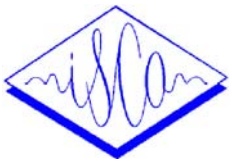

EUROSPEECH '97
5th European Conference on Speech Communication and Technology
Rhodes, Greece, September 22-25, 1997

SPEEDATA: A PROTOTYPE FOR MULTILINGUAL SPOKEN DATA-ENTRY *

U. Ackermann[2], *B. Angelini*[1], *F. Brugnara*[1], *M. Federico*[1], *D. Giuliani*[1], *R. Gretter*[1], *H. Niemann*[2]

[1]IRST – Istituto per la Ricerca Scientifica e Tecnologica
I–38050 Povo, Trento, Italy.
[2]FORWISS – Bayerisches Forschungszentrum für Wissensbasierte Systeme
D–91058 Erlangen, Germany.

ABSTRACT

In this work we describe the development and evaluation of SpeeData, a prototype for multilingual spoken data-entry. The SpeeData project aims at developing a demonstrator that provides a user-friendly interface for spoken data-entry in two languages: Italian and German. A real world application domain is considered, which is the Land Register of an Italian region in which both languages are officially spoken. Original topics of this paper are the interaction modality for spoken data-entry, the evaluation of a data-entry system, bilingual speech recognition, bilingual speaker adaptation.

1. INTRODUCTION

Data-entry can be particularly costly when non electronic information - e.g. contained in documents or pictures - has to be interpreted by a domain expert before being stored into the computer - e.g. medical reporting, diagnostics, cataloging, etc. SpeeData [1] aims at exploring how to gain efficiency in this task by employing state-of-the-art ASR technology. The ideal scenario would be to let the computer work like a secretary who types what the expert user dictates, so that she/he does not have to worry about how data are entered and organized into the computer. Further, the user can verify, correct and store the data through a user-friendly interface.

SpeeData focuses on a real application domain: the Land Register of the Italian bilingual autonomous region Trentino-Alto Adige/Südtirol (RATAA). The Land Register is an institution that since 1897 has been accumulating information about all rights on real estates.

With respect to the application of state-of-the-art speech technology, this data-entry task is challenging for several reasons:

- the complex structure of the data-entry itself;
- the high variety of data types that occur;
- the peculiar way multilinguality has to be managed.

With respect to the latter point an important aspect is that data can be entered in mixed language. For instance, an Italian text describing a real-estate may possibly contain German proper names, like a river or village.

The users of the SpeeData speech recognition system are people living in the Autonomous region of Trentino-Alto Adige/Südtirol. These people have either Italian or German as their mother language and have a different level of knowledge in the other language. Additionally, people with different levels/dialects of German use the system. Therefore, there is a big variety among the speakers.

For this reason, the possibilities have been considered of using two language-dependent unit sets, or a single unit set that encompasses both languages. Further, speaker adaptation of the acoustic models was investigated to compensate the lack in acoustic modeling accuracy for a new system user. A *batch adaptation* technique, which showed to be effective in previous work concerning automatic dictation, has been applied for this purpose.

The following of the paper is devoted to describe the state-of-the-art of the SpeeData prototype, its peculiar technical aspects, an original performance measure for spoken data-entry, and the so far achieved experimental results.

2. STATE-OF-THE-ART

With respect to the user task, the prototype operates with both languages, focuses on the interaction modality, does not yet provide access to a database. From a functional point of view, the prototype concentrate on the data-entry task (e.g. no save, load, etc. operations have been yet covered), but dictation with and without explicit field identification (keyword) is allowed.

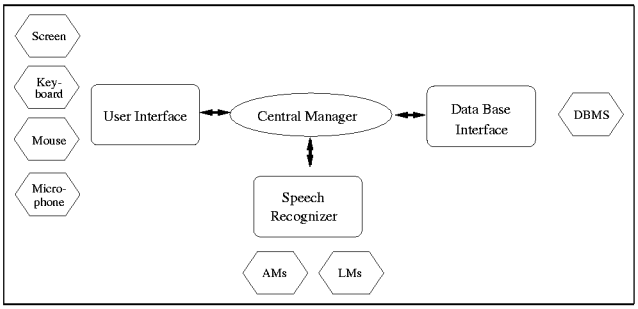

Figure 1: Architecture of the SpeeData demonstrator.

The Central Manager (MGR), the User Interface (UI) and

*This work is supported by the European Commission, Telematics Application Programme, project reference number LE 1999.

the Speech Recognizer (SR) are implemented by separate processes (see Figure 1). In particular, all the modules communicate through socket channels, the UI is implemented in Tcl-Tk, the MGR in Perl, while the SR in C language.

The UI displays forms and active keywords on the screen, forwards speech input to the MGR, manages form-changes, field-activation, and field-assignments commands.

The MGR maintains the state of the whole system. It manages form changes and field activation/assignments, form-to-form transitions, generates state-dependent LMs for the SR module, executes commands, e.g. microphone and idiom switching.

The SR module is based on the CSR system developed at IRST [2]. It dynamically generates search space representations, from the syntactical expressions supplied by the MGR, and uses them to decode the speech input. Moreover, it manages different sets of acoustic models that can be selected by the user. The SR returns command or field assignments.

During a typical session, the user can fill in the fields of the current form by voice, mouse or keyboard. She/he can also execute some general commands, like clean or remove form, open lists of words, change language. Finally, the user can visit and possibly modify forms previously inserted. A summary of the forms inserted so far, including the most significant data entered, is always shown on the screen. State-dependent LMs, generated by the MGR, allow to increase recognition performance. Normally fields can be filled in by means of continuous speech, specifying as many couples keyword-value as needed. Various empty fields can be filled in without pronouncing their respective keys, thus simplifying data insertion. Further, pronouncing an isolated field key selects that field and deactivates LMs of other fields; limiting the recognition to just the LM of one field is very useful in case of corrections.

A problem in such an architecture could be given by consistency between different modules. Thus all the information concerning each form (its layout, the list of reachable forms, its fields, its keys), each field (its data-type, its LM, its keys, an eventual list of values) and each general command is concentrated in a unique description. Each time a modification is made, a script program re-builds all the files needed by the different modules, assuring consistency, for instance, between the fields shown on the screen and the active LMs. Furthermore, introducing new forms, adding or removing fields results very easy, simplifying the porting toward other domains.

3. TECHNICAL CONTENTS

3.1. Language Modeling

Applications like the one foreseen by SpeeData require a system that dynamically adapts the LM to the status of the interaction. In fact, the active LM mainly depends on the form currently shown on the screen, on the eventual selected field, and on the so far filled in fields. Moreover, each data-field owns its specific LM which may either correspond to a simple word list, a hand designed grammar, or a statistically trained LM. This naturally suggest to organize the active LM in a hierarchical way. Starting from the bottom there are *primitive* LMs corresponding to simple data-types like numbers or proper names. Then, there are LMs of data-type that are defined in terms of primitive LMs, e.g. dates or free texts. In fact, free texts are modelled with class based n-gram LMs, where classes correspond to other LMs, e.g. proper names, dates, or even other free texts. All the so defined data-type LMs constitute the repository of static LMs. At any given phase of the interaction, the active LM is dynamically constructed by constraining possible combinations of control commands, keywords, and data-type LMs. All LMs for the data-entry task are represented through *Probabilistic Recursive Transition Networks*. Details about how LMs are implemented and exploited for speech decoding in this application can be found in a companion paper [2].

3.2. Speech Corpus

For acoustic training purposes a bilingual phonetically rich acoustic database was collected. It consists of over 12 hours of read speech uttered by 40 people, 20 having German as mother language and the other Italian. Speakers had a different level of knowledge of the *second* language. Each speaker uttered 80 utterances in both the languages.

For data-entry testing purposes, over 1 hour of read speech uttered by 8 test speakers (4 native German + 4 native Italian) was collected. Test material contains examples, in both languages, of all the application data-types, i.e. keywords, numbers, proper names, dates, free texts, etc. The recordings of training and testing material were performed in a quiet office environment. In order to evaluate the speaker adaptation algorithms, each test speaker also uttered 40 phonetically rich sentences in each language.

	Train		Test		Adapt	
	IT	DE	IT	DE	IT	DE
# Minutes	269	396	73	77	36	39
# Utterances	3163	3162	296	294	320	320
Vocabulary	2529	1050	251	246	228	278
# Speakers	40	40	8	8	8	8

Table 1: Bilingual speech corpus. IT and DE denote the Italian and German part of the corpus, respectively.

3.3. Acoustic Modeling

Starting from the SAM Phonetic Alphabet, three unit sets were thus defined: a set of 50 Italian units, a set of 52 German units and the union of the two, consisting in 81 bilingual units. For German, diphthongs were introduced following the SAMPA documentation, when enough examples were available in the training data.

Continuous Density Hidden Markov Models (CDHMMs) were used to represent the speech units in the three defined sets. Each output probability distribution was modeled with a mixture of 16 Gaussian densities. Gaussian

mixture components had diagonal covariance matrices.

The acoustic front-end produced for each speech frame: 12 mean-normalized mel-scaled cepstral coefficients, the frame energy, and their first and second order time derivatives.

The two language-dependent sets of models were trained exploiting the corresponding training portion of the database. Bilingual models were trained using the training material available for both the languages.

3.4. Speaker Adaptation

In the application under investigation the system should allow the user to indifferently speak in German or Italian, and even mix the two languages. This is the motivation underlying the design of a set of bilingual models. However, it is also true that bilingual models can be less accurate than language-dependent models.

When a speaker uses her/his second language not only inter-speaker acoustic difference but also phonetic variations in the pronunciation can affect recognition performance. In this work, in most cases just a single canonical phonetic transcription was assumed for each word. Unfortunately, this often results inappropriate also for native speaker, since people in the bilingual region have often a strong accent. In order to compensate for the lack of acoustic modeling accuracy, speaker adaptation of the bilingual models was investigated.

MLLR adaptation represents an effective solution for adapting an initial set of Gaussian mixture HMMs to a new speaker using a small amount of speaker-specific speech data [3]. In MLLR adaptation a set of linear transformations, each assigned to a set of model parameters, are estimated in order to maximize the likelihood of the adaptation data. The estimated transformations are then applied to model parameters in order to obtain a set of speaker adapted models. In this work, the MLLR adaptation technique was employed for adapting the means of the Gaussian densities of a set of bilingual CDHMMs.

4. EVALUATION METHODOLOGY

A well known and widely accepted measure to evaluate speech recognition systems is the word accuracy (WA). Given the recognizer's output of an utterance of N words:

$$WA = 1 - \frac{N_{sub} + N_{del} + N_{ins}}{N},$$

where N_{del}, N_{ins}, andN_{sub} respectively indicate the number of deleted, inserted, and substituted words by the speech recognizer.

For data-entry performed in this project, it is also important to estimate the ratio of correct assignments of data to the database fields. Filling-in data fields by speech into a form requires correctly selecting and correctly assigning each field. Selection can be done explicitly by uttering a keyword or implicitly by just uttering the field content. Assuming that the accuracy of a single field assignment utterance is equal to its contents WA when it has been correctly selected, and zero otherwise, the following *filling-in accuracy* (FA) can be defined for a test sample of N field assignment utterances:

$$FA = \frac{\sum_{i=1}^{N} DA_i \cdot WA_i}{N},$$

where $DA_i = 1$ if the i-th field was correctly selected, and 0 otherwise. According to how fields are selected, the above measure can be computed in several ways. For the sake of simplicity, it is assumed here that fields are always selected through keywords. Moreover, in order to compute a more robust estimate of the filling-in accuracy, the selection accuracy DA_i can be replaced by the expectation of the keyword WA, estimated on a test sample containing K isolated keyword utterances:

$$DA_i \approx \bar{DA} = \frac{\sum_{k=1}^{K} WA_k}{K}.$$

It must be noticed that the resulting value underestimates the real accuracy, as all the possible keywords are recognized in parallel and not according to form-dependent subsets.

Given that keywords are always *single* words, WA_k only takes into account substitution errors, hence $WA_k \in \{0, 1\}$. The same happens when WA is computed on data-types like numbers, dates, quotas, percentages, etc. The rationale is that for such data types, recovering from a speech recognition error requires re-uttering the entire content. The same does not apply, of course, to data fields containing free texts, for which single word correction would be preferable. Hence, in this case $WA \in [0, 1]$.

	Test set	
Model set	IT	DE
λ_{IT}	93.9	-
λ_{DE}	-	89.4
λ_{MIX}	94.3	89.5

Table 2: Average filling-in accuracy (%) for three sets of acoustic models: Italian and German monolingual models, and mixed models.

5. EXPERIMENTAL RESULTS[1]

Recognition experiments were carried out using the three sets of HMMs described above. In Table 2 average recognition performance are reported in terms of filling-in accuracy. Recognition results show that bilingual models perform as good as the language-dependent models. So the use of bilingual models results to be a viable solution for bilingual data-entry. However it results that Italian units are better trained than German ones. One reason for this is the German native speakers were really bilingual while the Italian ones often spoke a scholastic German. As an effect, the German acoustic training data results more heterogenous that the Italian one.

In Table 5 the list of the data types considered in the test suite is provided together with the corresponding average recognition accuracy measured with the mixed language models. As for recognition efficiency, the overall

[1]The experiments presented in this section have been carried out at IRST.

Data	Description	LM	Lexicon size		Word Accuracy	
			IT	DE	IT	DE
date	dates, e.g. 10.09.1977	gramm.	55	251	84.0	82.5
distr	cadastral districts	list	156	164	95.0	100.
name	firstnames	list	517	604	97.5	65.0
inscr	type of inscription	list	4	4	100.	100.
num	numbers $\leq 10^{12}$	gramm.	45	189	91.9	85.6
code	number codes with slash	gramm.	42	194	90.0	70.0
part	parcel description	n-gram	946	1560	96.7	91.4
partyp	parcel type	list	2	2	100.	100.
perc	percentages, e.g. 15,9%	gramm.	48	188	97.5	97.5
quota	quotas, e.g. 120345	gramm.	92	195	95.0	77.5
right	type of right	list	25	205	100.	100.
sex	gender of person	list	2	2	100.	100.
sheet	sheet type	list	4	4	96.9	96.9
style	companies, banks, etc.	n-gram	284	633	93.2	90.9
surn	surnames	list	308	564	100.	87.5
texta2	free text of sheet A2	n-gram	1059	965	88.8	98.4
textb	free text of sheet B	n-gram	1507	1099	94.2	83.0
title	title of legal act	n-gram	410	170	98.1	92.9
keywords	names of data fields	list	52	53	98.3	98.2
Average Filling-in Accuracy					94.3	89.5

Table 3: Results of the data-entry test task by employing speaker independent bilingual acoustic units. For each data-type and language, the vocabulary size and data accuracy are reported. The keyword recognition accuracy and the average filling-in accuracy over all data-types are also reported.

processing time was around 0.5 real-time on a 200MHz PentiumPro, and the process size was less than $10Mb$.

The MLLR adaptation technique recalled above was used to adapt the set of bilingual models. For each speaker, model adaptation was performed exploiting 20 adaptation utterances for each of the two languages. Bilingual models were thus simultaneously adapted for both languages. The number of linear transformations employed was fixed to 8 for all the test speakers. In Table 4 some results concerning preliminary adaptation experiments are reported. Rows IT and DE report the average FA of the 4 Italian and the 4 German native speakers, respectively. In the last row, the average FAs are reported by considering all the 8 test speakers together. For each set of speakers, and test set, the average FA is reported using the bilingual speaker independent (SI) models and their speaker adapted (SA) version. Results show that, on average, a significant error reduction can be obtained when adapting bilingual models to a speaker using utterances from both languages. However, for Italian native speakers only a marginal benefit was measured on the German test set. Hence, it is believed that a more specific acoustic training is needed for such speakers.

Speaker	Test IT		Test DE	
	SI	SA	SI	SA
IT native	94.9	96.1	87.2	87.7
DE native	93.6	97.2	91.9	93.6
Average	94.3	96.7	89.5	90.6

Table 4: Average FA (%) on the test speakers using speaker independent (SI) λ_{MIX} models or their speaker adapted (SA) versions.

6. CONCLUSIONS

In this paper the state-of-the-art of the SpeeData project (at May 1997) has been presented. The so far developed prototype system for bilingual data-entry has been outlined and an evaluation measure for evaluating its performance has been introduced. Moreover, the design, training, and speaker adapting of monolingual and bilingual acoustic models has been discussed. Bilingual models are almost required in the SpeeData application and they also performed better, probably because more training data becomes available. Moreover, speaker adaptation significantly improves performance when sufficiently well trained models are available. Performance achieved with the non-native German speakers was not quite satisfactory. Future work will be devoted to improve the acoustic modeling and phonetic transcriptions to cope with strong dialect/accent variations.

7. REFERENCES

[1] U. Ackermann, B. Angelini, F. Brugnara, M. Federico, D. Giuliani, R. Gretter, G. Lazzari, and H. Niemann. Speedata: Multilingual spoken data entry. In *Proc. of ICSLP*, Philadelphia, PA, 1996.

[2] F. Brugnara and M. Federico. Dynamic language models for interactive speech applications. In *Proc. of EUROSPEECH*, Rhodes, Greece, 1997.

[3] C. J. Leggetter and P. C. Woodland. Speaker adaptation of continuous density HMMs using multivariate linear regression. In *Proc. of ICSLP*, vol. 2, pp. 451-454, Yokohama, Japan, 1994.