1]$ and select the next state, $i$, according to the subinterval in which $x$ falls;

7. increment $t$;

8. if $t \leq T$ go to 2, else, stop.

In the next section we discuss several pragmatic solutions to deal with speech recognition weaknesses.
3.4 Dialogue model for speech recognition weaknesses

In this section we concentrate on several issues briefly mentioned in previous sections because we believe that coupling speech recognition with the dialogue model can be favorable in the sense that speech recognition weaknesses can be treated at the level of dialogue modeling. This reflects the idea presented in the previous section: speech recognition and dialogue modeling should not be separated, and their cooperation can lead to better human-machine interactions increasing the overall user satisfaction.

More exactly, this section indicates means allowing, at the level of dialogue modeling, to deal with several speech recognition limitations, while offering to the user additional possibility to interact with the system.

3.4.1 Implementing the dialogue manager

We propose a set of principles used to guide the development of dialogue model:

1. (limited) mixed-initiative interaction;
2. avoid repetitions in the dialogue;
3. deal with the dialogue repairs during the dialogue. Dialogue repairs are used as reference to speech repairs, i.e. dialogue acts related to the dialogue itself, and not to the subject of the dialogue;
4. minimize the duration of the dialogues;
5. feedback about the state of the system;
6. deal with conflictual informations, i.e. misconception, misunderstanding and non-understanding.

Next sections present how these principles can be implemented during the production of the dialogue model.

3.4.2 (Limited) mixed-initiative interaction

It consists in the possibility for the user on one hand to break the dialogue flow imposed by the system by asking for repetition or explanation of the last question (cases assimilated to dialogue repairs), and on the other hand to anticipate the future questions by providing the elements of response in advance. This is typically the case with the half-opened question serving to initiate the dialogue.
Note however that the user can indeed choose not to answer the asked question: on one hand, it is possible that the additional informations provided by anticipation result in realization of the current task, and thus progressing in the dialogue; on the other hand, as the attribution of a field to a dialogue node is realized dynamically, during every visit of the node, a loop on this node will not necessarily lead the system to ask the same question. In consequence, as in the case of a dialogue between humans, an unexperienced user will be guided by the system, while an experienced user will be able to provide the system with the pertinent elements at once.

### 3.4.3 Avoiding the repetitions

In order to avoid too “mechanical” aspect of the system during the dialogues, it appears important to avoid the repetition of the same system messages.

For this purpose, different alternative formulations, more or less equivalent, can be defined for each system message; in addition, some of them should be contextualized (for example, welcome and good-bye message, according to the hour and the day of the dialogue: “good evening”, “have a nice day/week-end”).

In the cases where one message has to be played several times during the dialogue, a formulation that differs from the most recent occurrences is chosen.

Beside more natural aspect, this approach can be implemented as mechanism of disambiguation, when the user indicates that he doesn’t understand the message.

### 3.4.4 Dealing with the dialogue repairs

For an answer of the user, the following situations should be considered, on the basis of values and contexts found in the answer, as well as in the context imposed by the system.

**Ok + Initiative** The user answers the question, or says something.

that can be interpreted (even partially) by the system as an answer to another question, and can be taken into account. This case is not a *dialogue repair*, and the dialogue can continue.

**Repetition** The user asks explicitly for the repetition of the last system utterance.

The last system message will be repeated. When the repetition is asked for several times (2, 3), it is preferable that the system gives an alternative formulation.

**User misunderstanding** The user says (more or less clearly) that he did not understand the question.
An alternative formulation of the last system message can be used at the first time, and in the case of new misunderstanding, the system continues with the request for assistance. When the concerned message is an opened question, and the misunderstanding is repeated, the system continues with the guided dialogue, in which the system asks the user to answer only one posed question at a time.

**Request for assistance** The user does not know how to answer the question, but keeps in mind that he is faced with a machine.

The system indicates to the user how to answer, by providing him with valid example of answer, or by closing (at least partially) the question.

**Timeout** The user stays mute after few second (recording stops on timeout).

The system asks the user to talk louder and to wait for the signal before answering, then continues with the repetition.

**System non-understanding + Out of context** Nothing can be extracted from the answer (problem with the data processing sequence (recognition), overcomplex answer, rumble and other non verbal, etc.), or the user answers something that has no relation with the task.

For dialogue models that do not take into account this kind of scenario, this case should be assimilated to a case *non-understanding*. Otherwise, the treatment requires the modeling of the “contexts” outside the application (word knowledge); this could be also foreseen for a well defined set of contexts, defined on the basis of *a posteriori* examination after sufficiently long period of functioning of the system, and for which this treatment would be justified.

The system indicate to the user that he was not understood, and require a repetition. If this situation occurs again, the request for assistance is triggered; as for the user *non-understanding*, the system can then switch to driven dialogue, or use some exit mechanism (based on watchdog) to go out of the loop.

### 3.4.5 Minimization of the duration of dialogues

The minimization of the duration of the dialogue (in terms of dialogue turns and total time), without any restriction on the task, is in fact more an objective than just a way to improve the dialogue\(^2\).

Many factors can be cited that influence the duration of the dialogue. However, following considerations are always useful:

\(^2\)For the same result, shorter dialogue is more “efficient”; one limits the risk to annoy the user, as well as possible miscommunications
• For long messages that can be played several times during the dialogue, it can be useful to dispose, in addition to alternative formulations, shorter (elliptic) reformulations. Multiple long formulations can be used to clarify the main message and its reformulations in the case where the same message has to be played later during the dialogue.

• Mixed initiative is also an efficient way, because it permits to indicate several informations at a time.

• Generally, loops have to be avoided in the dialogue. This is not necessarily easy to realize, because some loops are mandatory, for example, when the user asks explicitly several times the system to repeat.

One possible solution would be to implement the control at several levels:

– in the case of consecutive non-understandings (user or system) of an opened question, a driven dialogue is instantiated;

– for the same situation, but in a driven dialogue, for a frame containing several empty fields, the system gives up with the current field and switches to the next empty one;

– in the cases of loops on non-understandings, but with the questions “yes/no”, the system selects a (default) value inducing as less consequences as possible, and plays a message to the user explaining such a decision;

– global watchdog, instantiated in the case of non-progressing in the dialogue (for example, when no field is informed or modified during last $n$ interactions, strict repetitions excluded); in this case, depending on the application, different actions can be considered: either one proposes to the user to present already available informations, or user is proposed to abort the dialogue, possibly contacting the human operator, etc.

The dialogue can become untreatable in the case where non-understandings arrive at this moment. A good prevention is then to use a minimalist dialogue, by using DTMF keys.

In addition to these considerations, one can also profit of the nature of the task, either by choosing “intelligently” the questions to ask (prior fields to be informed first), or, in case of troubles, terminating the task by providing some partial information (or service):

**search task** information retrieval of one or several elements *a priori* presented in the database.
3.4. SPEECH RECOGNITION WEAKNESSES

For this kind of task, one can choose, for a driven dialogue, to inform in priority the fields with the highest potential to discriminate; this will minimize the average duration of each dialogue.

An additional advantage of this technique is that it offers the possibility to weight the elements of the base depending of their “popularity”, permitting to minimize even more the average duration of dialogues.

However, one should to control that the scenarios too surprising and able to destabilize the user are not chosen, for example, ask the user about the color of an article, before asking for the nature of this article.

In addition, mixed initiative constraints to realize this choice dynamically; if the database is huge, the number of attributes is big and the attributes are highly various, this choice can become relatively expensive operation.

**advice task** search for an advice or propositions; it is still information retrieval, but of the element that is not necessarily in the database.

In this strategy the system proposes a set of restaurants on the basis of preferences indicated by the user. In this case, the discriminating potential for a question has no sense any more. On the other hand, it appears possible to propose to the user a set of elements that do not satisfy the whole set of attributes, but only a sub-set of them.

This technique is relatively easy to implement: the user indicates her/his choices during the dialogue. When no database element satisfies the indicated choices, rather than to terminate or restart the dialogue or propose to the user to modify some choices, the system can ask if the user agrees to accept a sub-set of her/his preferences. It is possible then to relax the last specified constraints, considering that first indicated elements are most important, possibly in successive way (in this case be careful not to submit several times the same propositions).

Before proposing to the user the “approximate solutions”, one has to be sure that the constraints relaxation provides the solutions that conserve a sens in regard to the user request; if this technique is applied in the cases where the user indicated just few constraints, the relaxation of one can lead to huge number of uninteresting elements. In the framework of implemented prototype, one controls, before proposing the solutions, that the number of targets obtained after each relaxation remains reasonable compared to the number of excluded targets. In the cases where this ratio becomes too big, the mechanisms cited above (ask the user to modify the request, submit it or terminate) are applied.
CHAPTER 3. SYNTACTIC AND PRAGMATIC COUPLING

3.4.6 Feedback about the state of the system

In the case of dialogue between humans, the progression in the dialogue is
accompanied by acknowledges.

These acknowledges are not however easily implementable for different
pragmatic reasons, such as no barge-in, too hazardous recognition of ac-
knowledges, reaction time of the system, etc.

In order to avoid that system misunderstandings (numerous because of
speech recognition errors) don’t lead the system to provide the user with
solutions without any correlation with her/his needs, it is necessary to detect
and to handle these situations; the only possibility to do it is to inform the
user about the retained elements.

The first implementation of such feedback is, before terminating a sub-
task (for example, at the exit of a frame), to produce a synthesis of different
retained elements (the fields values recently informed after the last synthesis).

However, this technique alone was considered not sufficient (the correc-
tion of erroneous values can lead to tiresome dialogues, the users are not
always attentive enough to realize that the system misunderstood what they
said, especially when the number of fields is important, ...).

It can be decided to implement an approximation of acknowledge during
the dialogue, by including, as prefix of each closed question, an indication
of one of the fields. Several possibilities to choose a field can be considered:
the first informed, the last informed, depending on the order of appearing of
the values, etc.

One has in addition to consider the context of the question while choosing
the fields for confirmation, avoiding the couples with shared modalities.

Here is an example of system message composed of confirmation prefix
and question about the next field:

“For your meal tomorrow night, what localization do you prefer?”

In the case of agreement (or no explicit protest) from the user side, the
system marks the field as being “confirmed”; it will not be possible from this
moment to modify it by mixed-initiative mechanism. If the user protests,
either by negative response, or by indicating a value compatible with (but
different from) the confirmed one, the system asks the user for her/his pref-
erence about the field, by prefixing the possibly conflict values.

This solution can not guarantee the confirmation of all informations that
will be used by the system, especially if the user anticipates the questions of
the system, but it still remains balanced between a limited negative impact
on the dialogue quality and the most prompt correction of recognition errors.
3.4.7 Dealing with mismatches (conflictual informations)

When the user indicates several incompatible values for the same field, a sub-dialogue aiming to define the desired value is instantiated (the system asks the user to choose one value for that field, and indicates in addition the conflict elements).

To avoid that the recognition errors do not lead to these conflict situations, the informations identified by the system are strongly filtered, before the fields are update:

- If the response of the user is given in the context imposed by the last system message or contained in the confirmation feedback prefix, only the elements compatible with one or other of the contexts are conserved. In the case where no such elements exist, the filtering is canceled and all the elements of the response are taken into account.

Such filtering is relatively rough and strongly limits the initiative of the users. Better solution would be to filter only the elements for which the confidence score is too weak. This would also permit to short-cut the filtering for the Web interactions.

- If the analysis of the user response indicates the presence of at least one non conflict value, the conflict elements are filtered, except for those the context of which is compatible with the possible confirmation feedback.

All the proposed solutions were used during the production of a dialogue model implemented during the InfoVox project [66]. The evaluation of these solutions were made during several field-tests. The results are discussed in section 4.11.

3.5 Conclusion

This chapter described several strategies for the integration of speech recognition in dialogue-based vocal systems by coupling it with higher-level syntactic and pragmatic informations.

We studied tight and sequential approaches for the coupling in the perspective of improving speech recognition performances. We proposed an enhancement of the sequential approach that leads to increased lexical and syntactic variability necessary for successful syntactic parsing.

We then proposed several enhancements of the used methodology for dialogue prototyping. The goal here was to deal with speech recognition errors at the level of dialogue management. These enhancements are important
because speech recognition errors seem to be unavoidable for any real life dialogue system based on vocal interaction.

The first enhancement concerns the integration of recognition error modeling in the dialogue prototyping methodology. One important feature of this approach deserves to be highlighted: the simulation of substitution errors that, in our model, is made on the basis of phonetic similarities between words. Indeed, the correct estimation of the phonetic confusion matrix can guarantee that the substitutions performed during the simulation processes are very close to the ones made during real speech recognition.

Other enhancements concern the dialogue model itself: several pragmatic solutions have been proposed to deal with speech recognition errors.
Chapter 4

Experimental validation

Abstract

This chapter presents the results of the validation of the solutions proposed in the previous chapter. Three types of experiments are discussed:

Speech recognition evaluation We evaluate the impact of different components of a speech recognition system on its performance. Components that might be considered are the acoustic model, the phonetic lexicon and the language model, but our efforts mainly concern comparison between state-of-the-art hybrid and MultiGaussian acoustic models.

Syntactic coupling validation In this part we discuss the sequential coupling of speech recognition with a syntactic analyzer. The aim of the work is the validation of the feasibility of the sequential approach and its ability to yield, at a reasonable computational cost, better speech recognition performance when compared to the state-of-the-art One-Best approach. The proposed sequential approach was fully implemented. At the experimental level, preliminary experiments were set up to evaluate its performance. However, most of our experimental efforts focussed on the validation of the quality (as measured by lattice accuracy scores) of the proposed Enhanced N-Best approach, with respect to the lexical and syntactic variability of the produced word lattices.

Dialogue component validation We present the results of the validation of the speech recognition error simulation model. We also introduce several enhancements for the used dialogue prototyping methodology and evaluate their impact on the global quality of a vocal system as perceived by the users. This evaluation is performed in the framework of the concrete example of the InfoVox system.
4.1 Introduction

Evaluation is an extremely important research issue, both in general and in the specific case of dialogue-based vocal systems. Several approaches have been proposed for the evaluation of such systems, e.g. [43]. In our case, we concentrate on task-dependent techniques operating on real data.

As already presented in previous chapters, we focus on the three main dimensions of any dialogue-based vocal system: acoustics, syntax and pragmatics.

At the acoustic level, our goal is to measure the impact of the nature of the used acoustic model on speech recognition performance and to compare the influence of different state-of-the-art methodologies for the production of acoustic models. For this purpose, we propose an evaluation framework integrating several speech recognition systems that we used in different research projects. We also take advantage of the experimental setup to perform several additional tests: we evaluate the impact on recognition performance of the use of phonetic lexica of different sizes and varying linguistic resources. Finally we perform a comparative evaluation of different commercial speech recognition engines with recognition systems widely used in the academic domain.

At the syntactic level, our efforts mainly concentrate on the evaluation of the two approaches presented for the sequential coupling: the N-Best and the Enhanced N-Best sequential approaches. During the evaluation, we test two hypotheses. The first one (“assumption on the word lattice size”) concerns the assumption that lattice accuracy requires word lattices containing a huge number of hypotheses. The second one (“assumption on the parameter space”) concerns the assumption that fine-grained exploration of the parameter space is required to achieve better lattice accuracy. Preliminary experiments on the coupling with a real syntactic analyzer have also been performed.

Finally, at the pragmatic level, we consider two types of issues, both aiming at the increase of user satisfaction at the level of dialogue modeling: the simulation of speech recognition errors and the overall evaluation of our enhanced prototyping methodology for dialogue-based vocal systems. All our experiments are performed in the framework of the InfoVox and Inspire projects. For the first proposed issue, we validate our approach by comparing the utterances processed through simulation to the true recognition outputs produced by the Loquendo speech recognition system. For the second issue, all the proposed solutions are implemented in the dialogue model of the InfoVox project and validated during two field-tests. Although the individual impact of each of the enhancements cannot be measured due to the cost of the field-tests that would be necessary for that, the obtained results show that the use of the proposed solutions indeed has a positive impact on user satisfaction.
4.2 The used databases

In our experiments we essentially deal with French language. However, several additional experiments were performed with German.

**The Swiss French Polyphone (SFP)** is provided by the IDIAP\(^1\) institution. The database contains audio recordings from 4'500 speakers (2'500 females and 2'000 males) acquired over the ISDN network. The recording work took place in 1997 and was carried out by Swiss Telecom PTT\(^2\) and IDIAP.

Utterances acquired for the database were designed in a way to provide the maximum phonetic and syntactic coverage of the Swiss French language. There are sequences of single and connected digits including the hash and the star symbols, telephone and credit card numbers, date and time phrases, spelled words, as well as read and spontaneous sentences.

The used version of the SFP contains the audio files stored in 8 bit A-law speech format, the orthographic transcriptions of the audio files and the used phonetic lexicon in Sampa format.

**InfoVox** Additional telephone ISDN audio recordings were acquired during the InfoVox project [66]. The spoken language was French. The audio files are stored in 8 bit A-law format. The recordings were made during the following 3 sessions:

- Wizard-of-Oz experiments: 99 speakers were contacted and produced 256 dialogues containing about 2'185 utterances;
- The internal Field Test: 25 persons were recorded to produce 25 dialogues containing 287 utterances. All the contacted persons were either involved in the project or were aware about the functioning of the system;
- The external Field Test: 50 "external" users (i.e. users that did not have any prior knowledge about the system) were randomly selected in all French speaking cantons in Switzerland. 50 dialogues yielding 819 audio recordings were acquired.

**German SpeechDat(II) (SDGe)** was also provided by IDIAP. The structure and the content of the database are similar to the ones of SFP: about 4'000 speakers, audio recordings (8 bit A-law format) including isolated words, read and spontaneous speech, orthographic transcriptions and a phonetic lexicon. The SDGe was produced in 1999 at the University of Munich\(^3\).

---

\(^1\)http://www.idiap.ch
\(^2\)http://www.swisscom.com
\(^3\)http://www.phonetik.uni-muenchen.de
Inspire  Dialogue data was recorded during the Inspire project \(^4\), 10 dialogues in German were read by 10 speakers, 5 males and 5 females, giving 1'370 audio recordings. The recordings were made in good acoustic conditions over a microphone.

4.3 The used software

4.3.1 Speech Recognition Engines

We used two families of speech recognition engines: commercial and research systems. Commercial systems are considered as “black-boxes” delivered with all the necessary data and with tuned parameters; research systems are in fact source codes that we used to train the functional systems.

The used speech recognition engines were:

HTK  an academic system\(^5\) based on HMM technology; supports the whole (“front-end”) speech recognition process; includes numerous algorithms for features extraction and the possibility to train acoustic and language models. Version 3.1 of the software was used;

Strut  an academic system based on the hybrid technology; supports the whole speech recognition process; includes numerous algorithms for features extraction and the possibility to train acoustic models. The version 2.9p1 was used;

Sirrocco  an academic acoustic decoder\(^6\) integrating estimation of phonemes probability distribution on the basis of feature vectors extracted by another system, HTK for example. The version 1.2.1 of the software was used in combination with HTK;

Nuance  an HMM based commercial system\(^7\) provides different acoustic models, supports the whole recognition process. The version 7.0 was used;

Loquendo  an HMM/ANN-based commercial system\(^8\) includes different acoustic models, supports the whole recognition process. Implements the functions to train language models;

\(^4\)http://www.inspire-project.org  
\(^5\)http://htk.eng.cam.ac.uk  
\(^6\)http://www.irisa.fr/sirocco  
\(^7\)http://www.nuance.com  
\(^8\)http://www.loquendo.com
4.3.2 Language modeling

**CMU Toolkit (The Carnegie Mellon University Toolkit)** a set of utilities\(^9\) to create N-gram language models from large text corpora. The scheme for training of a language model with the CMU is illustrated in figure 4.1.

![Diagram of language modeling process](image)

Figure 4.1: Scheme of training of a language model with the CMU Toolkit

4.4 Acoustic hybrid models

The Artificial Neural Network used in our experiments is a Multi-Layer Perceptron (MLP). MLPs have been shown to be able to approximate any function, and, in particular, they can learn a posterior probabilities.

In theory, the output of the MLP (and input to the acoustic decoder) can be any phone like units. However, in our experiments we have used the phoneme set defined by the SAMPA standard which is used in BDLex.

The preprocessing of the speech signal for all the models consists of a RASTA-PLP feature calculation. 12 RASTA-PLP coefficients along with their first derivatives (delta's), as well as delta and delta-delta values of the log-energy, making a total of 26 features, are calculated every 10 ms (a frame) using a window of 30 ms (thus an overlap of 20 ms.) As it is usually done with MLPs in hybrid systems, we use a context of 9 consecutive feature vectors (4 vectors before and after the current frame), giving a total of 234 input to the MLP.

Using STRUT, several models have been trained and evaluated. Our goal was to measure the impact of the volumes of training material and the complexity of the MLP architecture on recognition performances with the

\(^9\)Http://ftp.cs.cmu.edu/project/speech/CMU_SLM
obtained acoustic models. We also wanted to compare the hybrid models to the “classical” MutliGaussian models.

**Initial model** The results concerning this model were first introduced in [3].

The initial model is defined as a net with 234 input units, 600 hidden units and 36 output units. The training of the model is done on a subset of 400 speakers (200 male and 200 female speakers) extracted from the Swiss French Polyphone database. The train database contains about 10 phonetically rich sentences for each speaker. However during the experiments different kinds of irregularities as e.g. noise on the recording or strange utterances, were discovered, and so the size of training set was reduced to a final number of 3'272 sentences corresponding to approximately 5 hours of speech. Although it was done only on a subset of the database, the training of the model is a time and CPU demanding task. It takes more than three days on an Sun Ultra-SPARC workstation.

In order to train an MLP, it is necessary to have a segmentation for each utterance, that is, for each time frame, one of the 36 phonemes must be selected as the one being pronounced. Hand-segmentation is a very time consuming task, and an automatic linear segmentation was performed.

For the training data used in the experiments reported here, only orthographic transcriptions were available. To obtain segmentation in phonemes, it was necessary to first convert the orthographic text into a sequence of phonemes; then this phoneme sequence was matched with the speech signal, using a dynamic programming method called forced Viterbi. The result is that each of the phonemes in the phonetic sequence was indeed assigned to one frame. At this point, the MLP training can be performed by the well-known error back-propagation method.

**Enhanced models** The enhancements consisted in increasing the proportion of train material and the number of hidden units in the models. The segmentation used for the enhanced models was produced with the initial model.

Table 4.1 gives information concerning the architecture of the models and the volumes of train data. Table 4.2 reports the scores (i.e. the phone error rates) obtained during the training and cross-validation of the models.

### 4.5 Acoustic Multi-Gaussian models

The model we used is composed of 36 Hidden Markov models, with 3 states per model and 24 Gaussians per state. The used speech features are the mel-frequency cepstral coefficients parameters (MFCC). For each frame we calculate 12 MFCC coefficients, with the delta and acceleration coefficients.
4.6. PHONETIC LEXICON

<table>
<thead>
<tr>
<th>Model</th>
<th>Hidden units</th>
<th>Train data, frames</th>
<th>CV data, frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>600</td>
<td>300'000</td>
<td>23'700</td>
</tr>
<tr>
<td>Enhanced 1</td>
<td>1'000</td>
<td>100'000</td>
<td>300'437</td>
</tr>
<tr>
<td>Enhanced 2</td>
<td>1'500</td>
<td>250'000</td>
<td>500'000</td>
</tr>
<tr>
<td>Enhanced 3</td>
<td>2'000</td>
<td>250'000</td>
<td>500'000</td>
</tr>
</tbody>
</table>

Table 4.1: Sizes of training and test data for hybrid acoustic models

<table>
<thead>
<tr>
<th>Model</th>
<th>Training Score</th>
<th>Cross-validation Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>73.08</td>
<td>69.16</td>
</tr>
<tr>
<td>Enhanced 1</td>
<td>55.67</td>
<td>43.86</td>
</tr>
<tr>
<td>Enhanced 2</td>
<td>56.36</td>
<td>44.06</td>
</tr>
<tr>
<td>Enhanced 3</td>
<td>56.71</td>
<td>44.38</td>
</tr>
</tbody>
</table>

Table 4.2: Training and Cross-validation scores for hybrid acoustic models

We also use the energy coefficient, as well as delta and acceleration of the energy, making a total of 39 coefficients. These features are calculated every 10 ms using a window size of 25 ms with a pre-emphasis coefficient of 0.98. The training of the models was done on the same data as the one used for the initial hybrid model.

A more detailed description of the training of this model can be found in [7].

4.6 Phonetic lexicon

BDLex  French dictionary specifically designed for speech recognition, containing the phonetic transcriptions of 50,000 words and their inflected forms. The use of this dictionary required the adaptation of the software modules accessing that database. Additional pre-processing software was also developed to deal with liaisons and other phonological phenomena in French.

German Speech Dat II (SDGe)  German dictionary with the phonetic transcription of about 23’000 words delivered with the SDGe database.

4.7 Language models

The French language model was trained on the dialogue transcriptions of Wizard-og-Oz experiments performed during the InfoVox project. A corpus of 22’000 words covering lexicon of 2’000 distinct words was used. The German language model was trained using the transcriptions of the dialogues
of Inspire project and the utterances of the German SpeechDat database. The training corpus used for German consisted of 300'000 words covering lexicon of 27'000 words.

4.8 Evaluations of speech recognition module

4.8.1 Global evaluation framework

The evaluation framework is similar to the one used in NIST evaluation campaigns: all the considered models and speech recognition engines are used to recognize the same test data set. The obtained Word Error Rate scores are then used as a measure of the recognition performance.

4.8.2 Acoustic models

The test set is composed of 100 audio recordings extracted from the Swiss French Polyphone. The recordings correspond to spoken digits. The phonetic lexicon consists of 30 words: 28 digits and the two words “star” and “asterisk”.

The results of the evaluation are presented in table 4.3. For MultiGaussian acoustic models, HTK software was used. For all hybrid models, STRUT was used.

The obtained results show that:

R.1: in the framework of our experiments, MultiGaussian technology overperforms the hybrid one

As far as enhanced hybrid models are concerned, notice that despite their better theoretical learning capacities due to their structures and the amount of data used for the training, their performances are largely worse than the one of the initial model. It is interesting to note that the second model is better than the first one during the tests, while being worse during the training.

The substantial difference between the performance on digits recognition with commercial and academic SREs may be due to the fact that the commercial SREs have often the possibility, either to switch between several
acoustic models tuned to specific applications, or to dynamically tune the used acoustic model during the recognition process.

4.8.3 Phonetic lexicon

The goal of this evaluation was to see the impact on the recognition of the variation of the size of the phonetic lexicon. For this purpose, words were iteratively added to the used phonetic lexicon with a starting size of 50 words.

Performance variation associated with the different sizes of the phonetic lexicon is displayed in figure 4.2. They experimentally confirm the quite intuitive hypothesis that

**R.2: the size of the phonetic lexicon has a direct impact on speech recognition performance: bigger lexica lead to worse recognition performance.**

![Image](image.png)

Figure 4.2: Recognition performance with a phonetic lexicon of varying size

4.8.4 Language models

In these experiments, we used the Loquendo engine. The goal of this evaluation was to measure the impact on the recognition performance of the different linguistic resources and their varying adaptation to the domain of application. We concentrated on the impact of the statistical language models and the grammars (a Context-Free Grammar (CFG)). To achieve a controllable variability for resource adaptation, the resources were derived from a corpus containing a set of sentences characteristic for the application (the Inspire WoZ database) and progressively extended with a growing
number of sentences not characteristic for the application (extracted from the German SpeechDat database). The derived CFGs were, of course, not very realistic for real life applications, but they allowed to keep comparable evaluation conditions. For the CFGs, the sentences contained in the training corpora were directly used as rules in grammar, therefore yielding a grammar of the following form:

\[ < S > = (sentence_1|sentence_2|\ldots|sentence_X); \]

To produce the language models, the sentences of the training corpora were used in the traditional way, i.e. for each training corpus, the 3-grams language model was trained.

The evaluation conditions are presented in table 4.4.

<table>
<thead>
<tr>
<th>Audio</th>
<th>Train (utterances)</th>
<th>Test (utterances)</th>
<th>Lexicon (words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inspire</td>
<td>26'370</td>
<td>1'370</td>
<td>430</td>
</tr>
</tbody>
</table>

Table 4.4: Conditions for linguistic resources evaluation

The obtained performance variation is displayed in figure 4.3.

![Figure 4.3: Evaluation with LMs and CFGs](image)

From the obtained results we can see that
R.3: statistical language models (SLMs) and CFG-based models (CFGMs) behave in a substantially different way: increasing size of the training data leads to better performance for the SLMs, even if the training data is not fully adequate for the task for which recognition is performed, while, for CFGMs, increasing size leads to a decrease in recognition performance if the added training data does not correspond to the task.

A possible reason for this can be that the perplexity of the trained CFGs models grows with increase of the training material, while this is not the case for statistical language models for which more training data leads to more accurate estimation of the same N-grams structures.

4.8.5 Speech Recognition Engines

The results of the evaluation of the SREs are presented in table 4.5 for the InfoVox data.

<table>
<thead>
<tr>
<th>System</th>
<th>WER, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTK</td>
<td>63.3</td>
</tr>
<tr>
<td>Nuance</td>
<td>65.0</td>
</tr>
<tr>
<td>Loquendo</td>
<td>66.3</td>
</tr>
<tr>
<td>Sirocco</td>
<td>68.9</td>
</tr>
<tr>
<td>Strut</td>
<td>76.6</td>
</tr>
</tbody>
</table>

Table 4.5: Evaluation of the SREs on the InfoVox data

The obtained results show that

R.4: the academic systems using the HMM technology have a recognition performance similar to the one of the commercial systems.
4.9 Coupling of the speech recognition module with a syntactic analyzer

We propose an evaluation framework integrating the complete effective coupling, i.e. we use a real syntactic analyzer to parse the outputs produced by the speech recognizer. However, in our experiments, the main efforts concentrated on speech recognition.

4.9.1 Evaluation framework

Three kinds of experiments were carried out during the evaluations:

1. preliminary experiments performed using 10 audio recordings. The goal is to prepare the experimental setup and to provide some “intuition” as to the directions to explore;

2. main experiments involving 100 audio recordings. These experiments are designed to verify the assumptions on the word lattice size and parameter space on more realistic data;

3. additional experiments with 500 audio recordings aimed at the validation of the main conclusions for more complex task.

**Speech recognition systems** The systems used during the evaluations were Strut for the preliminary experiments and HTK for the others.

One tuning parameter is used for STRUT: acoustic scaling factor (the option “-acoustic_scale”, hereafter called $\alpha$).

Three parameters are used for HTK: the inter model transition penalty (the option “-p”, hereafter called $p$), the language model scaling factor (the option “-s”, hereafter called $s$), and a number of intermediate hypotheses kept during the acoustic decoding (the option “-beam”, hereafter called $beam$). $p$, $s$ and $beam$ parameters are used for the production of word lattices. $p$ and $s$ parameters are used to verify the assumption on the parameter space. The $beam$ parameter is used to validate the assumption of the word lattice size.

The values of the systems chosen for the experiments are provided in table 4.6 for the STRUT parameter values, table 4.7 for the parameter values of HTK used to verify the assumption on the word lattice size, table 4.8 for the HTK parameter values used to verify the assumption on the parameter space, table 4.9 for the HTP parameter values used for the additional experiments.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>0.1 0.15 0.2 0.25 0.3 0.35 0.7</td>
</tr>
</tbody>
</table>

Table 4.6: STRUT parameters values used for the evaluations
4.9. SYNTACTIC COUPLING

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p )</td>
<td>0.1 0.4 1.0</td>
</tr>
<tr>
<td>( s )</td>
<td>0.1 0.5 1.0 2.5 5.0 7.5 10.0 15.0 20.0 25.0</td>
</tr>
<tr>
<td>beam</td>
<td>1 2 3 4 5 10 20</td>
</tr>
</tbody>
</table>

Table 4.7: HTK parameters values for the assumption on the word lattice size

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p )</td>
<td>0.0 0.1 0.25 0.4 0.75 1.0 5.0</td>
</tr>
<tr>
<td>( s )</td>
<td>0.0 0.1 0.2 0.5 1.0 2.5 5.0 5.5 6.0 6.5</td>
</tr>
<tr>
<td>beam</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.8: HTK parameters values for the assumption on the parameter space

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p )</td>
<td>0.1 0.4 1.0 5.0</td>
</tr>
<tr>
<td>( s )</td>
<td>0.1 0.5 1.0 2.5 5.0 7.5 10.0 15.0 20.0 25.0</td>
</tr>
<tr>
<td>beam</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.9: HTK parameters values for the additional experiments

**Syntactic analyzer**  
The syntactic analyzer used during the experiments was the SLP ToolKit (Syntactical Language Processing Toolkit) developed at the LIA [13]. It is a set of utilities providing correction, tagging, lemmatization and syntactic analysis (using a Context-Free Grammar) of input utterances. The syntactic module of the toolkit is based on an enhanced version of the CYK algorithm [18], which can also be considered as an Early bottom-up algorithm.

**Test data**  
During all the experiments the test data is composed of audio recordings randomly extracted from the Swiss French Polyphone database: 10 recordings for the preliminary experiments, 100 recordings for the main experiments and 500 recordings for the additional experiments. The recordings correspond to spontaneous and read utterances. The length of the utterances varies from 1 to 30 words and their durations from 0.78 second to 16.7 seconds.

The 10 audio recordings used for the preliminary experiments are given in table 4.10.

**Acoustic models**  
The acoustic models used for the experiments are described in section 4.5. As already mentionned, for HTK, it is a MultiGaussian
<table>
<thead>
<tr>
<th>Recording</th>
<th>SFP Index</th>
<th>Transcription</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>f0500s05</td>
<td>la suisse recule dans tous les domaines</td>
</tr>
<tr>
<td>2</td>
<td>f0501s03</td>
<td>le president du parti est aussi le president du comite national</td>
</tr>
<tr>
<td>3</td>
<td>f0501s04</td>
<td>un paradis terrestre possede un fleuve</td>
</tr>
<tr>
<td>4</td>
<td>f0502s00</td>
<td>le plastique a renvoye l'acier dans les limbes de l'age de fer</td>
</tr>
<tr>
<td>5</td>
<td>f0504s01</td>
<td>je n'ai jamais supporte le monde</td>
</tr>
<tr>
<td>6</td>
<td>f0511s06</td>
<td>ils s'installeront en france</td>
</tr>
<tr>
<td>7</td>
<td>f0512s02</td>
<td>les problemes restent nombreux</td>
</tr>
<tr>
<td>8</td>
<td>f0513s01</td>
<td>les causes de cet engouement sont variees</td>
</tr>
<tr>
<td>9</td>
<td>f0518s06</td>
<td>je me sens bien avec eux</td>
</tr>
<tr>
<td>10</td>
<td>f0519s06</td>
<td>la roue tourne</td>
</tr>
</tbody>
</table>

Table 4.10: Test data used during the preliminary experiments

 estimator, with 36 Hidden Markov Models with 3 states per model and 24 gaussians per state. The used features are 26 mel-frequency cepstral coefficients parameters: 12 MFCC and energy coefficients with delta coefficients calculated for each frame of 10 ms.

For Strut, the used acoustic model is the “initial” model, a MultiLayer Perceptron with 234 elements in the input layer, 600 elements in the hidden layer and 36 elements in the output layer. The used features are 12 RASTAPLP coefficients along with delta coefficients as well as delta and delta-delta values of the log-energy, which makes a total of 26 features. The features are calculated every 10 ms using a window of 30 ms. As it is usually done with MLPs in hybrid systems, a context of 9 consecutive feature vectors (4 vectors before and after the current frame) is used for features extraction, leading to a total of 234 features, provided as input values to the MLP.

**Language model**  The language model used by HTK is a bi-grams model trained with the CMU Toolkit using utterances extracted from the Swiss French Polyphone database. The utterances correspond to continuous spontaneous and read utterances. The language model used by Strut in the preliminary experiments consists of 1'186'664 uni-grams and 28'861'312 bi-grams trained on sentences extracted from the *Le Monde* newspaper.

To test the impact of the language model perplexity on the quality of the speech recognition, the language model used by HTK during the main experiments is trained on the material containing test utterances. 31'597 utterances covering 413'757 words are used to train the language model. Its perplexity is 412.95. For the additional experiments, the train material does not contain the test utterances. 38'751 utterances covering 517'098 words
are used to train the language model. Its perplexity is 556.55.

Grammar No real Context-Free Grammar was available for our experiments on sequential coupling. Therefore, a “toy” Context-Free Grammar composed of 140 rules was written in order to cover 10 test sentences presented in table 4.10.

Phonetic lexicon The phonetic lexicon used for the preliminary experiments is composed of 11'064 words including 99 words containing in the test data.

For the other experiments, the lexicon is restricted to the words contained in the test data: 442 words for the main experiments and 1'680 words for the additional experiments. The transcriptions of the words are extracted from the BDLex dictionary.

A synthetic presentation of the characteristics of all train and test data used during the experiments is presented in table 4.11.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Size</th>
<th>Preliminary</th>
<th>Main</th>
<th>Additional</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM train</td>
<td>Utterances</td>
<td>-</td>
<td>31'507</td>
<td>38'751</td>
</tr>
<tr>
<td></td>
<td>Words</td>
<td>-</td>
<td>413'757</td>
<td>517'098</td>
</tr>
<tr>
<td>Test data</td>
<td>Utterances</td>
<td>10</td>
<td>100</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>Phonetic Lexicon</td>
<td>11'064</td>
<td>442</td>
<td>1'680</td>
</tr>
</tbody>
</table>

Table 4.11: Sizes of train and test data

Evaluation metrics The used evaluation metrics are Sentence Accuracy (SA), Lattice Accuracy (LA), Sentence Error Rate (SER) and Lattice Error Rate (LER), Word Accuracy (WA) and Word Error Rate (WER) defined in chapter 2, section 2.6.

SA is defined as the ratio of correctly recognized sentences with respect to the total number of sentences to recognize:

\[ SA = \frac{\text{Number of correctly recognized sentences}}{\text{Total number of sentences to recognize}} \]

SER is defined as:

\[ SER = 1 - SA \]

LA is defined as the ratio of word lattices containing the right utterance with respect to the total number of produced word lattices:

\[ LA = \frac{\text{Number of word lattices containing the correct utterance}}{\text{Total number of word lattices}} \]
Chapter 4. Experimental Validation

LER is defined as:

\[ LER = 1 - LA \]

4.9.2 One-Best approach: results of the evaluations

The first considered problem is the measure of the quality of the One-Best approach. The aim is twofold: (i) to quantify, in our experimental setup, this state-of-the-art approach for speech recognition systems, so as to produce a baseline for the evaluation of the N-Best approaches, and (ii) to train the parameters necessary for the N-Best approaches.

**Preliminary experiments** During the preliminary experiments, self-test evaluation is performed. The recognition results for different values of the scaling factor \( \alpha \) are provided in table 4.12. The values of \( \alpha \) chosen for further experiments are \( \alpha = 0.15 \), \( \alpha = 0.20 \) and \( \alpha = 0.25 \).

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>WER, %</th>
<th>SER, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>55.1</td>
<td>100.0</td>
</tr>
<tr>
<td>0.30</td>
<td>56.5</td>
<td>100.0</td>
</tr>
<tr>
<td>0.35</td>
<td>65.2</td>
<td>100.0</td>
</tr>
<tr>
<td>0.20</td>
<td>65.2</td>
<td>100.0</td>
</tr>
<tr>
<td>0.15</td>
<td>66.7</td>
<td>90.0</td>
</tr>
<tr>
<td>0.10</td>
<td>71.0</td>
<td>90.0</td>
</tr>
<tr>
<td>0.70</td>
<td>108.7</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Table 4.12: One-Best recognition: preliminary experiments

Compared to the results obtained during the main and additional experiments (see results provided in next sections, tables 4.13), higher word and sentence error rates obtained during the preliminary experiments can be explained by bigger size of the phonetic lexicon.

**Main experiments** 100 audio recordings are split into two datasets: the first one is composed of 60 recordings used to train the \( p \) and \( s \) parameters of HTK, the second dataset composed of 40 recordings is used for the evaluation. The training and test procedures are repeated for different values of beam. But the results showed that in that case the same performance, i.e. the same scores, was obtained. The number of intermediate hypotheses does not seem to affect the performance of the speech recognition in One-Best approach.

The five best and five worse results of the training of the parameters are listed in table 4.13. The best scores are obtained with \( p = 0.0 \) and \( s = 7.5 \). The complete results are listed in appendix A.
4.9. **SYNTACTIC COUPLING**

<table>
<thead>
<tr>
<th>p</th>
<th>s</th>
<th>WA, %</th>
<th>WER, %</th>
<th>SER, %</th>
<th>Sub, %</th>
<th>Ins, %</th>
<th>Del, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>7.5</td>
<td>66.7</td>
<td>43.4</td>
<td>78.3</td>
<td>30.4</td>
<td>2.9</td>
<td>10.1</td>
</tr>
<tr>
<td>0.4</td>
<td>7.5</td>
<td>66.5</td>
<td>43.6</td>
<td>78.3</td>
<td>30.6</td>
<td>2.9</td>
<td>10.1</td>
</tr>
<tr>
<td>0.25</td>
<td>7.5</td>
<td>66.5</td>
<td>43.6</td>
<td>78.3</td>
<td>30.6</td>
<td>2.9</td>
<td>10.1</td>
</tr>
<tr>
<td>0.1</td>
<td>7.5</td>
<td>66.5</td>
<td>43.6</td>
<td>78.3</td>
<td>30.6</td>
<td>2.9</td>
<td>10.1</td>
</tr>
<tr>
<td>1.0</td>
<td>7.5</td>
<td>66.4</td>
<td>44.3</td>
<td>80.0</td>
<td>30.7</td>
<td>2.9</td>
<td>10.7</td>
</tr>
<tr>
<td>0.4</td>
<td>0.0</td>
<td>27.3</td>
<td>260.4</td>
<td>100.0</td>
<td>72.5</td>
<td>0.2</td>
<td>187.7</td>
</tr>
<tr>
<td>0.25</td>
<td>0.0</td>
<td>26.9</td>
<td>257.5</td>
<td>100.0</td>
<td>72.9</td>
<td>0.2</td>
<td>184.4</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0</td>
<td>26.9</td>
<td>253.7</td>
<td>100.0</td>
<td>73.1</td>
<td>0.0</td>
<td>180.7</td>
</tr>
<tr>
<td>0.75</td>
<td>0.0</td>
<td>26.4</td>
<td>271.1</td>
<td>100.0</td>
<td>73.4</td>
<td>0.2</td>
<td>197.5</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>26.2</td>
<td>275.0</td>
<td>100.0</td>
<td>73.6</td>
<td>0.2</td>
<td>201.3</td>
</tr>
</tbody>
</table>

Table 4.13: One-Best train recognition: main experiments

Based on these results, it is interesting to note that with the used acoustic and language models, the best recognition performance corresponds to the cases where more confidence is given to the language model. It is also obvious that tuning the word transition penalty does not affect much the recognition performance. These conclusions are confirmed by the results obtained during the additional experiments (see next section).

Three pairs of best values of \( p \) and \( s \) are chosen and used for the evaluation on 40 test utterances. As in the training case, different values of beam do not affect the performance of speech recognition.

The obtained results are presented in table 4.14.

<table>
<thead>
<tr>
<th>p</th>
<th>s</th>
<th>WA, %</th>
<th>WER, %</th>
<th>SER, %</th>
<th>Sub, %</th>
<th>Ins, %</th>
<th>Del, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>7.5</td>
<td>55.4</td>
<td>58.6</td>
<td>82.5</td>
<td>38.4</td>
<td>6.2</td>
<td>14.0</td>
</tr>
<tr>
<td>0.4</td>
<td>7.5</td>
<td>55.0</td>
<td>59.3</td>
<td>85.0</td>
<td>39.1</td>
<td>5.9</td>
<td>14.3</td>
</tr>
<tr>
<td>1.0</td>
<td>7.5</td>
<td>55.0</td>
<td>59.6</td>
<td>87.5</td>
<td>39.1</td>
<td>5.9</td>
<td>14.7</td>
</tr>
</tbody>
</table>

Table 4.14: One-Best test recognition: main experiments

**Additional experiments** During the additional experiments, we use 300 audio recordings to train the parameters and 200 to validate them.

The five best and five worst results are shown in table 4.15. The complete results are listed in appendix B.

The obtained results confirm the conclusions drawn from the main experiments: more confidence in the language model leads to better recognition performances, and word insertion penalty is not important. The overall worse scores are due to the fact that the recognition task is more difficult: 442 phonetic words for the main experiments and 1’680 for the additional ones.
<table>
<thead>
<tr>
<th>$p$</th>
<th>$s$</th>
<th>WA, %</th>
<th>WER, %</th>
<th>SER, %</th>
<th>Sub, %</th>
<th>Ins, %</th>
<th>Del, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>5.0</td>
<td>53.8</td>
<td>63.2</td>
<td>86.0</td>
<td>42.4</td>
<td>3.8</td>
<td>17.1</td>
</tr>
<tr>
<td>0.1</td>
<td>5.0</td>
<td>53.7</td>
<td>63.7</td>
<td>86.0</td>
<td>42.4</td>
<td>3.8</td>
<td>17.4</td>
</tr>
<tr>
<td>0.25</td>
<td>5.0</td>
<td>53.7</td>
<td>63.9</td>
<td>86.0</td>
<td>42.5</td>
<td>3.8</td>
<td>17.6</td>
</tr>
<tr>
<td>0.4</td>
<td>5.0</td>
<td>53.6</td>
<td>64.2</td>
<td>86.3</td>
<td>42.6</td>
<td>3.8</td>
<td>17.8</td>
</tr>
<tr>
<td>0.75</td>
<td>5.0</td>
<td>53.4</td>
<td>65.1</td>
<td>86.3</td>
<td>42.8</td>
<td>3.7</td>
<td>18.5</td>
</tr>
<tr>
<td>0.75</td>
<td>0.0</td>
<td>17.5</td>
<td>277.9</td>
<td>100.0</td>
<td>82.4</td>
<td>0.1</td>
<td>195.3</td>
</tr>
<tr>
<td>0.4</td>
<td>0.0</td>
<td>17.8</td>
<td>270.6</td>
<td>100.0</td>
<td>82.1</td>
<td>0.2</td>
<td>188.4</td>
</tr>
<tr>
<td>0.25</td>
<td>0.0</td>
<td>17.9</td>
<td>267.3</td>
<td>100.0</td>
<td>81.9</td>
<td>0.2</td>
<td>185.2</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0</td>
<td>18.0</td>
<td>264.5</td>
<td>100.0</td>
<td>81.8</td>
<td>0.2</td>
<td>182.5</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>19.1</td>
<td>257.8</td>
<td>100.0</td>
<td>80.7</td>
<td>0.2</td>
<td>176.9</td>
</tr>
</tbody>
</table>

Table 4.15: One-Best train recognition: additional experiments

The evaluation using 200 test utterances gives the results presented in table 4.16

<table>
<thead>
<tr>
<th>$p$</th>
<th>$s$</th>
<th>WA, %</th>
<th>WER, %</th>
<th>SER, %</th>
<th>Sub, %</th>
<th>Ins, %</th>
<th>Del, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>5.0</td>
<td>52.4</td>
<td>64.8</td>
<td>83.5</td>
<td>43.3</td>
<td>4.3</td>
<td>17.1</td>
</tr>
<tr>
<td>0.1</td>
<td>5.0</td>
<td>52.3</td>
<td>64.9</td>
<td>83.5</td>
<td>43.4</td>
<td>4.3</td>
<td>17.3</td>
</tr>
<tr>
<td>0.25</td>
<td>5.0</td>
<td>52.4</td>
<td>65.3</td>
<td>83.5</td>
<td>43.4</td>
<td>4.2</td>
<td>17.7</td>
</tr>
</tbody>
</table>

Table 4.16: One-Best test recognition: additional experiments

Both the main and the additional experiments lead to the result **R.5**
given below.

**R.5:** In our experiments, the Sentence Error Rate remains extremely high: on average, only two transcriptions out of ten are suitable for a reliable natural language processing. At the word level, on average, every second word is misrecognized.

Another interesting observation concerns the fact that the complexity of the used models has an impact on the relative importances of the acoustic and language models: the language model used during the main experiments has more importance than during the additional experiments, because its perplexity is smaller.

### 4.9.3 N-Best: results of the evaluations

We want to validate the idea that improved speech recognition can be achieved with the basic $N$-best approach. Our experimental set-up is the following:

- for each of the test audio recordings, we produce a word lattice using the trained previously values of $p$ and $s$;
• then, for each of the produced word lattices, the set of N-Best hypotheses is extracted and we check either it contains the correct transcription, to produce the lattice accuracy (LA) scores.

In order to validate the assumption concerning the word lattice size, this procedure is repeated for each value of beam.

In addition to the analysis of the presence-absence of the correct utterance in the word lattice (we call this “Lattice analysis”), we also check each of the word lattices for the presence-absence of every word present in the correct utterance. We call this procedure “Word analysis”. This kind of analysis is easy to implement, and it provides an upper bound for recognition performances.

**Preliminary experiments** Using the parameters trained for the One-Best approach, a word lattice is produced for each of the 10 utterances. The analysis of the produced word lattices give the results presented in table 4.17.

<table>
<thead>
<tr>
<th>α</th>
<th>Type of analysis</th>
<th>LA, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.15</td>
<td>Word</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>Lattice</td>
<td>40</td>
</tr>
<tr>
<td>0.20</td>
<td>Word</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Lattice</td>
<td>50</td>
</tr>
<tr>
<td>0.25</td>
<td>Word</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>Lattice</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 4.17: N-best recognition: preliminary experiments

**Main experiments** The used parameter values are $p = 0.0$, $s = 7.5$. The values of the beam parameter are provided in table 4.7. The results of the analysis of the word lattices are presented in figure 4.4.

From these results, the following main conclusion can be drawn:

**R.6: the size of word lattices has an important potential impact on lattice accuracy.** An increase of up to 40% in lattice accuracy can be obtained by increasing the word lattice size.

In our opinion based on the observed results, no additional increase of performance will be obtained for beam values greater than 30. However, due to the high computation cost of the experiences for large values of beam, no experimental confirmation was possible for this hypothesis.

Figures 4.5 to 4.7 present more “exploitable” information concerning the impact of the values of beam on the average computational cost of speech recognition (in CPU-seconds) (figure 4.5), the average size of the produced
Figure 4.4: N-Best approach: recognition performances

Figure 4.5: N-Best approach: average computational cost (in CPU-seconds)

word lattices (in Kilobytes) (figure 4.6), and the average number of hypotheses in the produced word lattices (in log-scale) (figure 4.7).

Notice that the obtained LER is the sum of a $LER_1$ due to words missing in the lattice and of a $LER_2$ due to the impossibility to extract from the word lattice the correct sequence of words in the right order. For example, for $beam = 10$, $LER_1 = 28\%$ and $LER_2 = 34\%$. 
Figure 4.6: N-Best approach: average size of word lattices (in Kilobytes)

Figure 4.7: N-Best approach: an average number of paths in word lattices (in log-scale)

Notice also that the fact that the LER is not null means that it is actually still possible to increase the lattice accuracy by implementing more advanced algorithm for the production of better word lattices. This observation is of important motivation for the additional experiments we have performed with a syntactic parser in the framework of the proposed Enhanced N-Best
CHAPTER 4. EXPERIMENTAL VALIDATION

approach.

Additional experiments The same procedure as the one used during the main experiments is performed during the additional experiments. The results of analyses of the lattices are presented in table 4.18.

As in the case of One-Best experiments, better recognition performances are obtained during the main experiments.

<table>
<thead>
<tr>
<th>Experiment</th>
<th></th>
<th>Type of analysis</th>
<th>LA, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional</td>
<td>0.0</td>
<td>Word</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>5.0</td>
<td>Lattice</td>
<td>31</td>
</tr>
<tr>
<td>Main</td>
<td>0.0</td>
<td>Word</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>7.5</td>
<td>Lattice</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 4.18: N-best recognition: additional experiments

4.9.4 Enhanced N-Best: results of the evaluations

For this approach, the presented results only concern the main and additional experiments.

As already explained in chapter 3, the algorithm used for the Enhanced N-Best approach is the following:

- for each recording, we use the combination of recognition runs with different values of $p$ and $s$ parameters to produce a set of word lattices;

- in our experiments, we merge 40 word lattices to validate the assumption on the word lattice size. For the assumption on the parameter space, we start with merging two word lattices, and then we iteratively merge additional word lattices one by one, and stop when 100 word lattices have been merged\(^\text{10}\);

- the resulting word lattice is parsed for the correct utterance as in the case of N-Best approach.

To validate the assumption on word lattice size, the procedure is repeated for each selected value of $beam$ (see table 4.7). For the assumption on the parameter space, we use $beam = 5$.

\(^{10}\) In fact, to merge several HTK word lattices, it is not sufficient just to concatenate them. This is due to the HTK format used for word lattices. In this format, a lattice node is identified in a relative way. Thus, we had first to convert the nodes identifiers of each lattice to absolute reference.
Main experiments  The results concerning the assumption on the word lattice size are presented in figures 4.8 for the word analysis and 4.9 for the lattice analysis. Results corresponding to the N-Best approach are added to the curves to facilitate the comparison.

Figure 4.8: N-Best and Enhanced N-Best approaches: assumption on the word lattice size (word analysis)

Figure 4.9: N-Best and Enhanced N-Best approaches: assumption on the word lattice size (lattice analysis)
The main conclusion that can be drawn from these results is that

**R.7:** the Enhanced N-Best approach leads to better lattice accuracy compared to the standard N-Best approach. An average relative gain in lattice accuracy of about 10% can be observed on the test data. The best gain is achieved for small values of beam.

The impact of the values of beam on the average computational cost of speech recognition, the average size of merged word lattices and the average number of hypotheses in the produced word lattices are presented in figures 4.10, 4.11 and 4.12, respectively. The data obtained with the N-Best approach is also added to the curves.

![Graph](image)

**Figure 4.10:** N-Best and Enhanced N-Best approaches: average computational cost (in CPU sec)

Figures 4.10, 4.11 and 4.12 show that

**R.8:** compared to the standard N-Best approach, the Enhanced N-Best approach requires more system resources in terms of computational cost and memory, without necessarily leading to substantially higher numbers of hypotheses in the produced lattices.
4.9. SYNTACTIC COUPLING

![Graph 4.11](image.png)

**Figure 4.11:** N-Best and Enhanced N-Best approach: average size of word lattices (in Kilobytes)

![Graph 4.12](image.png)

**Figure 4.12:** N-Best and Enhanced N-Best approaches: average number of hypotheses in word lattices (in log-scale)

The results concerning the assumption on the parameter space are presented in figure 4.13. Five first points are obtained with the values of $s$ and $p$ parameters presented in table 4.19. The other points are the concatenation of other parameter values.
<table>
<thead>
<tr>
<th>Point</th>
<th>p</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>7.5</td>
</tr>
<tr>
<td>2</td>
<td>0.0</td>
<td>5.0</td>
</tr>
<tr>
<td>3</td>
<td>0.0</td>
<td>10.0</td>
</tr>
<tr>
<td>4</td>
<td>0.0</td>
<td>25.0</td>
</tr>
<tr>
<td>5</td>
<td>0.0</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Table 4.19: Enhanced N-best recognition: additional experiments

![Graph](image)

Figure 4.13: Enhanced N-Best approach: assumption on the parameter space

The conclusion that can be drawn from these results is that

**R.9:** it is not necessary to perform a fine-grained exploration of the parameter space to obtain best results for lattice accuracy.

**Additional experiments** The corresponding results are presented in table 4.20

<table>
<thead>
<tr>
<th>Approach</th>
<th>Type of analysis</th>
<th>LA, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enhanced N-Best</td>
<td>Word</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>Path</td>
<td>60</td>
</tr>
<tr>
<td>N-Best</td>
<td>Word</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>Path</td>
<td>31</td>
</tr>
</tbody>
</table>

Table 4.20: N-Best and Enhanced N-best: additional experiments
4.9.5 Coupling speech recognition with a syntactic analyzer

The goal of the preliminary experiments on sequential coupling was to evaluate how well the coupling of the proposed N-Best approaches with a real syntactic analyzer can lead to the conversion of the potential recognition improvement made possible by the use of word lattices into real recognition improvement as measured in terms of WER. The results of the experiments were first introduced in [4, 15].

Table 4.21 provides overview of the experimental set-up for the experiments.

<table>
<thead>
<tr>
<th>Audio</th>
<th>Test data</th>
<th>Software</th>
<th>Grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFP</td>
<td>10 utterances</td>
<td>HTK, Slp ToolKit</td>
<td>140 CFG rules</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100,000 lexical rules</td>
</tr>
</tbody>
</table>

Table 4.21: Experimental set-up for experiments on sequential coupling

The results with and without syntactic coupling are shown in table 4.22. They only concern the preliminary experimental set-up and, as such, have to be considered as preliminary.

<table>
<thead>
<tr>
<th>Coupling</th>
<th>WA, %</th>
<th>WER, %</th>
<th>Sub, %</th>
<th>Del, %</th>
<th>Ins, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>40.9</td>
<td>65.2</td>
<td>54.5</td>
<td>4.5</td>
<td>6.1</td>
</tr>
<tr>
<td>Yes</td>
<td>42.4</td>
<td>62.1</td>
<td>50.0</td>
<td>7.6</td>
<td>4.5</td>
</tr>
</tbody>
</table>

Table 4.22: Recognition performances with and without syntactic coupling

R.10: When combined with a syntactic analyzer, the sequential N-Best approach improves the global speech recognition performance, but the improvement remains relatively weak.
4.10 Simulation of speech recognition

The evaluation of the proposed simulation approach was undertaken on the basis of data acquired during the Inspire project [51] and consisting of 137 utterances. 21 “office conditions”, corresponding to different noising techniques [61], were applied on the data set. The resulting test data sets are recognized by Loquendo speech recognition system [65]. The evaluation of the recognized utterances resulted in a set of WA and WER scores. Our simulation system was then used to treat the 137 sentences with 21 sets of parameters derived from the 21 observed WA and WER scores. The obtained noised sentences are compared to the original sentences and the resulting WA and WER scores are compared to the original Loquendo scores.

The overview of the experimental setup is given in figure 4.14.

![Diagram](image_url)

Figure 4.14: Experimental set-up for the evaluation of speech recognition

The obtained average relative difference is of 1.54%. Table 4.23 gives an extract of the recognition results. Lines 01 *, 07 *, 17 *, 21 * contain results of recognition as produced with Loquendo, while “noised” stands for the recognition results produced with the simulated approach. “rel. diff.” stands for the relative differences between Loquendo scores and simulated approach scores.

The obtained results show
<table>
<thead>
<tr>
<th>Conditions</th>
<th>WA</th>
<th>Subst</th>
<th>Del</th>
<th>Ins</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>01_03a_2</td>
<td>51.8</td>
<td>37.1</td>
<td>11.1</td>
<td>5.0</td>
<td>53.2</td>
</tr>
<tr>
<td>noised</td>
<td>51.3</td>
<td>39.3</td>
<td>9.4</td>
<td>4.6</td>
<td>53.3</td>
</tr>
<tr>
<td>rel. diff.</td>
<td>0.5</td>
<td>2.1</td>
<td>1.7</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>01_03a_3</td>
<td>41.7</td>
<td>47.3</td>
<td>11.1</td>
<td>18.1</td>
<td>76.4</td>
</tr>
<tr>
<td>noised</td>
<td>41.3</td>
<td>52.6</td>
<td>6.1</td>
<td>16.6</td>
<td>75.3</td>
</tr>
<tr>
<td>rel. diff.</td>
<td>0.4</td>
<td>5.3</td>
<td>5.0</td>
<td>1.5</td>
<td>1.1</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07_09_4</td>
<td>43.6</td>
<td>43.0</td>
<td>13.4</td>
<td>5.6</td>
<td>62.1</td>
</tr>
<tr>
<td>noised</td>
<td>43.4</td>
<td>45.6</td>
<td>11.0</td>
<td>4.6</td>
<td>61.2</td>
</tr>
<tr>
<td>rel. diff.</td>
<td>0.2</td>
<td>2.6</td>
<td>2.4</td>
<td>1.0</td>
<td>0.9</td>
</tr>
<tr>
<td>07_09_c</td>
<td>11.9</td>
<td>69.6</td>
<td>18.4</td>
<td>11.9</td>
<td>99.9</td>
</tr>
<tr>
<td>noised</td>
<td>11.2</td>
<td>77.2</td>
<td>11.6</td>
<td>7.1</td>
<td>95.8</td>
</tr>
<tr>
<td>rel. diff.</td>
<td>0.7</td>
<td>7.6</td>
<td>6.8</td>
<td>4.7</td>
<td>4.1</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17_20_c</td>
<td>36.2</td>
<td>48.9</td>
<td>14.9</td>
<td>6.4</td>
<td>70.1</td>
</tr>
<tr>
<td>noised</td>
<td>35.7</td>
<td>51.8</td>
<td>12.4</td>
<td>4.7</td>
<td>69.0</td>
</tr>
<tr>
<td>rel. diff.</td>
<td>0.5</td>
<td>2.9</td>
<td>2.5</td>
<td>1.7</td>
<td>1.1</td>
</tr>
<tr>
<td>21_24_2</td>
<td>47.5</td>
<td>40.0</td>
<td>12.5</td>
<td>5.2</td>
<td>57.7</td>
</tr>
<tr>
<td>noised</td>
<td>47.2</td>
<td>42.7</td>
<td>10.2</td>
<td>4.6</td>
<td>57.4</td>
</tr>
<tr>
<td>rel. diff.</td>
<td>0.3</td>
<td>2.7</td>
<td>2.3</td>
<td>0.6</td>
<td>0.3</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.23: Evaluation of the speech recognition simulation

R.11: there is the good adequacy of the proposed speech error simulation with respect to the real speech recognition engine in terms of word accuracy and a word error rate.

4.11 Evaluations of vocal systems: InfoVox system

The evaluation of the prototype of the InfoVox project was carried out on the basis of two (internal and external) field tests. The results were first presented in [52].

Several subjective and objective indicators have been derived from the raw data produced during the test. Subjective indicators essentially corresponded to average scores obtained for the various closed questions present in the satisfaction questionnaire, while objective indicators have been derived from the logfiles and corresponded to average measures of various system characteristics (such as Word Accuracy, Word Error Rate, interaction dura-
tion, number of Help requests, ...) describing its interaction with each of the users.

Concerning the exploitation of the produced indicators, we mainly focus here on the carried out retrospective trend analysis, i.e. the identification of the subjective indicators corresponding to significantly predominant modalities of some closed question, and can be used to provide retrospectively a synthetic view of the opinion of the users about the system.

In this framework, several conclusions were drawn. However, the two main conclusions that could be derived from this analysis about the InfoVox prototype were that

\[ \textbf{R.12: the average global satisfaction obtained with the used rapid dialogue prototyping methodology was of 63.75\%, which can be considered as a good achievement; in addition, the resulting dialogue-based vocal system was considered as easy to use by 89.8\% of the users.} \]

Concerning the problems of speech recognition weaknesses addressed in previous chapter, the interaction duration was considered as adequate (72.9\%), the sequencing of the questions was considered as natural (93.9\%), the users rarely (14.0\%) felt lost, the majority (79.2\%) of the users were sensitive to the confirmation messages and considered (96.8\%) such confirmations as useful. Finally, concerning dialogue initiative, no clear opinion emerged with respect to predominance of system- or user-initiative.

### 4.12 Conclusion

This chapter presented the results of the evaluation of the different solutions proposed in the previous chapters. A synoptic table containing all the obtained results is presented in appendix C.

Three types of evaluations were considered.

**Speech recognition** The main efforts here concentrated on the training and evaluation of the used acoustic models. We focused on two of the most popular technologies: hybrid models and MultiGaussian models. Our experimental results showed that despite of its strong theoretical advantages, the hybrid technology is outperformed by the MultiGaussian one. The main reason identified to explain this conclusion is the difficulty to train a hybrid model to achieve acceptable recognition performance.

A comparative evaluation of state-of-the-art speech recognition engines was also performed and showed that academic systems using HMM technology have performances similar to the commercial ones.
4.12. **CONCLUSION**

The experimental results also showed that, in the framework of the real world dialogue-based systems we worked with, the speech recognition performance achieved with the One-Best approach remains relatively weak. At the word level, on average, every second word is misrecognized. At the sentence level, on average, only two sentences out of ten can be recognized correctly.

As to the used linguistic resources, several interesting observations were made:

- statistical language models and CFG-based mode behave in a substantially different way: increasing the size of the training data leads to better performance for the statistical language models, even if the additional training data is not fully adequate for the task for which recognition is performed, while for CFG-based models increasing the size leads to a decrease of recognition performance if the added training data does not correspond to the task;

- the size of phonetic lexicon has a direct impact on speech recognition performance: a bigger lexicon leads to worse recognition.

**Coupling** Experimental results showed that the proposed Enhanced N-Best approach leads to an important gain in terms of lattice accuracy, when compared to the standard N-Best approach, but that (i) this solution still does not lead to potentially “perfect” recognition and (ii) the cost of the enhanced N-Best approach is too high to be used in the real applications. Another important conclusion is that it is not necessary to perform a fine-grained exploration of the parameter space to obtain best results.

When combined with a syntactic analyzer, the sequential N-Best approach seems to improve recognition performances at word level, but the improvement remains relatively poor.

**Dialogue** The results obtained for the evaluation of the recognition error simulation showed a very good adequacy of the simulated model with respect to a real speech recognition engine in terms of word accuracy and a word error rate.

As to the proposed solutions to deal with speech recognition weaknesses at the dialogue level, it was not possible to perform tests to evaluate the individual impact of each of the proposed enhancements, but the results showed that the enhanced dialogue prototyping methodology indeed translates into increased global user satisfaction.
Chapter 5

Conclusion

The main goal of the work presented in this dissertation was to present different solutions to overcome some of the weaknesses of continuous multi-speaker speech recognition in the framework of dialog-based vocal systems.

Among the components of a typical vocal system, we mainly concentrated on the acoustic component, the language component and the dialog component. We first focused on the acoustic models that often lead to weak recognition performance due to their inability to provide reliable estimates for phonemes probability distributions. In this context, we showed through experimental validation that, although it is repetitively claimed in the literature that advanced hybrid acoustic models should in principle yield a robust estimates for probability distributions, these models remain in fact often less performant than the more traditional ones based on Multi-Gaussian estimators. More precisely, we showed that one of the central reasons for the failure of the hybrid models is that, due to the huge size of their parameter space, it is extremely difficult in practice to train them to reach a reasonable recognition quality.

The main conclusion of this first part is that despite the already available theoretical results, more research and development work on the training of hybrid models is still needed to make these models really usable in most of the applications facing some real users.

We then concentrated on various ways of increasing the recognition performance by including more advanced syntactic information in the language models. In particular, we studied the sequential coupling of a speech recognition module with a syntactic analyzer. In this context, we proposed different techniques leading to greater linguistic and syntactic variability in the transcriptions (hypotheses) resulting from the recognition, making it possible for the recognizer to benefit from the existing efficient implementations of stochastic parsing algorithms able to process large word lattices. We then
experimentally showed that sequential coupling can indeed provide better recognition at word and sentence level. However, we also showed that the proposed sequential approach suffers from two serious drawbacks: “perfect” (or even almost perfect) recognition still remains out of reach and the practical implementation of the coupling is currently not realistic in terms of computational cost.

The above mentioned conclusions at the acoustic and syntactic levels led us to consider the necessity to deal with speech recognition weaknesses at the pragmatic level, i.e., in our case, at the level of dialog management. Besides several practical solutions concerning the required structure for the dialog model itself, we proposed an enhanced methodology for rapid dialog prototyping integrating the notion of speech recognition error simulation. The proposed simulation model exhibits a behavior similar to the one of real speech recognition engines, with the important advantage of providing the possibility to fully control the required level of recognition. Our approach therefore leads to a more complete dialogue prototyping methodology resulting in more efficient dialogue models that better take into account the limitations imposed by the speech recognition module in the targeted vocal applications.

We believe that the theoretical results and experimental validations obtained on real data in the framework of realistic large scale experiments led us to well substantiated results on two important research issues in speech recognition: the evaluation of practical usability of hybrid acoustic models and the feasibility and performance of the sequential syntactic coupling scheme. In addition, the proposed enhanced methodology for rapid dialog prototyping provides an original and interesting basis for further developments.

As far as future work is concerned, we believe that the following issues should be especially considered:

**Syntactic coupling** The syntactic coupling scheme proposed in our work should be further developed in, at least, the following two main directions: (i) improving the proposed coupling techniques in terms of computational cost in order to make it effectively usable in real-world applications, and (ii) further increasing the syntactic and lexical variability in the produced word lattices in order to further approach the ultimate goal of guaranteeing the presence of the correct solution in the word lattices generated in every situation.

Computational efficiency might be improved by considering dynamic parameter adaptation for the production of the word lattices, instead of the
static approach we used in our work. In addition to the expected lower “computational cost”, this would also lead to a better exploitation of the statistical nature of these parameters.

Another interesting direction to improve syntactic and lexical variability would be to study the combined use of tight and sequential coupling schemes, that are currently considered as distinct solutions and applied separately.

**Dialog modeling** As the iterative nature of the proposed enhanced methodology for rapid dialog prototyping is the key point for the successful implementation of an efficient dialog model, an interesting and useful improvement would consist in the automation of the iterative procedures required for the production of the dialogue models.

For example, the number of intermediate tests involving real users (both Wizard-of-Oz simulation and true field-tests), and thus the associated cost, might be substantially reduced by including in the methodology a notion of *user model*, as well as features such as automated prompt generation along with text-to-speech techniques.
Appendix A

One-Best train recognition: main experiments

<table>
<thead>
<tr>
<th>p</th>
<th>s</th>
<th>WA, %</th>
<th>WER, %</th>
<th>SER, %</th>
<th>Sub, %</th>
<th>Ins, %</th>
<th>Del, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>28.2</td>
<td>247.7</td>
<td>100.0</td>
<td>71.6</td>
<td>0.2</td>
<td>175.9</td>
</tr>
<tr>
<td>0.0</td>
<td>10.0</td>
<td>64.9</td>
<td>41.4</td>
<td>83.3</td>
<td>30.4</td>
<td>4.7</td>
<td>6.3</td>
</tr>
<tr>
<td>0.0</td>
<td>15.0</td>
<td>60.0</td>
<td>43.6</td>
<td>86.7</td>
<td>29.1</td>
<td>10.8</td>
<td>3.6</td>
</tr>
<tr>
<td>0.0</td>
<td>20.0</td>
<td>55.5</td>
<td>47.0</td>
<td>88.3</td>
<td>27.7</td>
<td>16.8</td>
<td>2.5</td>
</tr>
<tr>
<td>0.0</td>
<td>25.0</td>
<td>48.8</td>
<td>52.4</td>
<td>91.7</td>
<td>27.3</td>
<td>23.9</td>
<td>1.3</td>
</tr>
<tr>
<td>0.0</td>
<td>2.5</td>
<td>60.4</td>
<td>69.1</td>
<td>86.7</td>
<td>38.2</td>
<td>1.4</td>
<td>29.5</td>
</tr>
<tr>
<td>0.0</td>
<td>5.0</td>
<td>66.4</td>
<td>49.0</td>
<td>83.3</td>
<td>32.4</td>
<td>1.3</td>
<td>15.4</td>
</tr>
<tr>
<td>0.0</td>
<td>7.5</td>
<td>66.7</td>
<td>43.4</td>
<td>78.3</td>
<td>30.4</td>
<td>2.9</td>
<td>10.1</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0</td>
<td>26.9</td>
<td>253.7</td>
<td>100.0</td>
<td>73.1</td>
<td>0.0</td>
<td>180.7</td>
</tr>
<tr>
<td>0.1</td>
<td>10.0</td>
<td>64.9</td>
<td>41.6</td>
<td>83.3</td>
<td>30.4</td>
<td>4.7</td>
<td>6.5</td>
</tr>
<tr>
<td>0.1</td>
<td>15.0</td>
<td>59.7</td>
<td>44.1</td>
<td>86.7</td>
<td>29.5</td>
<td>10.8</td>
<td>3.8</td>
</tr>
<tr>
<td>0.1</td>
<td>20.0</td>
<td>55.5</td>
<td>47.0</td>
<td>88.3</td>
<td>27.7</td>
<td>16.8</td>
<td>2.5</td>
</tr>
<tr>
<td>0.1</td>
<td>25.0</td>
<td>48.8</td>
<td>52.4</td>
<td>91.7</td>
<td>27.7</td>
<td>23.5</td>
<td>1.3</td>
</tr>
<tr>
<td>0.1</td>
<td>2.5</td>
<td>60.6</td>
<td>68.9</td>
<td>86.7</td>
<td>38.0</td>
<td>1.4</td>
<td>29.5</td>
</tr>
<tr>
<td>0.1</td>
<td>5.0</td>
<td>66.0</td>
<td>49.5</td>
<td>83.3</td>
<td>32.7</td>
<td>1.3</td>
<td>15.6</td>
</tr>
<tr>
<td>0.1</td>
<td>7.5</td>
<td>66.5</td>
<td>43.6</td>
<td>78.3</td>
<td>30.6</td>
<td>2.9</td>
<td>10.1</td>
</tr>
<tr>
<td>0.25</td>
<td>0.0</td>
<td>26.9</td>
<td>257.5</td>
<td>100.0</td>
<td>72.9</td>
<td>0.2</td>
<td>184.4</td>
</tr>
<tr>
<td>0.25</td>
<td>10.0</td>
<td>64.9</td>
<td>41.6</td>
<td>83.3</td>
<td>30.4</td>
<td>4.7</td>
<td>6.5</td>
</tr>
<tr>
<td>0.25</td>
<td>15.0</td>
<td>59.7</td>
<td>44.1</td>
<td>86.7</td>
<td>29.5</td>
<td>10.8</td>
<td>3.8</td>
</tr>
<tr>
<td>0.25</td>
<td>20.0</td>
<td>55.7</td>
<td>46.8</td>
<td>88.3</td>
<td>27.8</td>
<td>16.5</td>
<td>2.5</td>
</tr>
<tr>
<td>0.25</td>
<td>25.0</td>
<td>48.8</td>
<td>52.4</td>
<td>91.7</td>
<td>27.7</td>
<td>23.5</td>
<td>1.3</td>
</tr>
<tr>
<td>0.25</td>
<td>2.5</td>
<td>60.4</td>
<td>69.8</td>
<td>86.7</td>
<td>38.2</td>
<td>1.4</td>
<td>30.2</td>
</tr>
<tr>
<td>0.25</td>
<td>5.0</td>
<td>65.8</td>
<td>49.9</td>
<td>83.3</td>
<td>33.1</td>
<td>1.1</td>
<td>15.7</td>
</tr>
<tr>
<td>0.25</td>
<td>7.5</td>
<td>66.5</td>
<td>43.6</td>
<td>78.3</td>
<td>30.6</td>
<td>2.9</td>
<td>10.1</td>
</tr>
</tbody>
</table>
## Appendix A. One-Best Train Recognition: Main Experiments

<table>
<thead>
<tr>
<th>$p$</th>
<th>$s$</th>
<th>WA, %</th>
<th>WER, %</th>
<th>SER, %</th>
<th>Sub, %</th>
<th>Ins, %</th>
<th>Del, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.0</td>
<td>27.3</td>
<td>260.4</td>
<td>100.0</td>
<td>72.5</td>
<td>0.2</td>
<td>187.7</td>
</tr>
<tr>
<td>0.4</td>
<td>10.0</td>
<td>64.9</td>
<td>41.8</td>
<td>83.3</td>
<td>30.6</td>
<td>4.5</td>
<td>6.7</td>
</tr>
<tr>
<td>0.4</td>
<td>15.0</td>
<td>59.5</td>
<td>44.7</td>
<td>86.7</td>
<td>29.7</td>
<td>10.8</td>
<td>4.2</td>
</tr>
<tr>
<td>0.4</td>
<td>20.0</td>
<td>55.5</td>
<td>47.0</td>
<td>88.3</td>
<td>28.0</td>
<td>16.5</td>
<td>2.5</td>
</tr>
<tr>
<td>0.4</td>
<td>25.0</td>
<td>48.8</td>
<td>52.4</td>
<td>91.7</td>
<td>27.7</td>
<td>23.5</td>
<td>1.3</td>
</tr>
<tr>
<td>0.4</td>
<td>2.5</td>
<td>60.4</td>
<td>70.2</td>
<td>86.7</td>
<td>38.3</td>
<td>1.3</td>
<td>30.6</td>
</tr>
<tr>
<td>0.4</td>
<td>5.0</td>
<td>65.8</td>
<td>49.9</td>
<td>83.3</td>
<td>33.1</td>
<td>1.1</td>
<td>15.7</td>
</tr>
<tr>
<td>0.4</td>
<td>7.5</td>
<td>66.5</td>
<td>43.6</td>
<td>78.3</td>
<td>30.6</td>
<td>2.9</td>
<td>10.1</td>
</tr>
<tr>
<td>0.75</td>
<td>0.0</td>
<td>26.4</td>
<td>271.1</td>
<td>100.0</td>
<td>73.4</td>
<td>0.2</td>
<td>197.5</td>
</tr>
<tr>
<td>0.75</td>
<td>10.0</td>
<td>64.9</td>
<td>42.1</td>
<td>83.3</td>
<td>30.7</td>
<td>4.3</td>
<td>7.1</td>
</tr>
<tr>
<td>0.75</td>
<td>15.0</td>
<td>59.5</td>
<td>44.7</td>
<td>86.7</td>
<td>29.7</td>
<td>10.8</td>
<td>4.2</td>
</tr>
<tr>
<td>0.75</td>
<td>20.0</td>
<td>55.5</td>
<td>47.0</td>
<td>88.3</td>
<td>28.0</td>
<td>16.5</td>
<td>2.5</td>
</tr>
<tr>
<td>0.75</td>
<td>25.0</td>
<td>49.0</td>
<td>52.4</td>
<td>91.7</td>
<td>27.7</td>
<td>23.3</td>
<td>1.4</td>
</tr>
<tr>
<td>0.75</td>
<td>2.5</td>
<td>60.2</td>
<td>72.0</td>
<td>90.0</td>
<td>38.5</td>
<td>1.3</td>
<td>32.2</td>
</tr>
<tr>
<td>0.75</td>
<td>5.0</td>
<td>66.0</td>
<td>50.1</td>
<td>83.3</td>
<td>32.9</td>
<td>1.1</td>
<td>16.1</td>
</tr>
<tr>
<td>0.75</td>
<td>7.5</td>
<td>66.4</td>
<td>44.1</td>
<td>80.0</td>
<td>30.7</td>
<td>2.9</td>
<td>10.5</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>26.2</td>
<td>275.0</td>
<td>100.0</td>
<td>73.6</td>
<td>0.2</td>
<td>201.3</td>
</tr>
<tr>
<td>1.0</td>
<td>10.0</td>
<td>65.3</td>
<td>42.5</td>
<td>83.3</td>
<td>30.4</td>
<td>4.3</td>
<td>7.8</td>
</tr>
<tr>
<td>1.0</td>
<td>15.0</td>
<td>59.5</td>
<td>44.7</td>
<td>86.7</td>
<td>29.7</td>
<td>10.8</td>
<td>4.2</td>
</tr>
<tr>
<td>1.0</td>
<td>20.0</td>
<td>55.9</td>
<td>46.7</td>
<td>88.3</td>
<td>28.8</td>
<td>15.4</td>
<td>2.5</td>
</tr>
<tr>
<td>1.0</td>
<td>25.0</td>
<td>49.4</td>
<td>52.3</td>
<td>91.7</td>
<td>27.5</td>
<td>23.1</td>
<td>1.6</td>
</tr>
<tr>
<td>1.0</td>
<td>2.5</td>
<td>59.9</td>
<td>72.7</td>
<td>90.0</td>
<td>38.9</td>
<td>1.3</td>
<td>32.5</td>
</tr>
<tr>
<td>1.0</td>
<td>5.0</td>
<td>66.0</td>
<td>50.5</td>
<td>83.3</td>
<td>32.9</td>
<td>1.1</td>
<td>16.5</td>
</tr>
<tr>
<td>1.0</td>
<td>7.5</td>
<td>66.4</td>
<td>44.3</td>
<td>80.0</td>
<td>30.7</td>
<td>2.9</td>
<td>10.7</td>
</tr>
</tbody>
</table>
Appendix B

One-Best train recognition: additional experiments

<table>
<thead>
<tr>
<th>p</th>
<th>s</th>
<th>WA, %</th>
<th>WER, %</th>
<th>SER, %</th>
<th>Sub, %</th>
<th>Ins, %</th>
<th>Del, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>19.1</td>
<td>257.8</td>
<td>100.0</td>
<td>80.7</td>
<td>0.2</td>
<td>176.9</td>
</tr>
<tr>
<td>0.0</td>
<td>10.0</td>
<td>51.7</td>
<td>56.1</td>
<td>80.7</td>
<td>39.9</td>
<td>8.4</td>
<td>7.9</td>
</tr>
<tr>
<td>0.0</td>
<td>15.0</td>
<td>44.6</td>
<td>60.8</td>
<td>83.7</td>
<td>41.4</td>
<td>13.9</td>
<td>5.4</td>
</tr>
<tr>
<td>0.0</td>
<td>20.0</td>
<td>37.5</td>
<td>66.2</td>
<td>84.3</td>
<td>43.4</td>
<td>19.0</td>
<td>3.7</td>
</tr>
<tr>
<td>0.0</td>
<td>25.0</td>
<td>29.0</td>
<td>73.6</td>
<td>85.7</td>
<td>44.3</td>
<td>26.7</td>
<td>2.6</td>
</tr>
<tr>
<td>0.0</td>
<td>2.5</td>
<td>47.6</td>
<td>87.5</td>
<td>90.7</td>
<td>50.4</td>
<td>2.0</td>
<td>35.1</td>
</tr>
<tr>
<td>0.0</td>
<td>5.0</td>
<td>53.8</td>
<td>63.2</td>
<td>86.0</td>
<td>42.4</td>
<td>3.8</td>
<td>17.1</td>
</tr>
<tr>
<td>0.0</td>
<td>7.5</td>
<td>53.2</td>
<td>57.8</td>
<td>82.3</td>
<td>40.9</td>
<td>5.9</td>
<td>11.0</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0</td>
<td>18.0</td>
<td>264.5</td>
<td>100.0</td>
<td>81.8</td>
<td>0.2</td>
<td>182.5</td>
</tr>
<tr>
<td>0.1</td>
<td>10.0</td>
<td>51.7</td>
<td>56.1</td>
<td>80.7</td>
<td>40.0</td>
<td>8.3</td>
<td>7.9</td>
</tr>
<tr>
<td>0.1</td>
<td>15.0</td>
<td>44.6</td>
<td>60.8</td>
<td>83.7</td>
<td>41.4</td>
<td>13.9</td>
<td>5.4</td>
</tr>
<tr>
<td>0.1</td>
<td>20.0</td>
<td>37.5</td>
<td>66.3</td>
<td>84.3</td>
<td>43.5</td>
<td>19.0</td>
<td>3.8</td>
</tr>
<tr>
<td>0.1</td>
<td>25.0</td>
<td>29.0</td>
<td>73.6</td>
<td>85.7</td>
<td>44.3</td>
<td>26.7</td>
<td>2.6</td>
</tr>
<tr>
<td>0.1</td>
<td>5.0</td>
<td>53.7</td>
<td>63.7</td>
<td>86.0</td>
<td>42.4</td>
<td>3.8</td>
<td>17.4</td>
</tr>
<tr>
<td>0.1</td>
<td>7.5</td>
<td>53.2</td>
<td>58.0</td>
<td>82.3</td>
<td>40.9</td>
<td>5.9</td>
<td>11.3</td>
</tr>
<tr>
<td>0.25</td>
<td>0.0</td>
<td>17.9</td>
<td>267.3</td>
<td>100.0</td>
<td>81.9</td>
<td>0.2</td>
<td>185.2</td>
</tr>
<tr>
<td>0.25</td>
<td>10.0</td>
<td>51.8</td>
<td>56.2</td>
<td>81.0</td>
<td>40.0</td>
<td>8.2</td>
<td>8.0</td>
</tr>
<tr>
<td>0.25</td>
<td>15.0</td>
<td>44.7</td>
<td>60.8</td>
<td>83.7</td>
<td>41.4</td>
<td>13.9</td>
<td>5.4</td>
</tr>
<tr>
<td>0.25</td>
<td>20.0</td>
<td>37.5</td>
<td>66.3</td>
<td>84.3</td>
<td>43.5</td>
<td>19.0</td>
<td>3.8</td>
</tr>
<tr>
<td>0.25</td>
<td>25.0</td>
<td>29.0</td>
<td>73.6</td>
<td>85.7</td>
<td>44.3</td>
<td>26.7</td>
<td>2.6</td>
</tr>
<tr>
<td>0.25</td>
<td>5.0</td>
<td>53.7</td>
<td>63.9</td>
<td>86.0</td>
<td>42.5</td>
<td>3.8</td>
<td>17.6</td>
</tr>
<tr>
<td>0.25</td>
<td>7.5</td>
<td>53.2</td>
<td>58.2</td>
<td>82.3</td>
<td>41.0</td>
<td>5.8</td>
<td>11.4</td>
</tr>
<tr>
<td>p</td>
<td>s</td>
<td>WA, %</td>
<td>WER, %</td>
<td>SER, %</td>
<td>Sub, %</td>
<td>Ins, %</td>
<td>Del, %</td>
</tr>
<tr>
<td>-----</td>
<td>----</td>
<td>-------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>0.4</td>
<td>0.0</td>
<td>17.8</td>
<td>270.6</td>
<td>100.0</td>
<td>82.1</td>
<td>0.2</td>
<td>188.4</td>
</tr>
<tr>
<td>0.4</td>
<td>10.0</td>
<td>51.8</td>
<td>56.2</td>
<td>81.0</td>
<td>40.0</td>
<td>8.2</td>
<td>8.0</td>
</tr>
<tr>
<td>0.4</td>
<td>15.0</td>
<td>44.7</td>
<td>60.8</td>
<td>83.7</td>
<td>41.6</td>
<td>13.7</td>
<td>5.5</td>
</tr>
<tr>
<td>0.4</td>
<td>20.0</td>
<td>37.5</td>
<td>66.3</td>
<td>84.3</td>
<td>43.6</td>
<td>18.9</td>
<td>3.8</td>
</tr>
<tr>
<td>0.4</td>
<td>25.0</td>
<td>29.4</td>
<td>73.3</td>
<td>85.7</td>
<td>44.2</td>
<td>26.5</td>
<td>2.6</td>
</tr>
<tr>
<td>0.4</td>
<td>5.0</td>
<td>53.6</td>
<td>64.2</td>
<td>86.3</td>
<td>42.6</td>
<td>3.8</td>
<td>17.8</td>
</tr>
<tr>
<td>0.4</td>
<td>7.5</td>
<td>53.3</td>
<td>58.2</td>
<td>82.3</td>
<td>41.0</td>
<td>5.8</td>
<td>11.5</td>
</tr>
<tr>
<td>0.75</td>
<td>0.0</td>
<td>17.5</td>
<td>277.9</td>
<td>100.0</td>
<td>82.4</td>
<td>0.1</td>
<td>195.3</td>
</tr>
<tr>
<td>0.75</td>
<td>10.0</td>
<td>51.9</td>
<td>56.3</td>
<td>81.0</td>
<td>40.1</td>
<td>8.0</td>
<td>8.2</td>
</tr>
<tr>
<td>0.75</td>
<td>15.0</td>
<td>45.1</td>
<td>60.5</td>
<td>83.3</td>
<td>41.5</td>
<td>13.4</td>
<td>5.6</td>
</tr>
<tr>
<td>0.75</td>
<td>20.0</td>
<td>37.5</td>
<td>66.3</td>
<td>84.3</td>
<td>43.6</td>
<td>18.9</td>
<td>3.8</td>
</tr>
<tr>
<td>0.75</td>
<td>25.0</td>
<td>29.6</td>
<td>73.1</td>
<td>85.7</td>
<td>44.4</td>
<td>26.1</td>
<td>2.7</td>
</tr>
<tr>
<td>0.75</td>
<td>5.0</td>
<td>53.4</td>
<td>65.1</td>
<td>86.3</td>
<td>42.8</td>
<td>3.7</td>
<td>18.5</td>
</tr>
<tr>
<td>0.75</td>
<td>7.5</td>
<td>53.1</td>
<td>58.7</td>
<td>82.3</td>
<td>41.2</td>
<td>5.8</td>
<td>11.7</td>
</tr>
</tbody>
</table>
### Appendix C

**Synoptic table of the obtained results**

<table>
<thead>
<tr>
<th>Id</th>
<th>Page</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Speech recognition</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R.1</td>
<td>74</td>
<td>In the framework of our experiments, MultiGaussian technology outperforms the hybrid one</td>
</tr>
<tr>
<td>R.2</td>
<td>75</td>
<td>The size of the phonetic lexicon has a direct impact on speech recognition performance: bigger lexica lead to worse recognition performance</td>
</tr>
<tr>
<td>R.3</td>
<td>77</td>
<td>Statistical language models (SLMs) and CFG-based models (CFGMs) behave in a substantially different way: increasing size of the training data leads to better performance for the SLMs, even if the training data is not fully adequate for the task for which recognition is performed, while, for CFGMs, increasing size leads to a decrease in recognition performance if the added training data does not correspond to the task</td>
</tr>
<tr>
<td>R.4</td>
<td>77</td>
<td>The academic systems using the HMM technology have a recognition performance similar to the one of the commercial systems</td>
</tr>
<tr>
<td>R.5</td>
<td>84</td>
<td>In our experiments, the Sentence Error Rate remains extremely high: on average, only two transcriptions out of ten are suitable for a reliable natural language processing. At the word level, on average, every second word is misrecognized</td>
</tr>
</tbody>
</table>
### Coupling

<table>
<thead>
<tr>
<th>Id</th>
<th>Page</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>R.6</strong></td>
<td>85</td>
<td>The size of word lattices has an important potential impact on lattice accuracy. An increase of up to 40% in lattice accuracy can be obtained by increasing the word lattice size.</td>
</tr>
<tr>
<td><strong>R.7</strong></td>
<td>90</td>
<td>The Enhanced N-Best approach leads to better lattice accuracy compared to the standard N-Best approach. An average relative gain in lattice accuracy of about 10% can be observed on the test data. The best gain is achieved for small values of beam.</td>
</tr>
<tr>
<td><strong>R.8</strong></td>
<td>90</td>
<td>Compared to the standard N-Best approach, the Enhanced N-Best approach requires more system resources in terms of computational cost and memory, without necessarily leading to substantially higher numbers of hypotheses in the produced lattices.</td>
</tr>
<tr>
<td><strong>R.9</strong></td>
<td>92</td>
<td>It is not necessary to perform a fine-grained exploration of the parameter space to obtain best results for lattice accuracy.</td>
</tr>
<tr>
<td><strong>R.10</strong></td>
<td>93</td>
<td>Preliminary experiments tend to show that, when combined with a syntactic analyzer, the sequential N-Best approach improves the global speech recognition performance, but the improvement remains relatively weak.</td>
</tr>
</tbody>
</table>

### Dialogue

<table>
<thead>
<tr>
<th>Id</th>
<th>Page</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>R.11</strong></td>
<td>95</td>
<td>There is the good adequacy of the proposed speech simulation error with respect to the real speech recognition engine in terms of word accuracy and a word error rate.</td>
</tr>
<tr>
<td><strong>R.12</strong></td>
<td>96</td>
<td>The average global satisfaction obtained with the used rapid dialogue prototyping methodology was of 63.75%, which can be considered as a good achievement; in addition, the resulting dialogue-based vocal system was considered as easy to use by 89.8% of the users.</td>
</tr>
</tbody>
</table>
Bibliography


