A smartphone-based ASR data collection tool for under-resourced languages

Nic J. de Vries\textsuperscript{a,b}, Marelie H. Davel\textsuperscript{a,c}, Jaco Badenhorst\textsuperscript{a,b}, Willem D. Basson\textsuperscript{a,b}, Febe de Wet\textsuperscript{a,c}, Etienne Barnard\textsuperscript{b}, Alta de Waal\textsuperscript{a}

\textsuperscript{a}Human Language Technologies Research Group, Meraka Institute, CSIR, Pretoria, South Africa
\textsuperscript{b}Multilingual Speech Technologies, North-West University, Vanderbijlpark 1900, South Africa
\textsuperscript{c}Department of Electrical and Electronic Engineering, Stellenbosch University, South Africa

Abstract

Acoustic data collection for automatic speech recognition (ASR) purposes is a particularly challenging task when working with under-resourced languages, many of which are found in the developing world. We provide a brief overview of related data collection strategies, highlighting some of the salient issues pertaining to collecting ASR data for under-resourced languages. We then describe the development of a smartphone-based data collection tool, \textit{Woefzela}, which is designed to function in a developing world context. Specifically, this tool is designed to function without any Internet connectivity, while remaining portable and allowing for the collection of multiple sessions in parallel; it also simplifies the data collection process by providing process support to various role players during the data collection process, and performs on-device quality control in order to maximise the use of recording opportunities.

The use of the tool is demonstrated as part of a South African data collection project, during which almost 800 hours of ASR data was collected, often in remote, rural areas, and subsequently used to successfully build acoustic models for eleven languages. The on-device quality control mechanism (referred to as QC-on-the-go) is an interesting aspect of the Woefzela tool and we discuss this functionality in more detail. We experiment with different uses of quality control information, and evaluate the impact of these on ASR accuracy. Woefzela was developed for the Android Operating System and is freely available for use on Android smartphones.

Keywords: smartphone-based, ASR data collection, under-resourced languages, automatic speech recognition, ASR corpora, speech resources, speech data collection, broadband speech corpora, Woefzela, on-device quality control, QC-on-the-go, Android

1. Introduction

With respect to developing human-machine interfaces, science and technology have come a long way in the past few decades, with varying degrees of success (Huang et al., 2001; Barnard et al., 2010b). In the developed world, speech technology has a well established track record of usefulness, with applications such as call routing, directory services, dictation and travel information considered mainstream applications. These applications generate annual revenues in the billions of dollars for some companies and are saving large companies and non-profit organisations alike significant amounts of money (Barnard et al., 2010b).

With the rapid increase in the number of mobile phones worldwide, the input modality of speech is becoming increasingly important. Speaking is much faster and more natural than keyboard entry, especially for languages such as Cantonese and Japanese with large character sets (Sung et al., 2011; Brewer et al., 2005; Kamvar and Beeferman, 2010; Schuster, 2010; Shan et al., 2010). For human-robotic interfaces, speech input will become a necessity as the tasks that robots can perform become increasingly complex.

In the developing world, this picture is in one way similar and yet in another, significantly different. With more recent applications in domains such as health information services (Sherwani et al., 2007; Grover et al., 2009), education (Brewer et al., 2005), information access (Grover and Barnard, 2011), agriculture (Sherwani et al., 2007) and government services (Barnard et al., 2003), speech technologies are beginning to demonstrate the impact that they could have in these environments, for example, breaking down barriers of inequality of information accessibility (Lata and Chandra, 2010) and generating revenue for future economic sustainability (Brewer et al., 2005).

However, for automatic speech recognition (ASR) technology to impact the developing world more significantly, a number of hurdles must first be overcome. One of these hurdles is the collection or expansion of ASR corpora (Grover et al., 2011; Barnard et al., 2010a, 2008; De Wet et al., 2006). Currently, only about 30 of the world’s 6,900 languages have significant quantities of digitised data, and much of this data is only in textual form (Abney and Bird, 2010; Barnard et al., 2010b; Maxwell and Hughes, 2006).

In order to maximise the data that is already available, much progress has been made with regard to using limited speech data efficiently. For example, techniques such as language adaptive acoustic modelling as discussed in Schultz et al. (Schultz and Kirchhoff, 2006) may be used when no or limited data of related languages exist, or when language independent models...
are available. Cross-lingual acoustic modelling techniques include cross-language transfer (i.e. no training data used for target language), language adaptation (i.e. limited target language training data used for adapting acoustic models), bootstrapping (i.e. initialising acoustic models from a different language) and data pooling (i.e. directly combining data from different languages) (Wheatley et al., 1994; Constantinescu and Chollet, 1997; Ackerman et al., 1996; Schultz and Waibel, 2001; Van Heerden et al., 2010).

However, even when all available data is used as well as possible, it is often found in practice that a point is reached where the only alternative is collecting additional, or at least some well-matched language-specific or domain-specific data. For under-resourced languages, it is particularly important that this should be done in a cost effective way (Hughes et al., 2010; Lane et al., 2010; Sung et al., 2011). Collecting ASR data for any language is no trivial task. While great progress has been made towards collecting ASR data cheaply and efficiently in general, some potential barriers for developing world contexts still remained to be addressed (Hughes et al., 2010; Lane et al., 2010).

Our goal here is to describe the need for and development of a data collection tool, called Woefela (De Vries et al., 2011), which is especially applicable in the developing world context where most under-resourced languages are commonly spoken, and to evaluate some of the functional components of this tool. In particular, we will evaluate its quality control (QC) system for usefulness and impact on ASR system performance.

This study was conducted in a South African context with the immediate impetus for this work provided by a project from the National Centre for Human Language Technology (NCHLT) of South Africa, funded by the South African Department of Arts and Culture. This project sought to collect around 50 hours of broadband speech data for each of the eleven official languages of the country, geographically spanning six of the nine provinces.

2. ASR data collection strategies and trends

Over the past few decades various strategies and processes for collecting ASR data have become well established, while other exciting new trends indicate a number of emerging strategies. In this section we provide a brief overview of a selection of these strategies, highlighting some of the salient issues related to collecting ASR data for under-resourced languages. For a more expanded version, refer to De Vries (2012).

The complete corpus creation process from corpus design or specification, prompt text selection, through validation to final documentation and distribution is not considered here, but the focus instead is on the actual audio collection stage. For a broader view of the subject and other aspects of this process, excellent cookbooks, for example Schiel and Draxler (2003), and other resources (Schultz and Kirchhoff, 2006; Badenhorst et al., 2011b; Barnard et al., 2009; Van Den Heuvel et al., 2008; De Wet et al., 2006; Botha and Barnard, 2005) are available.

2.1. Established strategies

Professional studio environments have the advantage that environmental noise and disturbances can be more easily controlled, leading to reduced unwanted acoustic events and thus higher quality speech data. However, this quality comes at a price, and such data is generally not well-matched with the speech application environment. The logistics of transporting all speakers to a fixed location in order to collect voice data combined with the serial nature of recording only one speaker at a time, is prohibitive in many contexts. Although temporary studio environments alleviate some of these drawbacks, they introduce other potential problems such as higher noise levels caused by the reduced quality of insulation to external noises compared to professional studio environments. Despite these drawbacks, this is still a viable alternative for some speech data collection projects such as the GlobalPhone corpus (Schultz, 2002).

Telephone-based recordings are often closely matched with the intended use of speech data, especially when spoken dialogue systems are to be deployed on such networks. With telephone networks, including mobile phone networks growing rapidly in developing world regions (Barnard et al., 2010a), telephone-based ASR data collection is sometimes a viable, if not preferred alternative over studio based data collection. However, bandwidth limitations, a lack of control over the speaker’s environment, handset noise, effective user screening (for example, first language reading and speaking ability) and user identity verification (to avoid duplication of users calling into the system), are highlighted by De Wet et al. (2006). Examples of corpora for under-resourced languages collected in this way are the Lwazi corpus (Barnard et al., 2009) and the AST corpus (Roux et al., 2004).

Collecting additional ASR data on a live recognition service such as Google’s Voice Search service (Barnard et al., 2010b), is a well established and cost-effective means of extending the amount of available speech data through a continual process of live-service collection. Although the quality of such data needs to be verified prior to employing it in adapting or retraining acoustic models (to avoid deterioration of recogniser performance), this is a powerful strategy, and a technique often utilised by commercial ASR systems. However, where insufficient ASR data is available to train the initial acoustic models, or when no live service to deploy such initial models on exists, this alternative is simply not available. To our knowledge, the only live service currently available for African languages is Google’s Voice search service (Barnard et al., 2010b).

2.2. Emerging strategies

In recent years, various web-based speech data collection strategies built around Internet infrastructure have emerged, which has many advantages over other data collection strategies (McGraw et al., 2010; Draxler, 2007). With the rapid growth of crowdsourcing approaches for various Human Intelligence Tasks (HITs) in the domains of natural language processing and other human language technology tasks such as machine translation, several crowdsourcing corpus creation strategies have emerged (Parent and Eskenazi, 2011). For example,
in a recent study the authors used Amazon Mechanical Turk for both collecting speech data from users as well as transcribing this data with HITs (McGraw et al., 2010).

Inherent in this approach, however, is the variability of recording equipment attached to on-line computers, the acoustic environments in which the participants choose to record the speech and other potential drawbacks similar to that of telephone-based collection (apart from the band-width limitation). Nevertheless, in the developed world (and some parts of the developing world) where Internet access is readily available, this approach has definite advantages.

A further strategy emerging in recent years is that of data harvesting from existing sources, often using lightly or unsupervised learning (Gauvain et al., 1998; Lamel et al., 2002). Harvesting audio data with associated “approximate transcriptions” from on-line sources provides opportunities especially valuable for under-resourced languages to build corpora in a cost-effective way. Other sources such as broadcast news (Kamper et al., 2012; De Wet et al., 2011; Davel et al., 2011; Kim et al., 2003) or lecture notes are also mined for such audio data, each presenting its own unique challenges (Lee and Glass, 2011).

3. Smartphone-based ASR data collection

Another particularly promising emerging data-collection strategy is that of using smartphones as general purpose recording devices to collect speech data. With smartphones becoming increasingly available and sharply decreasing in cost, even in the developing world, viewing smartphones as general purpose recording devices is rapidly changing the face of ASR data collection – especially for under-resourced languages. Dynamically selecting and presenting previously compiled prompts to be recorded on the mobile screen, recording the actual speech utterances at broadband quality (typically to the SD card), and collecting meta data associated with each prompt, has become feasible with these mobile computers.

This trend was led by Google’s smartphone application (Hughes et al., 2010) which was first used in a large-scale data collection campaign to collect speech data cheaply and efficiently for their Voice Search project (Barnard et al., 2010b; Sung et al., 2011).

A similar approach was taken by Lane et al. (2010) around the same time, also offering the functionality of downloading a list of previously compiled prompt texts to be recorded to the smartphone, recording the utterances in an off-line manner on the device and subsequently uploading the resulting audio to a server for further processing.

A potential drawback of these approaches is the fact that the only speaking style appropriate for such collection is that of read speech. However, this data has been found to be extremely valuable as initial and final application corpora for ASR system development.

Inspired by the Google approach, an open-source application called Woefzela was developed, specifically geared towards collecting speech data for under-resourced languages in the developing world.

Woefzela also includes an on-device quality control system for identifying gross recording errors, and prompts the respondent accordingly to record additional speech data, thereby achieving a higher per-speaker number of good recordings and thus maximising the opportunity for collecting speech data from each speaker (De Vries et al., 2011).

In De Vries (2012), data collected with Woefzela was shown to give good ASR system performance (74.23% phone accuracy with 54 hours of training data) without any further validation or selection of the data apart from the on-device quality control system. Badenhorst et al. (2012) further evaluated the effectiveness of this basic quality control mechanism as well as proposing additional quality control criteria for on-device execution. The basic quality criteria were shown to be useful and the proposed additional criteria seemed to indicate promising potential.

Not strictly dependent on the smartphone-based approach, but strongly coupled to it, is the subsequent use of these prompt texts as “approximate” orthographic transcriptions of the recorded speech data. This prompted further research into verifying the quality of smartphone-based collected ASR data, developing techniques for automatically validating and selecting subsets of data for ASR acoustic model training (Davel et al., 2012). Thus, with the increasing availability of smartphones, the overall usefulness of this approach has already enabled a number of projects involving under-resourced languages (Modipa et al., 2012; Basson and Davel, 2012; Giwa et al., 2011; Badenhorst et al., 2011a; Van Heerden et al., 2011; Kleynhans et al., 2012; Van Heerden et al., 2012).

In designing such smartphone-based ASR data collection tools for under-resourced languages, particularly against the backdrop of the developing world, the following aspects needed to be carefully considered.

3.1. Design considerations

Given the opportunity to develop an open-source platform for data collection in the developing world, a number of questions could be asked: Which general design requirements should such a tool address? Which requirements are indeed unique to developing world contexts that should shape the design of such a tool? A few are addressed here as a precursor for the later specification of Woefzela.

3.1.1. Portability

Equipment portability is a primary requirement for effective collection of ASR data in developing world areas. Mother-tongue speakers of under-resourced languages often reside either in remote rural areas, or in small communities of speakers distributed over large geographic areas. In conducting data collection campaigns for these languages, transporting people to stationary recording environments or bulky equipment to remote locations, is often unfeasible in terms of cost and may be risky due to the high potential of damage to the equipment or the safety of passengers transported on rural roads (Breuer et al., 2006).
3.1.2. Parallelisation

In order to maximise the opportunity when travelling long distances to remote locations to access small communities of mother-tongue speakers, the ability to collect speech data from a number of speakers in parallel is key. By employing cost effective and highly portable equipment for such campaigns, more than one speaker’s audio data may be recorded at the same time (or in more than one geographic location where mother-tongue speakers are distributed over large geographic areas). The primary upper limit of parallelisation in this regard would be specific equipment budget constraints and any associated manpower constraints.

In contrast, renting professional or semi-professional studio time or constructing such studios, only allow for very limited parallelisation. Compounded by the dynamic recruitment nature typical of under-resourced language data collection campaigns, these fixed-location approaches become impractical.

3.1.3. Internet independence

If wired or wireless Internet connectivity was assumed, a number of opportunities would be available for ASR data collection, e.g., downloading textual prompts for recording, uploading recorded data to a central server and even performing some form of quality control of the audio data on back-end servers in time to give feedback to the speaker during the same session. Unfortunately, such an assumption is not yet valid for the vast majority of developing world regions (Brewer et al., 2005; Pentland et al., 2004). While some of these regions have cheap, reliable Internet connectivity, this is not generally the case for most developing regions. Such connectivity may be non-existent, highly congested or provided on an ad-hoc basis. For example, the DakNet project discussed in Pentland et al. (2004) uses a wireless router mounted on top of a bus to “transport” email between villages and an Internet connection in a nearby city and thus to the rest of the world. This bus is effectively acting as a “digital postman” collecting and delivering “mail”.

In South Africa, wireless connectivity to the Internet, while generally available in large parts of the country, can unfortunately not yet be assumed in some of the more rural areas. Even if connectivity existed in some of these areas, the costs of accessing such services are often prohibitive. In the bigger cities some open wireless access points are present, but with the limited bandwidth typically available through these access points, field workers have spent many hours in uploading data collected for similar data collection campaigns. When private access points were used, field workers had to spend large amounts of money for uploading the data recorded.

Tools that were primarily aimed at developed-world environments (Hughes et al., 2010; Lane et al., 2010) did not consider this limitation: at some stage during the data collection process, an Internet connection was required to complete the task. The alternative which we pursue is to record the data directly onto the SD card, and to retrieve the data through physical card exchange (De Vries et al., 2011; De Vries, 2012). In conclusion, in the instances that Internet connectivity is available in these developing regions, cost, throughput, latency and stability may be prohibitive factors for large scale data collection campaigns, especially on limited budgets.

3.1.4. Open-source software

One of the best arguments for the use of open-source software for ASR data collection, is the flexibility and opportunity for customisation that it provides for diverse contexts (Brewer et al., 2005). For ASR data collection of under-resourced languages, this is a particularly important aspect as the unique needs of different languages, locales and recording campaigns are simply too diverse to be envisaged a priori for all contexts. Also, in projects with highly constrained budgets, free or low cost software may not only provide the necessary impetus for ASR data collection projects, but may also be the only means of completing such projects within budget requirements.

3.1.5. Maximising recording opportunity

When first-language speakers need to be reached in remote areas or when such speakers are sparsely distributed over wide geographic areas, recording opportunity is of prime importance. Once a respondent is engaged in a recording process, maximum usable data must be obtained during a session because it may not be feasible to procure the services of the same person (or another first language speaker in the same area) due to cost or other practical constraints.

3.1.6. Providing support for field work

The process of simultaneously assisting a number of respondents to donate speech data can be very exhausting, potentially impacting on the quality of such supervision and thus the quality of the data. In order to support field workers as much as possible, some specific requirements need to be considered: Firstly, through program design, a prescribed usage protocol or data collection protocol can be enforced to a lesser or greater degree, relieving field workers to attend to more important issues. Secondly, by requiring specific, structured information from each respondent for successful enrolment, the challenge of sourcing much needed information at a later stage, is alleviated. In creating standard profiles for respondents, the user interface can be used to enforce entry of the required fields and their various formats. Lastly, an important aspect of speech data collection is the ethical confirmation of consent. By ensuring that all respondents have seen and agreed to the terms and conditions prior to taking part in a recording session, this important issue is addressed in a straightforward fashion.

3.1.7. Providing support for contractors

As the main agents overseeing complete recording campaigns, contractors may be responsible for recording any financial remuneration awarded to respondents for services rendered. By providing a specific field in the graphical user interface during session enrolment for the remuneration agreed upon, contractors can easily keep track of any such expenditure in electronic form. Further, since the overall responsibility of the quality of data typically lies with the contractor, it is important for contractors to be able to associate specific recording sessions
with each field worker responsible. By enforcing an enrolment process for each field worker along with the enrolment of each respondent, individual performance management of field workers is facilitated, aiding the contractor and potentially impacting the overall data quality.

3.1.8. Simplifying post-processing of data

When data has been collected for a language, this data needs to be developed into an ASR corpus. This involves, for instance, renaming files according to corpus conventions, grouping files into a predefined folder structure and performing advanced quality control. In order to simplify the automation of the post-processing of these files – which are typically large volumes of data – a number of additional requirements need to be adhered to, such as, consistent file and folder naming conventions to ensure that all file names are unique across all recording devices, and across all languages recorded. Furthermore, personal information of field workers and respondents must be easily separable from collected data to avoid additional post-processing of data to remove references that could potentially be linked to an individual’s identity.

The Woefzela tool was developed with the aim of incorporating these and other high-level design requirements, as further discussed in the next section.

4. The Woefzela tool

Typical usage of Woefzela is based on field workers canvassing, enrolling, training and guiding respondents to provide the actual speech data. These field workers are therefore responsible for the actual data collection process. Contractors, on the other hand, are generally responsible for a complete data collection campaign, typically recording a number of languages in parallel, and are required to recruit field workers as needed. From the perspective of a field worker, the process of acquiring data from a single respondent, corresponds to the following protocol:

1. Screening: The language ability and fluency of the respondent is assessed by a qualified mother-tongue speaker, prior to the respondent being enrolled for any further activities. This is usually done on paper.
2. Registration: A basic electronic record of the respondent’s personal information is entered into the mobile device, including a record of data collection consent, and any agreed rewards for services rendered.
3. Training: The respondent is trained on the use of the tool by a field worker, and records a small set of prompts (we have found 15 prompts to be a reasonable trade-off), in order to familiarise the respondent with the general function of the application. This is called a training session.
4. Recording: Upon successfully completing the training session, the respondent is presented with a target number of prompts, in our case 500, while recording the audio data. This is called a recording session.
5. Reward: Upon completion of the recording session, the session is automatically terminated by the application and the respondent is rewarded by the field worker, as per prior agreement.

4.1. Functional description

A description of the functionality implemented in Woefzela is provided below, highlighting the means through which the design requirements have been addressed.

Capturing respondent and field worker profiles. This information is essential in keeping track of any data associated with both field workers and respondents. The profiles generated from this information also enable a new session to load information from a field worker or respondent when subsequent sessions involve the same person, aiding the association of data from a single speaker. Typically, only the field worker information is frequently re-used, but for early session termination, or other reasons, respondent profiles can also be reloaded. Also of importance is the fact that this profile information can be used, with proper consent, in the process of recruiting further first language speakers through these already established contacts.

In order to associate field worker profiles and respondent profiles with session meta data as well as all recordings, a unique key is generated for each field worker and respondent. The primary criteria used for such a reference key are its uniqueness across all persons for any compiled speech corpus recorded with Woefzela, and that the association between the key and the individual must only be traceable in one direction. That is, by knowing the person, the key must be calculable, but by knowing the key, it must not be computationally tractable to determine who the person is. This is to protect the privacy of individuals. To facilitate such a reference, a Message-Digest algorithm (MD5) value is generated for each profile.

Controlling training and recording sessions. The successful completion of a training session prior to advancing to a recording session is controlled by the application to ensure compliance to the recording protocol.
Selecting and loading prompt batches. As the initial step of any recording session, the software selects the targeted number of prompts from a textual corpus provided on the SD card. The primary requirement in this regard is that the actual recorded prompts must converge over time, and across all devices, towards a uniform distribution of the frequency of each textual prompt in the input corpus. This is to ensure that the tri-phone coverage provided by the final corpus closely approximates the intended coverage at the time of the design of the textual input corpus. To achieve this, the starting point of the batch of utterances to be recorded is randomly selected with a uniform distribution, from the complete input prompt corpus.

Presenting a prompt. The software selects the first and subsequent prompts from the loaded batch of prompts, and presents these to the respondent for recording. The key design requirement in this case is compliance with the UTF-8 Unicode standard, since the languages to be recorded may employ non-ASCII characters – for example, diacritic symbols which may be crucial to the pronunciation of words or phrases.

Controlling audio recording button functionality. The primary functional component of Woefzela is the recording, possible re-recording, playback and finalization of the audio for each prompt. The respondent is able to start a recording at will and commence to speak the required prompt. Stopping the recording either after completion of recording the audio, or due to the desire to re-record or skip a specific prompt, forms part of this functionality. Should a respondent wish to play back a recording, he or she is able to do so. If the respondent chooses to skip a prompt, he or she is presented with a list of reasons for skipping a prompt which could aid later prompt text revision should they desire to make known their reason.

The next button, causes the software to move to a subsequent prompt if the target number of good prompts have not yet been reached. However, the respondent is only able to move to the next prompt by either recording audio data to be associated with the current prompt, or by opting to select the skip button.

For simplicity of manipulation, the WAVE-format was chosen for recording. Although not the most compact format of storing or transferring audio data, these files can be easily understood, manipulated and inspected with basic software tools available. The specific file format used has a fixed sample rate of 16 kHz and uses 16-bit mono linear PCM encoding, which was found to be sufficient for wideband data collection.

Store audio and meta data on SD card using associative file names. Since the primary means of storing and exchanging data with Woefzela had to be Internet independent, all file retrieval and storage is done via the SD card. The recorded audio files are stored on the SD card in a logical file system per session. Any accompanying files related to a specific audio file, such as the XML-file containing the prompt string and selected meta data, is stored in the same location with an associated file name.

Perform QC-on-the-go on each audio file. Integral to supporting the notion of maximising recording opportunity, and avoiding any unnecessary losses, is Woefzela’s on-device QC-on-the-go functionality which will be discussed in more detail in the next section. When quality control has been performed on an audio file, a resulting meta data file is created, capturing the outcome of this quality analysis. This file is also in an XML-format to facilitate easy parsing and aggregation of data during post-processing, and is linked to the prompt and audio files through associated file names. It is also worth noting that Woefzela informs the respondent of the current number of recordings that have passed or failed the QC criteria by indicating these totals at the bottom of the main recording interface during a recording session. This is a passive form of feedback to the respondent to avoid any disruptive messages while at the same time acting as a teaching aid to avoid repeated errors.

Control session termination and closure. Upon reaching the target number of recordings passing the quality control criteria, Woefzela informs the respondent of this state and subsequently finalises the session and all its associated outputs.

Typical user interfaces for capturing the field worker’s profile and performing the recording are shown in Figure 1. For more information on Woefzela’s functionality and use, please refer to https://sites.google.com/site/woefzela.

4.2 QC-on-the-go

One of the observations that featured in the design of Woefzela was that traditional methods of verifying data quality during post-processing introduce unnecessary losses when collecting speech data. If this can be done on the fly, such inefficiencies can be avoided. Since Internet independence must be maintained, this quality control mechanism cannot be performed on back-end servers in time to change the target number of prompts based on the quality of the recorded prompts, and thus this functionality must be implemented on the device itself, hence, QC-on-the-go.

Thus, once a respondent has finished recording a prompt, the audio file is submitted to a QC-on-the-go service, running in the background on the device for quality checks, with the results of these checks being written to an XML-file. If any of the specified quality criteria are not met for an audio file, an additional prompt is loaded for the respondent to record, thus pursuing a target number of good recordings, and not simply any audio data (De Vries et al., 2011).

A basic set of quality control criteria that was deemed appropriate and computationally tractable for the hardware available at the time, were initially chosen. These criteria was by no means exhaustive, but simply provided an initial starting point for the proposed novel QC-on-the-go process. Further research into the use of other criterion such as signal-to-noise ratio, to name but one, could be explored. The quality criteria currently implemented in Woefzela are:

---

1An accompanying application called WUpload was developed that extends the functionality of Woefzela to upload recorded files to a server when Internet access is available.
Volume level. On mobile phones it is often easy for the user to unintentionally cover the microphone causing the volume of the recording to be too low, or speaking too closely into the microphone – causing the volume to be too high. This flag in the XML-file will indicate such QC failure.

Start/stop errors. Should a user start speaking prematurely, truncation of the speech signal will happen at the start of the recording. On the other hand, if the user presses the button to stop recording too early while still speaking, truncation at the end of the recording results. In order to avoid both these errors, as well as provide feedback to the user, an empirical root-mean-square threshold was set for the first and last N milliseconds of the audio. Should the threshold be reached, it was deemed that an unacceptable amount of energy was present in the signal at the start/end, and that potential information could be lost (Badenhorst et al., 2012). To further inform the user when the hardware is ready, in an attempt to avoid such truncation errors, the prompt text changes colour to indicate the various states.

A side effect of this criterion is important: In very noisy environments such as traffic, close proximity to air-conditioning noise or background sources such as bird songs (common in rural African settings), the recordings often failed the quality criteria, seemingly indicating a start/stop truncation error, when the actual problem was too much energy in the surrounding ambient noise, causing such a flag to be raised. This is in fact a desirable conclusion, as such high ambient noise levels indicate that the environment is unsuitable for speech data collection.

4.3. Successfully collected corpora

The initial ASR corpora, consisting of audio data with associated transcriptions and meta data, successfully collected with Woefzel, are summarised in Table 1. These files all passed the QC-on-the-go criteria built into Woefzel; the quantity of data collected thus highlights the usability and effectiveness of this tool for collecting ASR data for under-resourced languages.

<table>
<thead>
<tr>
<th>Language</th>
<th>Speech data (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>isiZulu</td>
<td>59.02</td>
</tr>
<tr>
<td>isiXhosa</td>
<td>76.48</td>
</tr>
<tr>
<td>Afrikaans</td>
<td>68.97</td>
</tr>
<tr>
<td>Sepedi</td>
<td>69.16</td>
</tr>
<tr>
<td>Setswana</td>
<td>70.16</td>
</tr>
<tr>
<td>Sesotho</td>
<td>72.42</td>
</tr>
<tr>
<td>SA English</td>
<td>72.86</td>
</tr>
<tr>
<td>Xitsonga</td>
<td>86.20</td>
</tr>
<tr>
<td>siSwati</td>
<td>81.90</td>
</tr>
<tr>
<td>Tshivenda</td>
<td>80.80</td>
</tr>
<tr>
<td>isiNdebele</td>
<td>60.40</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>798.37</strong></td>
</tr>
</tbody>
</table>

Table 1: Summary of ASR corpora successfully collected with Woefzel (De Wet and Titmus, 2011).

5. Experimental methodology

In De Vries et al. (2011), the effectiveness of the basic quality criteria described above was evaluated; this approach was shown to provide a common-sense way of improving the amount of acceptable data that is collected from each speaker. Subsequently to that initial study, Badenhorst et al. (2012) also proposed an additional criterion, namely utterance length as a measure of the quality of the recorded data, and showed that it could add value if the number of additional errors detected is taken as the success metric.

In this study, we continue to use the quality criteria described above in Section 4.2, but also additionally employ the utterance length criterion proposed by Badenhorst et al. (2012). Hereafter in this article we will simply refer to quality criteria implying both sets of criteria, and will thus be investigating the combined effect of these criteria on ASR system performance.

Knowing whether to target a specific number of good recordings versus a fixed total number of recordings regardless of the recording quality, is an important upfront decision to be made when starting the data collection process with a tool like Woefzel – which has the capability of requiring a speaker to read additional prompts during the same recording session, if a previous recording did not meet required quality criteria. This decision is important because of the huge logistical effort involved in making large scale, typically distributed, recordings. It is also important because of the effort that the actual speaker making the recordings has to make by either (a) recording a fixed number of prompts without any interruption or feedback, or by (b) pursuing a moving target number of recordings and thus having to record additional prompts until a set of quality criteria have been met. The former is a straightforward task, while the latter could be more tiring and even disheartening if the speaker keeps on making recording "mistakes".

If a moving target is to be pursued, two subsequent choices have to be made once the data has been collected and further processing or use of the data needs to start. These options are either using only the good data during ASR system development, or using both the good and the bad data (that failed the quality criteria) to build an ASR system.

This might seem to contradict the philosophy of on-device quality control in the first place, but remains a very realistic question for under-resourced languages i.e. we have made all the effort, what do we do with the ‘extra’ data that was collected? Will adding it to the training set help or hurt the overall ASR system performance?

Thus with the above three decisions to be made, two comparative experiments were developed and are summarised in Table 2. In the first experiment (Section 6.1: Comparing performance with equal training-set sizes), we will be comparing

---

2In practice, the field worker should monitor the progress of each respondent and stop the recording process altogether if the speaker is consistently not meeting the quality criteria. This could also be enforced by the software tool, but such a hard-limit is currently not yet implemented in Woefzel.
scenarios 1 and 2 (with and without quality control). In scenario 1 only data that passed all the quality criteria for acoustic model training are included, while scenario 2 includes a similarly sized corpus, but randomly selected from all data (without any QC filtering). From this experiment we could learn whether the quality control system is worth the additional effort of implementing and adhering to. We should note that scenario 2 is simulated to some degree because QC was actually used, but its effect could be discounted through careful experimentation by selecting an increasing amount of data randomly from each speaker regardless of the QC results.

In the second experiment (Section 6.2: Comparing performance with equal amounts of collected data), we are comparing scenarios 1 and 3 where quality control was used in both cases, but where in the one case only data passing all the quality control criteria was employed (Scenario 1), and in the other scenario all data was employed, regardless of quality control results (Scenario 3). From this experiment we could learn whether we should train ASR acoustic models only on data that passed all the QC criteria, or we should capitalise on all the data collected and simply train on all the data instead of only the good data.

5.1. Speech corpora

Two of the eleven initially collected languages were selected for this study, namely isiZulu and Afrikaans. Afrikaans was selected for its limited amount of background noise (on average), while isiZulu was selected for its higher ambient noise level – both based on the ambient noise analysis in Badenhorst et al. (2012). The data was collected using Woefzela with the on-device QC described in Section 4 enabled.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>QC</th>
<th>Target</th>
<th>Typically</th>
<th>Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>500</td>
<td>600</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>500</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
<td>500</td>
<td>600</td>
<td>600</td>
</tr>
</tbody>
</table>

In these experiments only flat phone recognition is used as a measure as it is less influenced by language modelling and the recognition vocabulary choices which diverge greatly across applications and languages.

Table 2: Summary of recording scenarios and experiments explored. Experiment 1 compares scenarios 1 and 2, while experiment 2 compares scenarios 1 and 3.

The collected data was divided into three data pools or subsets, namely all data, base350, and clean data as depicted in Figure 2. The all data pool consists of all the data that was collected during the recording campaign without any quality assurance or selection performed on this data. A subset of all this data is the base350 pool. This pool consists of data only from speakers that have recorded at least 350 utterances to ensure a fair distribution of data across speakers. A further subset of the base350 pool is data that have passed all the quality assurance criteria; the latter is referred to as the clean data pool.

Table 3: Summary of ASR corpora for Afrikaans (af) and isiZulu (zu) used in experiments.

<table>
<thead>
<tr>
<th>Data pool</th>
<th>Recordings</th>
<th>Hours</th>
<th>Male</th>
<th>Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>af all</td>
<td>99 115</td>
<td>82</td>
<td>90</td>
<td>99</td>
</tr>
<tr>
<td>base350</td>
<td>94 138</td>
<td>79</td>
<td>86</td>
<td>94</td>
</tr>
<tr>
<td>clean</td>
<td>75 483</td>
<td>64</td>
<td>82</td>
<td>89</td>
</tr>
<tr>
<td>zu all</td>
<td>82 552</td>
<td>89</td>
<td>90</td>
<td>76</td>
</tr>
<tr>
<td>base350</td>
<td>73 463</td>
<td>81</td>
<td>87</td>
<td>71</td>
</tr>
<tr>
<td>clean</td>
<td>56 308</td>
<td>62</td>
<td>78</td>
<td>63</td>
</tr>
</tbody>
</table>

The test set selection

Since the creation of a “gold standard” test set of significant size compared to 75 483 recordings for Afrikaans, and 56 308 recordings for isiZulu is no small task—and even then still subject to inter-rater variability—both good and poor quality data were included in the test set with confidence in the results evaluated by means of a ten-fold cross validation of all data for a specific experiment. These ten folds are illustrated by the segments in Figure 2.

The test set, although not a pool, is also indicated on the same figure in order to show that during ten-fold cross-validation, a test fold consisted of data from a mutually exclusive set of speakers, with data in such a set consisting of all the data in the base350 pool for the same speakers. That is, the test fold always consists of a fraction of the base350 data (both passing and failing the various QC criteria) from a mutually exclusive set of speakers. Acoustic model evaluation therefore involved all the data in the base350 pool once all ten folds have been evaluated. Acoustic models were trained on nine of the ten folds, and testing of the model performance was conducted using the tenth mutually exclusive fold, consisting of all the data in the base350 pool for the selected speakers.

5.2. Acoustic modelling

A context-dependent, cross-word, tri-phone recogniser was built using HTK (Young et al., 2009). Feature extraction on the speech audio data yielded 39 dimensional feature vectors:
13 MFCCs with their first- and second-order derivatives. An MFCC window size of 25ms was used, with a step size of 10ms. Cepstral Mean Normalisation was applied on an utterance level. Each phone has three emitting states with 8 Gaussian mixtures per state and a diagonal covariance matrix. Pronunciation prediction was performed on all words by using the Default&Refine algorithm (Davel and Barnard, 2008), with grapheme-to-phoneme rules derived from the Lwazi dictionaries (Davel and Martirosian, 2009).

In order to balance the reported accuracies, the insertion/deletion ratio was initially balanced across ten cross-validation folds and the average insertion penalty selected and kept constant for all further experiments.

6. Experimental results

6.1. Comparing performance with equal training-set sizes

Training set. The training set for the baseline experiment consisted of all the data in the base350 data pool (i.e. prior to the application of any QC criteria), that is scenario 2 in Table 2, divided into the ten folds for cross-validation purposes. The clean experiment’s training data consisted of only data that passed all the QC criteria as depicted in Table 2 by scenario 1. Equal numbers of training utterances were employed for both systems.

Results. Figures 3 and 4 show the mean recognition accuracy calculated over the ten folds during cross validation for the Afrikaans and isiZulu corpora, respectively.

Although the observed differences in performance are small, they consistently favour the clean data as shown in Figure 5, where the mean accuracy differences between comparable folds in the two data sets are shown across all folds for both languages. The error bars in Figure 5 correspond to one standard error. These differences are seen to be statistically significant in almost all cases. When using a paired t-test, with folds involving the same test data being paired together, the differences between the two Afrikaans systems are all found to be significant at the $p = 0.05$ level, as are all the isiZulu differences except the pair around 5 000 training utterances.

6.2. Comparing performance with equal amounts of collected data

Training set. The previous experiment compared the accuracies when a fixed number of utterances are used for training, with and without QC. However, an alternative measure would be to compare the accuracies achieved when fixed amounts of collected data are compared – hence, the amount of data used for training a clean system is actually less than for the comparable baseline system, since some data is discarded during QC. In this experiment scenarios 1 and 3 as summarized in Table 2, will be compared.


**Results.** Figure 6 shows the relative gain (with negative numbers indicating a loss) in system accuracy when comparing the clean and baseline systems. By this measure, we see that the baseline system actually outperforms the system trained on clean data, especially for small quantities of collected data. This outcome clearly depends on the ratio of clean data versus bad data present in the recordings. In Figure 6 that ratio was kept at 80%, which is approximately the value observed in our collections. Table 4 summarises the accuracies achieved with different good-to-bad ratios of data for Afrikaans. (This comparison was made on one randomly selected fold, and at 22,000 recordings.) We see that the loss in accuracy with more “contaminated” data is relatively small, suggesting that the data failing our QC is only mildly detrimental to ASR system performance compared to the clean data.

Table 4: Summary of ASR performance for various good-to-bad ratios at 22,000 recordings for Afrikaans.

<table>
<thead>
<tr>
<th>Ratio (good/bad)</th>
<th>Contaminated dataset Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100/0</td>
<td>74.16</td>
</tr>
<tr>
<td>70/30</td>
<td>73.70</td>
</tr>
<tr>
<td>50/50</td>
<td>73.02</td>
</tr>
<tr>
<td>40/60</td>
<td>72.31</td>
</tr>
</tbody>
</table>

7. Conclusions and future work

Based on our practical experience, we have described several requirements of a tool that is suitable for ASR data collection for typical under-resourced languages. We have also described a smartphone application, Woefzela, that was designed to meet these requirements.

One of the novel capabilities of Woefzela is its ability to perform quality control during data collection. Although this function performs according to its design specifications, we have found its influence to be surprisingly limited during real-world data collections: by one measure (accuracy of a trained ASR system with fixed amounts of training data), quality control leads to only small improvements, whereas the impact on accuracy by another measure (accuracy of a trained ASR system with fixed amounts of collected data) is actually negative.

Our intuition on the benefits of quality control may still be confirmed in environments where more severe errors occur — for example, if respondents have more trouble reading the target language. However, we have not seen such circumstances in our data collection efforts. Furthermore, the measures used do not include the more severe errors as these recordings generally failed at the automatic alignment stage.

At the same time, a number of common sense and external reasons remain as to why at least basic on-device quality criteria should be employed:

1. De Vries et al. (2011) showed that an increased amount of good data should result from employing very basic on-device criteria.
2. Device or protocol failures could be detected earlier on without wasting effort and time if basic criteria is employed.
3. A too high ambient noise level could be detected as described above.
4. Although QC did not seem to have much of an impact, the fact that basic QC was used, caused speakers to record additional data until the target number of good recordings were met. This has a salient impact on the good-to-bad ratio of recordings, which increased the overall ASR system performance when all collected data is used as described in scenario 3 above.
5. Basic QC can help to identify individuals who consistently make mistakes early on, as can also be seen from Van Heerden et al. (2012), and thus affording someone else the opportunity to rather occupy a recording device than continuing to assume that certain speakers will...
improve after a cut-off point. This cut-off point could possibly be implemented in software.

We believe that Woefzel is a versatile and useful tool for ASR data collection in under-resourced languages. Its value would be enhanced by tools that manage further aspects of the ASR development process (e.g. tools that automatically perform corpus collation and training of the acoustic models for ASR). When such tools gain widespread use, ASR applications in under-resourced languages are likely to become commonplace.

References


