ICT for development: people, policy and practice.
IDIA2011 Conference Proceedings. 80-99

# A framework for developing speech recognition systems for resource-scarce languages

Dr. Melih Kirlidog
Marmara University
Computer Engineering
Computer Engineering Department
Marmara University - Goztepe
34722 Istanbul
Turkey
melihk@marmara.edu.tr
melihk76@gmail.com
and North-West University, South Africa

Etienne Barnard
North-West University, South Africa

***Abstract***

*Automatic speech recognition systems are commonly used in the developed world for a wide range of applications. Their usage in the developing world, however, is much less common. To some extent this is a result of the widespread perception that the development of such systems for resource-scarce languages is not feasible. Recent advances in speech technology offer several tools and techniques to circumvent this difficulty. We present an analysis of the requirements for the development of speech recognition systems in resource-constrained environments, and describe some of the recent developments that contribute to a significantly better outlook for the development of such systems in the near future.*

## Introduction

The most common way of interacting with computers is through visual communication. This is true for both human-to-computer interaction and vice versa. There are  two main types of visual communication with computers, namely the command line and point-and-click methods. In the older command line method the commands for driving the computer are typed in by the user which are shown on the screen, and the computer can respond in a number of ways, such as displaying the result on the screen or printing it out on paper. In the last few decades the point-and-click method has been the dominant mode of issuing commands to computers. This is associated with the GUI (graphical user interface) environment. In this method the commands are usually issued by placing the pointer over an icon or text on the screen and clicking on it with the mouse.

Both of these methods are, however, clearly not the “native” mode of human interaction, and require humans to know the rules of the computer in order to communicate with it. This mainly stems from the fact that advances in the interaction between computers and humans have lagged far behind advances in other areas of computer technology.

The “native” and easiest way of human interaction is speech, which does not require any formal education and is readily available to the vast majority of all humans. However, it is an irony that quite sophisticated computer technology is required for “native” and “easy” spoken communication with humans. Although this technology is available today, it is more prone to errors than the more established text-based interaction. Benzeghiba et al. (2007) argue that differences in age, gender and speaker physiology pose certain difficulties for automatic speech recognition (ASR) systems. The differences between native and foreign-accented speech and the emotional state of the speaker affect the performance of these systems.

Interacting with computers through speech can nevertheless be an important tool for people who live in developing areas of the world. Such areas are usually associated with limitations in basic formal education which is normally required for reaping the benefits of information technology. Illiteracy leads to the lack of the prerequisites for interacting with computers in “conventional” ways, i.e. through the keyboard and screen. This is also true for visually impaired people. Interacting through speech can be a solution in such cases. Culturally, people in developing countries tend to favour oral rather than written communication (Metcalfe, 2007). Parikh &

Lazowska (2006) found that semi-literate and illiterate people prefer speech-based interaction when they need to access information services. Speech technology can fulfil this requirement and computers can play a much larger role in addressing developmental issues. The dissemination of computers in developing areas is less than ideal, but telephone networks can provide the necessary communication based on speech technology. GSM technology, which is regarded as one of the fastest diffused technology in history (Boretos, 2007), is particularly encouraging due to its widespread availability in disadvantaged areas.

There are, however, a number of linguistic and technical prerequisites for this to be realised. Linguistic problems mainly stem from the fact that, unlike in developed areas of the world, many languages in developing areas lack the phonetic, phonological and syntactic studies which are required for speech technology. The situation is aggravated by the fact that such areas are often characterised by a variety of languages and dialects. This means that the linguistic studies required for speech technology need to be replicated for each language in an area.

There is a growing body of literature on speech recognition systems (SRSs) in developing country settings. Several authors report a wide range of SRS applications in a number of countries, for example, see Nasfors (2007) and Patel et al. (2010) for agriculture applications in Kenya and India respectively, and Sherwani et al. (2007) and Sharma et al. (2009) for healthcare in Pakistan and Botswana. It can be expected that new applications of spoken language technology for development (SLT4D) will be developed in future.

This article aims to develop a framework for collecting and codifying the necessary linguistic information for SRSs in the computer environment. To date, studies have been done for some languages such as English, and the experience acquired in these endeavours can be applied to the languages that are resource scarce in this sense. However, these studies have been performed without a formal, agreed methodology or framework. Given that there are about 7 000 languages in the world, almost all of which are resource scarce, such a framework can be important for the penetration of computer technology in developing areas of the world.

The remainder of this article is structured as follows: The next section explains the characteristics of the SRS, followed by the sections that deal with the development of the basic components of an SRS such as an acoustic model, a pronunciation dictionary, a user interface and a language model. Next, open-source tools for developing these components and a framework for building a simple SRS are discussed. The article ends with the conclusion section.

**Basic aspects of the SRS**

Two forms of speech technology are particularly relevant for the developing world, namely automatic speech recognition (ASR) and text-to-speech (TTS) systems. This article focuses on ASR systems, and although a similar analysis of TTS systems would be very worthwhile, it falls outside the scope of this discussion.

The main difficulty of ASR is the sheer number of the languages spoken in the world. Each language has its own phonological system of words and morphemes with their corresponding written symbols. Morphemes are the smallest grammatical units that make up words in a language. The syntactic systems that govern how words and morphemes form phrases and utterances also differ from language to language (Language, n.d.). Although there may be similarities among the languages in a particular language family, an ASR system has to tackle all the individual characteristics of a language.

State-of-the-art ASR systems implement statistical pattern recognition methods to identify individual words or larger spoken units. Hidden Markov Models are the most common formalism used for this process, and a tremendous amount has been learnt about the practicalities of achieving good speech-recognition accuracy within this formalism. Current systems are still significantly inferior to human capabilities, but are sufficiently accurate to be used on a daily basis by millions of users in the developed world.

In order to develop a working ASR system for a particular language, three key components have to be developed, namely acoustic models, language models and pronunciation dictionaries. An acoustic model is the statistical representation of individual sounds that make up words in a language. The first step in building an acoustic model is to develop a speech corpus by systematically recording the spoken language. The next step involves identifying individual sounds in the corpus and determining the statistical representations of the sounds that make up the words. In the recognition process these statistical representations are compared with the phonemes, and the "fittest" words are estimated. Since this process involves approximation rather than mathematical accuracy, ASR systems have varying degrees of error rates in identifying the words. The error rates are usually inversely proportional to the size of the speech corpus. For most purposes, the acoustic model needs to be speaker-independent due to the large number of potential users (although some training of such a speaker-independent system for a particular user or group of users may be feasible in certain applications).

The language model also involves prediction based on probability distribution. In contrast to the acoustic model, which involves prediction of likely phonemes from the speech sounds received as input, the language model involves prediction of the sequences of words that are likely to occur.

Thus the language model is primarily in the domain of words, the acoustic model is primarily concerned with the acoustics of phonemes, and the pronunciation dictionary relates these two domains to one another. In other words, the pronunciation dictionary describes the sequence of phonemes that represent the pronunciation of each of the words that may occur in the language model.

Taken together, these elements constitute the core of an ASR system. However, to be practically useful, this core must be deployed in an application environment, and a user interface must be developed that allows the relevant user population to access that environment. The properties of this user interface are of vital importance, since it must compensate for the limitations of the technology components and take into account the characteristics of the expected user.

Figure 1 shows the relationships between the various components. We next discuss each of the ASR components in more detail.

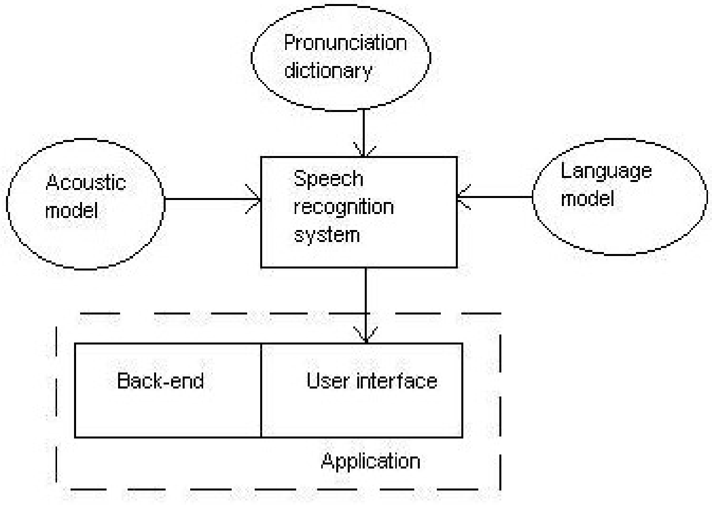

*Figure 1. Block diagram of a speech-enabled application*

**Acoustic model**

As mentioned above, state-of-the-art ASR systems employ Hidden Markov Models for acoustic modelling. These models are well-defined statistical entities, and are therefore not specific to any particular language. The language-specific nature of acoustic models is a result of the *training* which is required to develop a model for a specific language – such training allows the model to capture the acoustic properties of a target language, and is based on a *transcribed speech corpus* in that language. Thus, the first step in creating the acoustic model is to develop a speech corpus in which the speech of the native speakers of the language is recorded. Below we briefly describe the process of corpus development, and in the section on Tools we mention how such a corpus is used for training.

### *Developing a speech corpus*

Speech corpus development starts with the recording of spoken utterances by a number of speakers who are representative of the target population that will be using the system. The recorded raw speech is generally not directly suitable for the purpose of ASR, and it needs to be curated and transcribed. The process of speech recording in a systematic way requires a working organisation as well as a suitable infrastructure comprising, for example, suitable recording devices, networks of people who can be canvassed for recording purposes, etc. The process of curation and transcription of the recorded speech requires speech scientists who are knowledgeable in this area. This knowledge includes the theoretical background of the processes involved as well as familiarity with the software tools used in these processes. Although not absolutely necessary, it would be desirable for these scientists also to have a good command of the language involved. Barnard et al. (2009) argue that the number of speakers who need to be canvassed and the amount of speech are the factors that increase the complexity of the speech corpus. In developing country settings this complexity needs to be reduced and an acceptable level of error rate in speech recognition must be achieved. Since these are conflicting demands, a balance must be sought between them. The authors report that the Lwazi speech corpus (Meraka Institute, 2009), which has been developed for the eleven official languages in South Africa, has 200 speakers per language whose speech was recorded over a telephone channel. According to the authors, fewer than 50 speakers and 10 to 20 hours of speech were sufficient for acceptable phone-based recognition.

If there is a large corpus, then some representative subset of speech data must be determined. This is due to the fact that random data are not suitable for achieving the optimum recognition in speech corpus design. The aim of determining this subset is to increase the variability as much as possible while designing the corpus specifically for the intended operating environment (ibid.). The subset can be determined by supervised and unsupervised learning methods (Riccardi & Hakkani-Tur, 2003).

Due to technical and economic difficulties in developing countries, the corpus usually cannot be very large, which means that the speech resources must be efficiently designed. Wheatly et al. (1994) attempted to estimate the number of speakers and the amount of speech for each speaker needed to develop a corpus for Japanese. The authors found that the number of speakers is more important than the amount of speech for each speaker. Barnard et al. (2010) argue that access to useful information in developing areas does not necessarily require large vocabularies, and that as few as a dozen words can be satisfactory for a working system with acceptable error rates of 2% to 12% (van Heerden et al., 2009). Such a modest start can later pave the way to more sophisticated systems.

### Language model

Prior to the 1980s, ASR systems did not generally use explicit language models. The view then was that incorporating the syntactic and semantic rules of the language into the ASR system greatly enhanced its accuracy (O'Shaughnessy, 2008).

The majority of ASR research has been on large datasets implemented in larger speech and text corpora, but these systems are usually not very meaningful in the settings of disadvantaged areas.

This implies that, like the size of the speech corpus, a small text corpus is often the only realistic option in resource-constrained areas.

Language models can employ statistical or rule-based methods. For large systems statistical models such as n-grams can be used. These models assess the likelihood of word sequences according to language usage and accordingly inject the right bias in an ASR to prefer an output sequence of words with high probability according to the language model (Zhai, 2008). The development of the language model starts with compiling the text corpus. The next step is to extract the vocabulary from this corpus and, depending on the nature of the language,  further processing is carried out, such as text normalization or morpheme segmentation. For morphologically complex languages, this segmentation is typically processed by the Viterbi algorithm which finds the most probable segmentation of a word in view of the morpheme lexicon. Next, n-grams are trained to establish the language model (Hirsimaki et al., 2006). Since it is unfeasible and difficult to collect an all-encompassing text corpus, a small amount of domain-specific data can be sufficient for the language model in some resource-constrained SLT4D applications (Bellegarda, 2004).

Alternatively, rule-based language models may be employed. Smaller systems which are more relevant for SLT4D involve many fewer rules than the large systems. In such cases, a rule-based language model with a set of simple rules may be quite sufficient (Pisarn & Theeramunkong, 2008).

Like the acoustic model, the process of developing the language model also requires linguistic knowledge about the relevant language. One role of such knowledge is to develop corpora for the language model if it is not readily available. Unlike the acoustic model, this knowledge can be retrieved from text-based sources such as printed material or electronic texts. It is therefore easier to obtain the text corpora due to the availability of such material for many languages. The text corpora must be analysed for aspects such as stemming and morphology by the linguists.

There are some severely under-resourced languages that have no writing systems that can be used for this purpose. They constitute an overwhelming majority of the world's languages (Nettle & Romaine, 2000). As it is difficult to develop the language model for these languages, it is probably not feasible to develop ASR systems for them.

## Pronunciation dictionaries

A crucial component of speech recognition systems (and similarly of text-to-speech systems) is the *pronunciation dictionary* (or lexicon), which maps between the spoken and written forms of a language. Most current systems employ a *phonetic* representation of speech – that is, speech is represented as an ordered sequence of phonemes. The goal of the pronunciation dictionary is to represent the relationship between this sequence of phonemes and the sequence of symbols that constitute the written form of the language. This relationship is, of course, highly language dependent: both in structure and in details, different languages have adopted writing systems that reflect the particular history of the language. Compare, for example, Chinese characters, which represent syllables or larger units of speech, with the Latin characters of this text, which are closely (but not perfectly) related to the phonemes themselves. The requirements for dictionary development depend greatly on factors such as these, and two broad families of approaches are currently predominant.

1. In *manual dictionary development*, a phonetician or other person knowledgeable about the target language captures the phonetic equivalent of each relevant written (orthographic) unit (which could be a word, character, syllable, etc.)
2. *Rule-based dictionary development* can be employed when there are regularities between the pronunciation and spelling systems that can be captured algorithmically. These rules can either be written down manually by a human expert or derived automatically by a *learning algorithm*, based on a set of manually produced pronunciations.

Many practical complications need to be considered during dictionary development, including dialectal differences, heteronyms (words with the same spelling but different pronunciations depending on context, such as the present and past tense of *read* in English), pronunciation variants (words that may be pronounced differently depending on speaker or context), etc. Even when pronunciation dictionaries (or rules) exist in a target language, some of these factors may not be accounted for (since they may not be important for tasks other than speech recognition), thus requiring more or less adaptation before such pre-existing resources can be used in practice. Another issue to be considered in practice is that loan words, proper names and other items may not be present in a given dictionary – thus rules for dealing with such entries are often required in addition to the primary pronunciation dictionary.

Dictionary development can sometimes be a highly labour-intensive task – for a language such as English, with an irregular spelling system, several person-years may be invested in developing a general-purpose dictionary. However, many resource-scarce languages have more regular spelling systems (as a consequence of the more recent development of the systems) – for such languages, high-quality dictionaries can be developed in weeks with appropriate tools as discussed in the section on tools below.

**User interfaces**

A wide range of user interfaces are used in conjunction with speech-recognition systems, depending on the nature of the application. Four broad classes of applications can be distinguished (although the distinction is not always unambiguous):

1. *Applications on personal computers* such as dictation, command-and-control and data-entry applications.
2. *Telephone-based applications*, such as self-help services (telephone banking, travel reservations, etc.) and information portals.

3. *Embedded applications*, which include voice diallers on personal telephones, control interfaces for industrial devices, etc.
4. *Kiosk-based applications*, which allow users to obtain information from dedicated kiosks through interfaces which may include touch screens, spoken commands and keyboards.

Within each category of applications, a wide range of interface options exist. Consider, for example, telephone-based applications. Historically, most of these applications have employed a menu-based structure that prompts the user for a particular input, processes that input and then responds with information, actions or additional prompts (depending on the context and the user's request). As a consequence, the most widely used standard for telephone-based speech interfaces (VoiceXML) is primarily geared towards the development of such menu-based applications. However, more recently, search-based interfaces such as voice search (Erol et al., 2009) have become popular alternatives to menu-based structures.

Different interface choices are likely to place significantly different sets of requirements on both users and the technology components. For example, a kiosk which can use both speech and touch screens for input and output is likely to accommodate illiterate users more easily than a menu-based system, but requires that users have access to appropriate hardware and also that the relevant multimodal software interface be developed.

Resource-scarce languages are often (although certainly not exclusively) used by populations in which literacy or exposure to advanced technology is less common. Consequently, the design of user interfaces for such populations is a topic that is closely related to the development of speech-recognition systems for under-resourced languages. Although some initial work on this topic has been published (Patel, 2010), it is fair to say that speech user-interface design for illiterate or technologically unsophisticated users is still a field in its infancy.

**Tools**

There is a wide variety of open-source tools on the Internet for developing SRS applications. These tools are almost exclusively free-of-charge downloadable under a variety of licence conditions, satisfying one of the most important requirements for SLT4D applications, namely affordability. Many of them come with reasonably satisfactory tutorials and documentation. Additionally, they usually have Internet forums and mailing lists with people eager to help in case of a problem.

Although Linux-Unix is the "native" environment for most of the tools, many of them can also run on MS Windows operating systems with Linux emulators such as Cygwin (www.cygwin.com) which can also be freely downloaded. It must, however, be stated that there are some limits to the compatibility of Cygwin with Linux.

### *ASR tools*

*HTK* (Hidden Markov Model Toolkit) is an ASR tool that has been developed at Cambridge University (htk.eng.cam.ac.uk) and it has been affiliated with Microsoft. Although its main usage area is ASR, it can also be used in some other areas where HMM can be used, such as DNA sequencing. The system can be downloaded free after registration with a valid e-mail address. The website contains a user manual for advanced users and a tutorial for beginners.

An HMM-based ASR decoder is *Julius,* which has been developed by a consortium in Japan (http://julius.sourceforge.jp/en_index.php). Julius uses the acoustic models and pronunciation dictionaries that have been developed in HTK format. The role of the decoder is to perform run-time speech recognition given trained acoustic models (as well as the relevant language models and pronunciation dictionaries) – hence Julius complements HTK (which has more restricted run-time capabilities) well.

*CMU Sphinx* is another ASR toolkit that has been developed at Carnegie Mellon University and is downloadable from the Sourceforge website (http://cmusphinx.sourceforge.net/). The system has been developed by C, Perl, Java and Python, and is suitable both for developing simple applications as well as speaker-independent, large-vocabulary continuous speech recognisers. It also has a wide variety of ASR tools and resources for developers and researchers.

### *Data Collection*

*Woefzela* is a speech data collection tool for SLT4D applications that has been developed in South Africa (De Vries et al., 2011). The system runs on mobile devices with the Android operating system. The tool has been designed for use in typical developing country conditions and contains a semi-real-time quality monitoring system which reduces errors.

### *Pronunciation Dictionary*

*DictionaryMaker* is an electronic pronunciation dictionary that consists of a list of words, each associated with one or more phonetic pronunciations (http://dictionarymaker.sourceforge.net/) (Davel & Barnard, 2003). The system does not require the knowledge of an expert linguist nor any programming skills, and it creates a related set of grapheme-to-phoneme (g-to-p) rules automatically. It is suitable for languages such as English, Turkish or Zulu – that is, languages in which the writing symbols (letters) roughly correspond to phonemes.

### *Interfaces*

*VoiceXML* is the World Wide Web Consortium (W3C) standard for voice interaction between humans and computers. It is the voice equivalent of the HTML and XML standards for visual applications. It requires a voice browser just as the visual applications require visual web browsers, and is primarily designed for interactions in the form of dialogues (such as those that are used in telephone-based systems). Thus most of the interface-development tools that are suitable for such interactions employ the VoiceXML standard. Such tools include *OpenVXI*, which is an open-source VoiceXML interpreter, and *VoiceGlue*, which integrates OpenVXI on the popular open-source telephone platform *Asterisk.*

## The framework

To put the above discussion into perspective, a typical ASR development process is graphically depicted in Figure. 2. As can be seen, the process starts with the definition of the basic linguistic elements of the target language, followed by the collection of a text corpus. (The size of the text corpus depends on the language and the application at hand – for typical developing-world applications, corpora of a few hundred thousand word tokens may be sufficient, whereas sophisticated statistical language models require at least a hundred-fold more data.) This corpus is used in three ways: for the extraction of common words that are used for pronunciation modelling, for the selection of prompting materials that are read by speakers for ASR corpus collection and for the development of language models. (For statistical language models, the statistics of the corpus are modelled directly; for rule-based language models, the corpus is used as guidance for the creation of appropriate rules.)

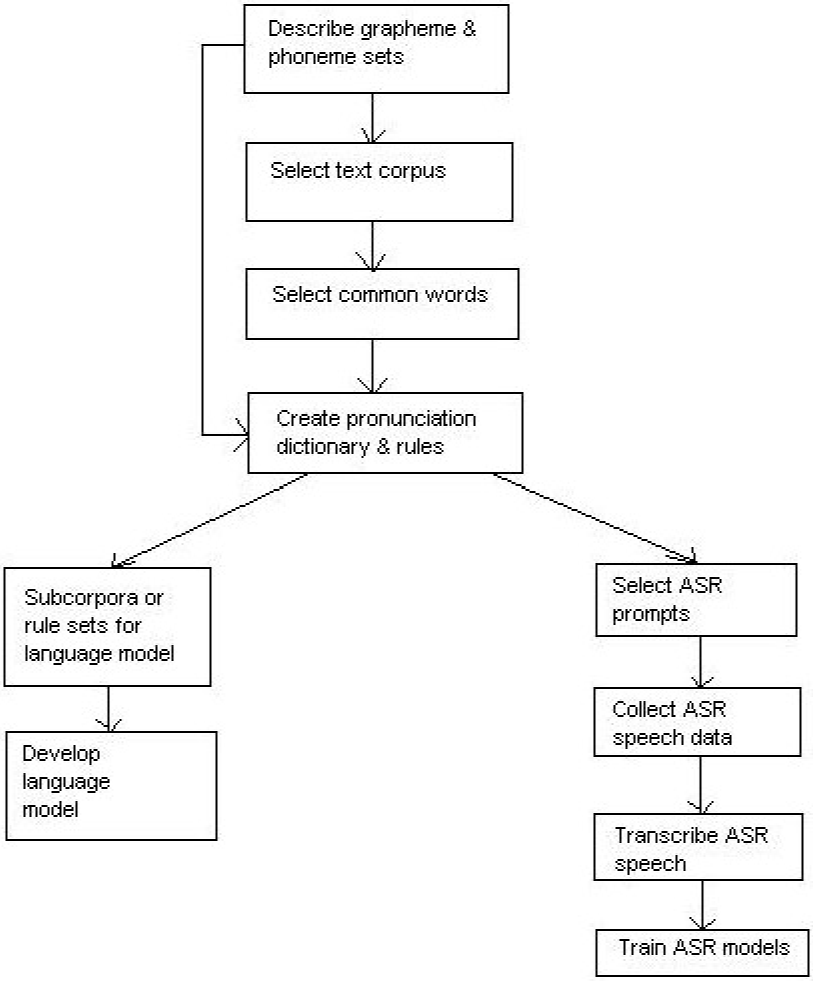

*Figure 2. Typical ASR development process*

Once these materials have been collected, the process of developing each of the three ASR components is mostly guided by the tools that are employed. Most of the tools described above have been used for the processing of a wide range of languages, and can therefore be efficiently adapted to new languages.

## Conclusion

There is a growing literature on ICT applications in under-resourced areas. A significant part of this literature deals with seamless and cheap connection of remote and disadvantaged areas to knowledge sources. The knowledge can be expert advice on a specific topic, daily crop prices or the weather forecast.

ICT4D applications may or may not involve computers. In an overwhelming majority of the applications where computers are involved, the interaction between humans and computers is performed in traditional ways, i.e. visual browsers are used. ASR systems offer an alternative to visual browsing, which requires a certain degree of literacy and computer knowledge. This alternative involves using speech which is the "native" communication tool of humans. ASR systems are commonly used in more developed areas of the world and substantial knowledge has been accumulated for developing them for the languages used in disadvantaged areas. The potential benefits of using them can compensate for the relative difficulty of their development process.

## References

*Barnard, E., Davel, M. & Van Heerden, C. (2009).* ASR corpus design for resource-scarce languages: *Proceedings of Interspeech, Brighton, UK, 2847-2850.*

Barnard, E., Davel, M. H. & Van Huyssteen, G. B. (2010). Speech technology for information access: a South African case study. *Proceedings of the AAAI Spring Symposium on Artificial Intelligence for Development (AI-D)*, Palo Alto, California, 8-13.

Bellegarda, J. (2004). Statistical language model adaptation: review and perspectives. *Speech Communication,* 42(1), 93-108.

Benzeghiba, M., De Mori, R., Deroo, O., Dupont, S., Erbes T., Jouvet, D., Fissore, L., Laface, P., Mertins, A., Ris, C., Rose, R, Tyagi, V. & Wellekens, C. (2007). Automatic speech recognition and speech variability: A review. *Speech Communication*, 49(10-11), 763–786.

Boretos, G. P. (2007). The future of the mobile phone business. *Technological Forecasting and Social Change,* 74(3), 331-340.

Davel, M. & Barnard, E. (2003). Bootstrapping in language resource generation. *Proceedings of the 13th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, Langebaan, South Africa, 97-100.

De Vries, N., Badenhorst, J., Davel, M., Barnard, E. & De Waal, A. (2011). Woefzela - An open-source platform for ASR data collection in the developing world: *Proceedings of Interspeech, Florence, Italy.*

Erol, B., Cohen, J., Etoh, M., Hon, H., Luo, J. & Schalkwyk, J. (2009). Mobile media search: *ICASSP '09: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing,* Washington, DC, USA, 4897–4900.

Hirsimaki, T., Creutz, M., Siivola, V., Kurimo, M., Virpioja, S. & Pylkkonen, J. (2006). Unlimited vocabulary speech recognition with morph language models applied to Finnish. *Computer Speech and Language*, 20(4), 515–541.

Language. (n.d.). Retrieved 12 June 2011 from Wikipedia: http://en.wikipedia.org/wiki/Language

Meraka Institute, CSIR. (2009). Lwazi ASR corpus. Retrieved 14 June 2011 from: http://www.meraka.org.za/lwazi

Metcalfe, M. (2007). Development and oral technologies. *Information Technology for Development*, 13(2), 199–204.

Nasfors, P. (2007). *Efficient voice information services for developing countries*. Masters thesis, Department of Information Technology, Uppsala University.

Nettle, D. & Romaine, S. (2000). *Vanishing Voices*. New York, NY: Oxford University Press Inc.

O'Shaughnessy, D. (2008). Invited paper: Automatic speech recognition: History, methods and challenges. *Pattern Recognition*, 41(10), 2965–2979.

Parikh, T. S. & Lazowska, E. D. (2006). Designing an architecture for delivering mobile information services to the rural developing world: *Proceedings of the International Conference on the World Wide Web (WWW),* Edinburgh, Scotland, 791–800.

Pisarn, C. & Theeramunkong, T. (2008). Thai spelling analysis for automatic spelling speech recognition. *Information Sciences*, 178(1), 122–136.

Patel, N., Chittamuru, D., Jain, A., Dave, P. & Parikh, T. S. (2010). Avaaj Otalo – A field study of an interactive voice forum for small farmers in rural India: *Proceedings of CHI2010,* Atlanta, Georgia, 733-742.

Riccardi, G. & Hakkani-Tur, D. (2003) Active and unsupervised learning for automatic speech recognition: *Proceedings of Eurospeech*, Geneva, Switzerland, 1825–1828.

Sharma, A., Plauche, M., Kuun, C. & Barnard, E. (2009). HIV health information access using spoken dialogue systems: Touchtone vs. speech: *Proceedings of the IEEE International Conference on ICTD,* Doha, Qatar, 95–107.

Sherwani, J., Palijo, S., Mirza, S., Ahmed, T., Ali, N. & Rosenfeld, R. (2009). Speech vs. touch-tone: Telephony interfaces for information access by low literate users: *Proceedings of the IEEE International Conference on ICTD,* Doha, Qatar, 447–457.

Van Heerden, C., Barnard, E. & Davel, M. (2009). Basic speech recognition for spoken dialogues: *Proceedings of Interspeech*, Brighton, UK, 3003–3006.

Wheatley, B., Kondo, K., Anderson, W. & Muthusumy, Y. (1994). An evaluation of cross-language adaptation for rapid HMM development in a new language: *Proceedings of ICASSP,* Adelaide, Australia, 237–240.

Zhai, C. (2008). Statistical language models for information retrieval: a critical review. *Foundations and Trends in Information Retrieval,* 2(3), 137–213.