Towards Best Practice in the Development and Evaluation of Speech Recognition Components of a Spoken Language Dialogue System [†]

Lori Lamel, Wolfgang Minker, Patrick Paroubek
Spoken Language Processing Group
LIMSI-CNRS, BP 133
91403 Orsay cedex, FRANCE

October, 29th 1999

Keywords:

Number of pages: 14

Abstract

Spoken Language Dialog Systems (SLDSs) aim to use natural spoken input for performing an information processing task such as call routing or train ticket reservation (Lamel *et al.*, 1995). The main functionality of an SLDS are speech recognition, natural language understanding, dialog management, response generation and the speech synthesis. This article summarizes key aspects of the current practice in the design, implementation and evaluation of speech recognition components for spoken language dialogue systems. It is based on the frameworke used in the European project DISC.

1 Introduction

Recent years have seen the development of an increasing number of Spoken Language Dialogue Systems (SLDSs), including both commercial and research systems (Peckham, 1993; Guss *et al.*, 1998; ETRW 1999).

Most SLDSs are designed to enable a dialogue between a human (user) and a computer (the SLDS) with no outside intervention of any kind, but when the dialog fails some systems provide operator fallback. This means that the system functionality requires not only an accurate transcription or recognition of the words uttered by the user but also the understanding of the utterances in the context of the application. In the end, such system must make a response as appropriate as possible, be it dialing the correct telephone number, making the correct train reservation or translating a sentence. In many cases several exchanges between the user and the computer are required justifying the term *spoken language dialogue system.*

The DISC project (Dybkjaer *et al.*, 1998; Bernsen *et al.*, 1999) aims at building an in-depth description of the state-of-the-art in SLDSs development and evaluation with the purpose of developing a first best practice methodology in this field accompanied by a series of development and evaluation

[†]This paper is based on research carried out within the ESPRIT 4th Framework LTR Concerted action projects 24823 and 29597 DISC - Spoken Language Dialogue Systems and Components Best Practice in Development and Evaluation.

velopers both from industry and research. Special focus is given on packaging in order to ensure a common representation and a user-friendly access to information.

In a SLDS, the role of speech recognition is to translate the user's utterances (audio signal) into a form that other system components can process (text). Depending on the application and the performance level required, it may be possible to build a SLDS with no or rather limited functionality for semantic analysis, dialogue management or response generation. However, it is impossible to imagine designing a SLDS without the speech recognition functionality, as it is the first module in the analysis process.

However, there exist applications based on speech recognition alone where it is sufficient to transcribe the uttered speech and/or to identify the speaker, for instance voice dictation, video indexing, voice command or speaker verification. Other applications, such as call routing (Abella & Gorin, 1997), information retrieval (Lamel, 1998d; Rosset *et al.*, 1999) and real-time machine translation systems (Maier, 1997) require additional understanding, dialogue management and response generation components in order to allow the system to react accordingly.

2 General Architecture of a Spoken Language Dialogue System

The speech recognizer transforms the acoustic signal into the most probable word sequence. The recognizer output is passed to natural language understanding, which extracts the meaning of the spoken query. The response generation component outputs a natural language response based on the dialogue state, the user utterance, and the information returned from the database. The dialogue manager maintains both the dialogue and the response generation history. Information can be returned to the user in the form of synthesized speech or by using any other dialogue modality depending on the requirement made by the application. Natural-sounding utterances are synthesized by concatenation of variable-sized speech units that are stored in dictionaries. An overview of a generic SLDS architecture (Lamel *et al.*, 1998a) is shown in Figure 1. On the basis of this generic architecture, DISC has

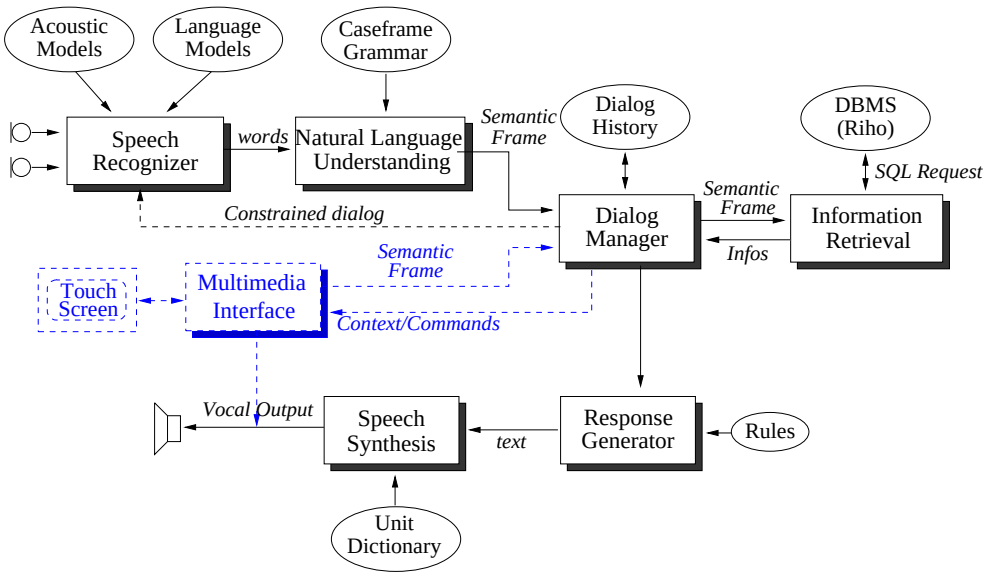

Figure 1: SLDS architecture.

identified six different aspects, that are considered to be essential for SLDSs development. These are

human factors and systems integration.

Currently available speech recognizers are advanced enough, so that users can speak continuously, without placing pauses between words. But the performance decreases rapidly in noisy environments. The recognizers are also generally able to handle any native speaker of the language in question, and are thus referred to as speaker-independent. Making recognizers robust against various kinds of environmental noise (Matrouf & Gauvain, 1998; Bippus *et al.*, 1999) and channel distortion problems (Miksic & Horvat, 1997; Das 1999) is still an active research area.

Today's best performing speech recognition systems use statistical models of speech generation. From this point of view, the message generation is represented by a language model (Katz, 1987; Kneser & Ney, 1995) which provides estimates of $\Pr(w)$ for all word strings w, and the acoustic channel encoding the message w in the signal x is represented by a probability density function $f(x|w)$. The speech decoding problem then consists of finding the most probable word sequence given the input signal. This is equivalent to maximizing the *a posteriori* probability of w, or equivalently, maximizing the product $\Pr(w)f(x|w)$. The principles on which these systems are based have been known for many years. They include the application of information theory to speech recognition, the use of a spectral representation of the speech signal, the use of dynamic programming for decoding and the use of context-dependent acoustic models. Strictly speaking, the aim of decoding is to determine the word sequence with the highest likelihood given the lexicon and the acoustic and language models. In pratice, however, it is common to search for the best path through a trellis (the search space) where each node associates an Hidden Markkov Model state with given time information. Since an exhaustive search for the best path would be prohibitive, techniques have been developed to reduce the computational load by limiting the search to a small part of the search space. The most commonly used approach for small and medium vocabulary size systems is the one-pass frame-synchronous Viterbi beam search which uses a dynamic programming algorithm. This basic strategy has been extended to deal with large vocabularies by adding features such as dynamic decoding, multipass search and N-best rescoring. Despite the fact that some of these techniques were proposed well over a decade ago, considerable progress has been made recently making speaker-independent, continuous speech dictation feasible for relatively large vocabularies. The substantial progress in this domain are due to the availability of large speech and text corpora and by significant advances made in computational processing power facilitating the implementation of more complex models and algorithms.

Depending on the size of the vocabulary they can handle, we can distinguish three major classes of speech recognizers:

1. Small size vocabulary systems (e.g., voice command interfaces) which recognize from 10 to several hundred of words.

2. Medium size vocabulary systems (e.g., SLDSs) which use from a few hundred up to several thousand words.

3. Large size vocabulary (e.g., dictation systems, broadcast news transcribers) with vocabulary of 64,000 words and more.

Speaker independent medium size vocabulary speech recognizers are difficult to bring to an appropriate performance level because training data are sparse and costly to produce and training is one of the main factor which determines the future quality of the system. Small vocabulary independent or connected word recognition systems can usually be designed to be robust by using word specific models. Usually these words can be selected to reduce confusability. Note that it is very hard to rec-

consonants. In general, speech recognition for SLDS uses medium sized vocabulary.

For dictation tasks, it is relatively easy to obtain text data to build the language model. Usually, it is done by first normalizing the text material and then transforming it until its language emulates an observed reading style. After this process, a task vocabulary is selected and language models are trained. A subset of texts can be selected to ensure good phonetic coverage and be used as prompts to obtain spoken data. Obtaining representative data for spontaneous speech is much more difficult and expensive. It is almost impossible to control the content of the speech data, either at the semantic, lexical or phonetic level, or for whatever concerns the speaking style.

Speaker-independence can be obtained by recording speech from many different speakers, in order to cover the speaker population. Phonetic models are relatively task-independent, if many different phonetic contexts are covered in the training corpus. In a more general perspective, it is difficult to design and to train accurate task-independent models that can be used for various applications without the need for additional data collection.

3 Development of Speech Recognition Components

Two sources of influence may guide the development of any application. The first one regroups the intrinsic properties of the application and of its components as well as their various inter-relationships. The second deals with the development process itself. It is generally called life cycle or development cycle in software engineering (Gilbert, 1983).

The DISC project decided to stick to this division and adopted two different points of view (Heid, 1998). A *grid* is used to locate and reference element called *grid properties* which document issues associated with the realization of the modules and of the functionalities of the SLDS under development, while issues associated with the other point of view are simply listed under the label *life cycle properties*. This dichotomy has been refined for each of the six different aspects that have been introduced previously with the SLDS generic architecture (Lamel *et al.*, 1998a).

3.1 Speech recognition grid elements

The DISC grid model intends to give a full generic description fitting both the complete system, as well as their different components and the way these components interact. The grid model consists of a set of questions pertaining to the relevant factual properties of the system or component under scrutiny.

Speech recognizers (c.f. Figure 2) vary in the details of how they are implemented. However, most of them can be accurately described by discussing each of the following grid elements (Lamel *et al.*, 1998b): signal capture, feature analysis, basephone sets, lexicons, acoustic models, language models and, finally decoding (search organization and control). The specification of these elements depends on the application. However, their mutual interaction decides, to what extent they are used.

In (Lamel *et al.*, 1998b) the grid elements are presented in details. Here we describe three speech recognition grid elements, the first one was chosen because it is independent of the application for a given language, the others were chosen for the opposite reason.

Phone set: State-of-the-art systems use phone-based models for words and short phrases rather than word-level models. Since phone sets are application-independent, they allow new vocabulary items to be added without requiring training for the new words. The selection of a phone set to be used in a language is still more an art than a science, as the correspondence between phonemes and phone models is generally close but not exact.

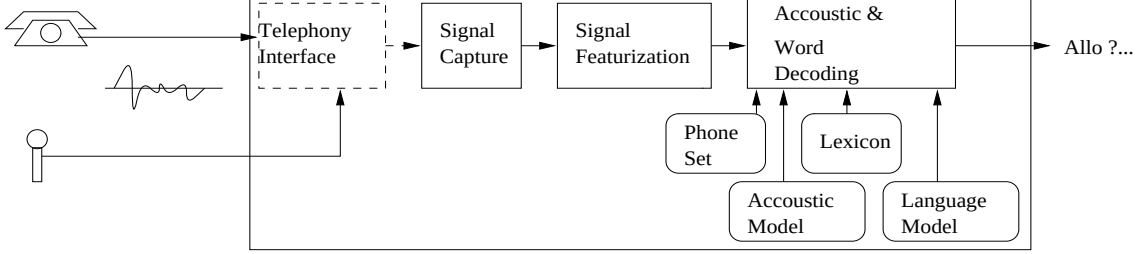

Figure 2: SR architecture.

In general, a set of *basephones* are used that correspond roughly to the phonemes used in the language in question. A typical basephone list will have between 35 and 50 elements. The optimal number of basephones used is determined experimentally based on the language, and for each task. Silence is generally modeled as a single phone. Other common non-speech sounds, such as lip smacks, coughs or breath noises and door slams or beeps may also be included as distinct elementary units.

Lexicons: For spontaneous speech, lexical entries contain, in addition to words from the written language, specific entries for *filler* words such as hesitations or false starts, and for typical noises made by speakers such as *cough.* The definition of words is constrained by the development data and some systems associate frequent word sequences or acronyms, such as *I don't know, Roissy Charles de Gaules*, to a single lexical entry. For most of the current tasks the lexicon used cover in general more than 90% of the utterances even when the vocabulary is a priori unrestricted (for instance in transcribing broadcast news bulletins a vocabulary of around 64k entries is enough). The list of lexical entries is usually determined by automatic processing of the training material transcripts followed by additional hand editing of *obviously* missing items from the task domain such as numbers, days of the week, etc. State-of-the-art large vocabulary speech recognizers are in general capable of handling up to 64,000 words and sometime a bit more. Newer version are expected to work with double that in a near future. All lexicon entries are labeled with one or more pronunciations (sequences of phones drawn from the phone set described above). Many pronunciations are drawn from on-line pronouncing dictionaries or are generated by grapheme-to-phoneme systems that are deemed reliable. Hand editing of these entries is frequently carried out, especially for common words and important words that have multiple pronunciations.

Language Model: N-gram backoff language models represent the state of the art (Zeppenfeld 1997; Lamel 1998c) . The statistics for these models are estimated using training material (transcription from data collected either from system log files or from Wizard of Oz (WOz) experiments (Life *et al.*, 1996; Pirker *et al.*, 1999), in which the automated system is secretly replaced by a human, often completed by large amount of textual material). Various smoothing algorithms are employed, with the (Katz, 1987) and (Kneser and Ney, 1997) models being most common. Some systems use class-based models for common entities, such as dates and times, where the training data may not be representative. Because the amount of language model training data is small, some grammatical classes such as cities, days, months, etc. are used to provide more robust estimates of the N-gram probabilities (in general N equals 2 or 3). Nowadays most language models allow the user to speak in partial sentences and to be free of constraints such as

3.2 Life cycle models for speech recognition components

The DISC dialogue engineering life cycle model draws on a general software engineering life cycle model, but the development process for speech applications differs from that of most other software in that the user interface is significantly more complex and sensitive to underlying system errors. Furthermore, speech applications require the addition of regular tests and final deployments, since tuning performance to user behavior is critical. This requirement introduces several iterations of *usability* analysis and tuning to improve the performance of the speech user interface. Thus, the life cycle for speech recognition is characterized by a highly iterative nature, both within and across the development phases (Bernsen *et al.*, 1998). Note that whatever the level of performance achieved by a systems, speech recognition will never be perfect, therefore error recovery mechanisms need to be provided.

3.3 Stages of a development in life cycle model

In the following, we present the place of speech recognition development activities in the different development phases.

Specification Phase: It begins with a requirement analysis that has two primary goals. The first one is to develop a preliminary user interface design that will identify the framework for building the linguistic coverage of speech recognition (i.e. the grammars and vocabularies to be recognized and understood by the system). Various techniques can be applied to quickly create the user interface design, including interactive role-playing and Wizard of Oz (WOz) testing (Life *et al.*, 1996; Pirker *et al.*, 1999). Concurrent with the user interface design, identification of the functional components of the overall system design should be done with a particular attention to the database and telephony system integration requirements. This approach allows parallel development paths for the speech application development and systems integration, which will converge in the next phase of the development life cycle (den Os *et al.*, 1999).

Development Phase: It consists of first, the creation of the speech application completed in a series of rapid prototyping loops and second, the interaction with hardware and software (e.g. hosts database or telephony middleware). These two paths converge at the systems integration stage, which is the first point in the development life cycle where the application can be subjected a functional end-to-end test. The output of the development phase is a complete and functionality tested application, integrated with the database transaction engine and the telephony network. The functional tests include both user interface and application validation; i.e. they verify whether the interaction logic executes the steps as defined and whether it returns the data requested (Lamel *et al.*, 1996). At this point, the system is ready to take on live interactions, albeit in a controlled fashion, and the project moves into the final phase: the deployment.

Deployment Phase: This final phase of development addresses performance testing, user interface tuning and recognition accuracy tuning. At this stage, the tested system is confronted to end-users in deployment conditions. Based upon user experience and the collection of spoken utterances, both the user interface and the recognition models are tuned in general to balance the highest possible transaction completion rate with the shortest possible dialogue duration. This development may take up to several months.

and deployment. An ideal development team is composed of a Project Manager, Speech User Interface Designer, Speech Recognition Scientist, Application Developer, Systems Integrator, and an Operations Manager.

In a similar way as for the *grid properties*, DISC has identified a set of *life cycle properties* to document aspects of the development life cycle which are specific to speech recognition. These aspects cover a much larger domain that the *grid properties* as they range from purely development cycle specific consideration like those attached to the development team, to functional requirements like the overall design goal or the constraints deriving from end-user specificity.

Development time, teams and problems: The recognizers are generally special case configurations of systems built within the framework of relatively long-term programs, sometimes spaning several generations of a system. Configuration for new tasks and specific development is highly dependent on the previous experience of the team and may require specific knowledge about the application, especially for commercially available systems. The size of the development teams vary from four to fifteen persons, including support staff for data collection and transcription. Average mastery level before starting is highest among the smaller teams. Debugging and problem handling are generally carried out through informal communication, as the size of the teams is generally suitable for this approach.

Overall design goals: Most state-of-the-art speech recognizers are designed as continuous speech, speaker-independent small-to-medium vocabulary systems to be embedded in a spoken language system. They are also designed to run in real-time (or close enough so as to be perceived as real-time) with a minimum word error rate.

The set of *life cycle properties* is particularly useful for planning the deployment of SLDS development activities and for auditing speech recognition development activities.

4 Evaluation of Speech Recognition Components

Most of the protocols currently used for the evaluation of continuous speech recognition systems have been developed while considering the recognizer as a stand-alone application without any specific thought for its use in a SLDS. These methods are principally corpus-based. In this section, we first review fundamental aspects of corpus utilization (Chase *et al.*, 1999), and present less formal methods specific for the evaluation of speech recognition in SLDSs.

When evaluating speech recognition components, we first need to consider whether we have insight to the functionality of the recognizer. In this case, we can then apply *white box* evaluation techniques in order to determine which subcomponent causes some particular error types. In most of the cases, this issue resolves itselfs into choosing to assign the blame to either the accoustic modeling or language modeling. Adding confidence annotations to the transcription has been proposed to solve this problem by (Chase, 1998). It should also be noted, that as far as evaluation is concerned, the vocabulary size of the recognizer has almost no influence on the type of evaluation methodology. Speech recognizers are generally evaluated by comparing their performance on pre-recorded and manually transcribed corpora. A corpus is a collection of spoken utterances, each which is typically not longer than about thirty seconds. They are the result of human transcription at the word level. The corpus is usually divided into three sections:

Training data: It is a typically quite large data subset used to train the system. Often separate bodies of training data are available for the development of acoustic and language models.

ative fashion to tune the system's performance to the characteristics of the data source. The development sub-corpus is typically not as large as the training sub-corpus.

Independent test data: It is taken from the same source as the training and development material. The independent test sub-corpus is typically about the same size as the development sub-corpus and used for independent tests of the tuned system. This data subset must be used sparingly, otherwise unreliable results may be generated due to over-tuning of the system. (After an evaluation cycle, the test data set is often used a development material, and new independent test data is obtained for the next iteration.)

System evaluation aims at comparing algorithms and/or systems in controlled tests, at assessing performance on specific tasks, sometimes in *uncontrolled* or *real* deployments, and at measuring progress both in the laboratory and in the deployment environment.

In practice there are difficulties with each of these goals. For example, a detailed comparison of algorithms or systems requires a potentially large number of contrasts to be measured. This is at odds with the need to produce statistically significant results, that can only be obtained if large test sets are used. Meeting both goals is be computationally expensive. Evaluating the performance of a deployed system generally adds to the already significant burden of system maintenance. All of these competing demands must be considered when evaluating speech recognition components.

4.1 Components for systems evaluation

This section reviews the key components required to evaluate a speech recognizer. These are:

- corpora, and possibly the means to add *live* data collected from deployed systems,
- appropriate transcription protocols and/or text normalization routines,
- scoring methods and analysis tools to determine the significance of the results, and
- for *live* deployed systems, tools for parallel listening and/or signal archiving as well as creation of log files for analysis of caller/system interactions.

Data The acoustic data must be annotated at the level where the recognition is supposed to take place (e.g. words or characters). Any artifacts, like noise or music, must be marked.

Data used for training of models should normally be taken from the same domain as the test material. Data for a development test is essential for preparation and system tuning. These development data is best sampled from the same corpus as the test subset, but from a different epoch in the case of time-sensitive materials.

In order to track the technology improvement in controlled laboratory conditions, the evaluation data should keep the level of difficulty constant from one test to another. For continuous speech, the test set word perplexity is often used as the primary metric in making this judgement. However, there are two problems with word perplexity. The first one is that perplexity is sensitive to the average word length. The second is that perplexity is calculated with respect to a particular language model, implying that the task difficulty can only be evaluated with respect to a particular reference point.

Other measures which correlate with the level of difficulty of a recognition task include the quality of lexical coverage for the task in question, the speech rate, the disfluency rate, the amount of mumbling or faint speech, of foreign words and the number of non-native speakers of the language employed in the sample data.

curacy. Measures of these can be used as the basis of contrast conditions to further study their impact. They can also be used to control the match in difficulty between development and evaluation data, and to ensure a smooth transition in difficulty for successive editions of official tests.

For measuring improvement in deployed systems, the general rule is to consider any and all data that is collected under real conditions as part of the evaluation test set. While this may not lead to easy analysis of results, it often leads to identification of important system improvements.

Scoring The key measure of continuous speech recognition systems is the percentage word error rate (WER). It is a proportional count of word errors made with respect to the human-produced word transcripts. This is usually computed while respecting utterance boundaries, but the errors are usually aggregated across the whole test set to give the overall results.

Given a reference word string containing N words and a recognition hypothesis, the WER is determined by first aligning both word strings and then counting the number of substitutions (S), deletions (D) and insertions (I).

$$WER = \frac{S + I + D}{N} \times 100\% \tag{1}$$

A more detailled presentation of the measure is available in the DISC deliverable: (Chase *et al.*, 1999). Toolkits for scoring speech transcription are available, some of them are even freely available, for instance the National Institute of Standards and Technology (NIST) standard scoring package[1].

This simple alignment scheme performs well for read speech, for which the WER is generally small. However, it inappropriately minimizes the error rate at high values of WER (Hunt, 1988). Consequently, a method which depends on the phonological distance between words was investigated. It uses distinctive features derived from a set of assumed lexical word base forms (Fischer *et al.*, 1995). This method did yield improved diagnostic capabilities and reduced biased measurement errors (Hunt, 1988). But because it requires a dictionary (backed up by a default general-purpose text-to-phone function) it is much more complex to use. More detailed methods have been investigated in the past, for instance in the context of the 1996 Switchboard DARPA evaluation, where time stamps were used. Automatic alignments of reference transcripts was judged too inaccurate, and human annotators were employed, in spite of the much higher cost. Following this experiment, it was decided that the potential benefits of the more detailed scoring method were not substantial enough to justify their adoption in the DARPA speech recognition evaluation protocol.

Statistical analysis Evaluation gives an idea of the speech recognition performance with respect to other systems, to different versions of the same system or to a predefined target. But are these answers significant? Here, statistical analysis is of a great help to ensure, within a certain margin of certainty, that the results measured are really consequences of the system characteristics and not arbitrary results. In the same spirit as for the kappa statistics (Cohen, 1960, 1968; Krippendorff, 1980) which is now widely used in computational linguistics to measure inter-annotator agreement, a series of tests are currently used when evaluating speech recognition. The most representative are those which have been used in the DARPA evaluation of large vocabulary continuous speech recognizer evaluations (Gillick & Cox, 1989).

MP: Matched Pair Sentence Segment (Word Error) Test

SI: Signed Paired Comparison (Speaker Word Accuracy) Test

[1] available at http://www.itl.nist.gov/div894/894.01/software.htm.

MN: McNemar (Sentence Error) Test

These tests are applied between pairs of contrast conditions or systems. The *MP* test is based on the null hypothesis that the mean difference in the number of word errors per sentence is zero. *SI* and *WI* are standard statistical nonparametric tests to determine whether or not two pairs of samples are from the same distribution, where in this case the samples are speaker word error rates. The *MN* test is based on the count of the errors made by one of the pair but not the other compared to the total number of errors that are not common to both systems. The null hypothesis is that this ratio should be divided by two.

Adapting Transcription and Scoring Practices The scoring methods discussed in the previous paragraph rely on the presence of an accurate reference transcription. It is clearly important to decide what exactly should be captured during transcription and how it should be represented.

When working in a given language, some words are bound to appear in the training and test sections of the corpus with multiple spellings, including misspellings. Many languages include homophones which are distinct as written words. These are normally not treated as equivalent under the weighted error metric and must be correctly spelled in order to be scored as correct. In English, for example, three homophones appear in the sentence, *I'd like to write to Mr. Wright right now.* Each of these would have to be spelled correctly in the recognizer output in order to be judged as correct. In the European SQALE project (Young *et al.,* 1997) the relatively high homophone rate in French was an important issue.

For languages such as English that commonly use contractions, it must be decided whether or not to define the reference transcription by expanding contractions to their underlying form often with mapping rules applied to both the reference and hypothesis (output of a system) data. But this operation changes the number of words compared. There is no general practice in this matter, decision depend on the context of the evaluation.

All of these variations indicate the flexibility brought in by using the weighted error metric. They also show that this metric should carefully be used. Attention should be paid in advance to the issues that might arise for any new application.

4.2 Less formal evaluation methods

Especially when dealing with deployed systems, it is often important to employ less formal evaluation methods in order to truly understand how well the speech recognizer is working.

There are several phases involved in evaluating a speech recognizer under these conditions, each with its own set of important issues:

Specification Phase: The specification phase begins the development of a preliminary user interface design that will identify the framework for building the speech recognition contexts (i.e., the grammars and vocabularies to be recognized and understood by the system). Several techniques may be employed to quickly create the user interface design including interactive role-plays and WOz testing (Fraser & Gilbert, 1991), in which the automated system is secretly replaced by a human. This specification phase should include laboratory-style formal evaluations of the newly specified recognizer, with tests performed on whatever corpora seem to best match the deployment environment.

Development Phase: The development phase should involve the creation of the speech application itself which is both accuracy tested in isolation and then usability tested during integration in

by norm ISO9126 as *Attributes of software that bear on the provision of right or agreed results or effects* (ISO9126, 1991) and usability is defined by the same norm as a software quality characteristic composed of *A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users* (King *et al.*, 1996). Tools for listening to test users and analyzing error in parallel are essential in this phase for identifying unexpected problems or weak points in the recognizer. After this evaluation step (Gauvain *et al.*, 1996; Lamel 1998d, Lamel 1999), the system is ready to be deployed in the environment for which it was designed.

Deployment Phase: The final phase addresses performance testing, user interface tuning and recognition accuracy tuning. This stage involves testing of all system components in parallel with the speech recognizer. Based upon user experience and the collection of spoken utterances, both the user interface and the recognition models have to be tuned in parallel. The same *listen and analyze* tools that were used in the development phase can be used here again to improve the recognizer accuracy along with the performance of other components, thus increasing the global performance of the overall SLDS.

5 Conclusion

Even though rapid progress has been made in large vocabulary speech recognition components, many factors may influence the speech recognition performance. Many outstanding problems still remain to be resolved, for instance, inter-speaker variability, speaking rate, and lexical and language modeling.

Due to inter-speaker variability, even today's best systems show a significant difference in performance between the word error of the best speaker (1-2%) and the word error of the worst speaker (25-30%). These performance variations are often due to differences in speaking rate, notably if the locutionary style is much faster or slower than the average. Differences in speaking rate affect not only the acoustic level, but also the phonological and even the word level. At the lexical level, it should be possible to choose among pronunciation variants according to observed pronunciations for a given speaker (a person pronouncing a word in a given way is likely to produce derived forms, and other similar words in a similar way). At the cross-word level, different speakers make use of different phonological rules. Despite the fact that for most speakers, the choice of rules is systematic, no state-of-the-art system is able to make use of this consistency.

These are outstanding problems. More generally, today's systems do not easily adapt to different accents, either from dialects or from non-native speakers. The technology needs to make substantial progress in this area to obtain a performance level comparable to the one achieved by humans. Despite the fact that attempts at crafting generic best practice guidelines for software development began to appear early in the history of computer science, nothing has been done for SLDSs prior to the DISC project to our knowledge.

Since SLDS and in particular speech recognition modules are becoming a common facility in industry, the DISC results provide essential information to established development teams, auditing teams and decision planning to use or develop speech technology for SLDSs. Of course, the current guidelines provide a snapshot of the field corresponding to the state-of-the-art at a particular time. In order to make a live resource of the guidelines (which is a condition for their long term usability) the DISC project used the feedback provided by an Industrial Advisor Pannel at regular intervals throughout the project. It is actively seeking a solution for the maintenance and upgrading of the guidelines after completion of the project, for instance in collaboration with excellency networks like ELSNET in Europe (Krauwer, 1999).

6 References

Abella A., and Gorin A.L. (1997), "Generating Semantically Consistent Inputs to a Dialog Manager", *In Proceedings of Eurospeech*, Rhodes, Greece, pp. 1879-1882.

Bernsen N.O., Dybkjaer H., and Dybkjaer L. (1998), "Designing Interactive Speech Systems. From First Ideas to User Testing," Springer Verlag, Berlin, Heidelberg.

Bernsen N.O., Dybkjaer L., and Heid U (1999), "Current Practice in the Development and Evaluation of Spoken Language Dialogue Systems," *In Proceedings of Eurospeech*, Budapest, Hungaria, pp. 1147-1150.

Bippus R., Fischer A., and Stahl V. (1999), "Domain Adaptation for Robust Automatic Speech Recognition in Car Environments," *In Proceedings of Eurospeech*, Budapest, Hungaria, pp. 1943-1946.

Chase L. (1998), "Evaluating Word Confidence Annotation for Speech Recognition Systems," *In Proceedings of the First International Conference on Language Resources and Evaluation*, Granada, Spain, pp. 167-173.

Chase, L, Lamel L., and Paroubek P. (1999) "Guidelines and Testing Protocols for the Development of Speech Recognition Components for SLDSs., DISC Deliverable D2.2.

Cohen J. (1960), "A coefficient of agreement for nominal scales," Educational and Psychological Measurement, 20:37-46.

Cohen J. (1968), "Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit," Psychological Bulletin, (70)4:213-220.

Das S., Lubensky D., and Wu C. (1999), "Towards Robust Speech Recognition in the Telephony Network Environment - Cellular and Landline Conditions," *In Proceedings of Eurospeech*, Budapest, Hungaria, pp. 1959-1962.

Dybkjaer L., Bernsen N. O., Carlson R., Chase L., Dahlbäck N., Failenschmid K., Heid U., Heisterkamp P., Jönson A., Kamp H., Karlsson I., Kuppevelt J.v., Lamel L., Paroubek P., and Williams D (1998), "The DISC Approach to Spoken Language Dialog Systems Development and Evaluation," *In Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, Granada, Spain, pp. 185-189.

ETRW - ESCA Tutorial and Research Workshop (1999) on "Interactive Dialogue in Multi-Modal Systems", Proceedings, Kloster Irsee, Germany, June 22-25, 1999.

Fisher W.M., Fiscus J.G., and Martin A. (1995), "Further Studies in Phonological Scoring," *In Proceedings of the ARPA Spoken Language Workshop*, Austin, USA, pp. 181-186.

Fraser N., and Gilbert, N. (1991), "Simulating speech systems," Computer, Speech, and Language, 5(1), pp. 81-99, January 1991.

Gauvain J.L., Gangolf J.J., and Lamel L. (1996), "Speech Recognition for an Information Kiosk," *In Proceedings of the International Conference on Speech and Language Processing*, Philadelphia., USA.

Gilbert P. (1983), "Software Design and Development," chapter in Science Research Asso-

Gillick L. and S. Cox (1989), "Statistical Significance Tests for Speech Recognition Algorithms," *In Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Glasgow, Scotland, pp. 532-535.

Guss N., Judge P. C., Port O. and H. Wildstram (1998), "Let's talk!" Business Week. McGraw-Hill. February, 1998.

Heid U. (1998), "Dialogue Engineering Best Practice Model - Outline Skeleton," DISC-Deliverable D1.1., pp. 1-9.

Hunt M.J. (1988), "Evaluating the Performance of Connected Word Speech Recognition Systems," *In Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, New York, USA, pp. 457-460.

International Standard ISO/IEC 9126 (1991). Information technology - Software product evaluation - Quality Characteristics and guidelines for their use. Geneva, International Organization for Standardization, International Electrotechnical Commission.

Katz, S. (1987), "Estimation of probabilities from sparse data for the language model component of a speech recognizer," IEEE Transactions on Acoustics, Speech and Signal Processing (ASSP), 35:400-401.

King M., Maegaard B., Schutz J., Tombe L.D. et al. (1996), "Evaluation of Natural Language Processing Systems," In EAGLES Final Report, EAG-WEG-PR.2.

Kneser, R., and Ney. H. (1995), "Improved backing-off for M-gram language modeling," *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Detroit, USA.

Krauwer, S. (1999), "ELSNET in FP5," Elsnews 8(3):1.

Krippendorff K. (1980), "Content Analysis: An Introduction to Its Methodology," Sage Publications, Beverly Hills, USA.

Lamel L., Rosset S., Bennacef S., Bonneau-Maynard H., Devillers L. and J.-L. Gauvain (1995), "Development of spoken language corpora for travel information. *In Proceedings of the European Conference on Speech Technology*, EuroSpeech, volume 3, pages 1961-1964, Madrid, September 1995.

Lamel L., Gauvain J.L., Bennacef S.K., Devillers L., Foukia S., Gangolf J.J., and Rosset S. (1996), "Field Trials of a Telephone Service for Rail Travel Information," *In Proceedings of IVTTA*.

Lamel L., Bennacef S., Gauvain J.L., Dartigues H., and Temem J.-N. (1998a), "User Evaluation of the MASK Kiosk", *In Proceedings of the International Conference on Speech and Language (ICSLP98)*, volume 7, Sydney, Australia, December 1998, pp 2875-2878.

Lamel L., Chase L., and Paroubek P. (1998b) "Working Paper on Speech Recognition Current Practice," DISC-Deliverable D1.2., pp. 1-26.

Lamel L, Rosset S., Gauvain J.-L., Bennacef S. (1998c), "The Limsi ARISE System", IVTTA'98, Torino, pp. 209-214, September 1998.

Lamel L (1998d). "Spoken language dialog system development and evaluation at LIMSI". *In Proceedings of the 1998 International Symposium on Spoken Dialogue*, Sydney, Australia, November 1998.

travel information". *In Proceedings of the IEEE International Conference On Acoustics*, Speech, and Signal Processing, Phoenix, April 1999.

Life A., Salter I., Temem J.N., Bernard F., Rosset S., Bennacef S. and Lamel L. (1996), "Data Collection for the MASK Kiosk: WOz vs. Prototype System," *In Proceedings of the International Conference on Speech and Language Processing*, Philadelphia, USA.

Maier E. (1997), "Clarification Dialogues in Verbmobil," *In Proceedings of Eurospeech* Rhodes, Greece, pp. 1891-1894.

Miksic A., and Horvat B. (1997), "Subband Echo Cancellation in Automatic Speech Dialog Systems," *In Proceedings of Eurospeech*, Rhodes, Greece, pp. 2579-2582.

den Os E., Boves L. Lamel L., and Baggia P. (1999), "Overview of the ARISE Project," *In Proceedings of Eurospeech*, Budapest, Hungaria, pp. 1527-1530.

Peckham J. (1993), "A New Generation of Spoken Dialog Systems: Results and Lessons from the SUNDIAL Project," *In Proceedings of Eurospeech*, Berlin, Germany.

Pirker H., Loderer G., Trost H. (1999), "Thus Spoke the User to the Wizard," *In Proceedings of Eurospeech*, Budapest, Hungaria, pp. 1171-1174.

Rosset S., Bennacef S., and Lamel L. (1999), 'Design Strategies for Spoken Language Dialog Systems," *In Proceedings of Eurospeech*, Budapest, Hungaria, pp. 1535-1538.

Young S. J., Adda-Dekker M., Aubert X., Dugast C., Gauvain J.L., Kershaw D.J., Lamel L., Leeuwen D., Pye D., Robinson A.J., Steeneken H.J.M., and Woodland P.C. (1997), "Multilingual Large Vocabulary Speech Recognition: The European SQALE Project," Computer Speech and Language. 11:73-89.

Zeppenfeld T., Finke M., Ries K., Westphal M. and Waibel A. (1997). "Recognition of Conversational Telephone Speech using the Janus Speech Engine". In the proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 1997.