Woefzela - An open-source platform for ASR data collection in the developing world

Nic J. de Vries [1,2], *Jaco Badenhorst* [1,2], *Marelie H. Davel* [2], *Etienne Barnard* [2], *Alta de Waal* [1]

[1]Human Language Technologies Research Group, Meraka Institute, CSIR, Pretoria, South Africa
[2]Multilingual Speech Technologies, North-West University, Vanderbijlpark 1900, South Africa
{ndevries, jbadenhorst, mdavel, adewaal}@csir.co.za, etienne.barnard@nwu.ac.za

Abstract

Building transcribed speech corpora for under-resourced languages plays a pivotal role in developing speech technologies for such languages. We have developed an open-source tool for devices running the Android operating system to facilitate the efficient collection of speech data for Automatic Speech Recognition system development. The tool was designed for use in typical developing-world conditions; we present the relevant design choices and analyse the effectiveness of this tool by means of a case study. In particular, we introduce a novel semi-real-time quality monitoring system, which increases the efficiency of the data collection process.

Index Terms: speech resource collection, automatic speech recognition, developing world, resource-scarce environment, under-resourced languages, android

10.21437/Interspeech.2011-795

1. Introduction

A key step in developing speech technologies for under-resourced languages is the collection of high quality speech data, consisting of transcribed audio with associated meta-data. Collecting such data in typical developing-world environments poses a number of additional challenges such as cost [1], general infrastructure limitations (for example, reliable, cheap access to cloud computing resources [2]), and even the accessibility of first language speakers due to small remote communities or widely distributed speakers. One solution is to make use of relatively inexpensive hand-held devices [3]. The use of smart phones for speech corpus creation has been demonstrated in the *developed* world by Hughes *et al.* [4], and our goal is to provide similar tools that are freely available and adapted for use in the *developing* world.

Below, we first provide motivations for several of our design choices as well as the overall approach taken. We then describe a case study, including analysis of some of the collected data.

2. Approach

DataHound [4] is a commercial speech data collection application developed for the Android platform. It has shown itself as extremely useful for speech data collection in the developed world, and the mobility afforded by the DataHound approach holds additional advantages in the developing world. However, we discovered a number of drawbacks when using DataHound in our unique developing-world context, which lead to a somewhat modified approach in designing a new tool. Our general approach is based on the assumption that collection will be overseen by *field workers*, who are responsible for canvassing, enrolling, training and guiding *respondents* who provide the actual speech data. The field workers are therefore responsible for the collection process, including aggregation of data collected at different venues and times. Our aim is to provide practical support for these field workers in a number of ways.

2.1. Limiting reliance on Internet connectivity

Limited Internet connectivity and the cost of uploading recordings, when using DataHound, posed serious financial and logistic problems for field workers in South Africa. Thus, the general client/server architecture of DataHound needed to be eliminated, for environments such as ours. In order to overcome this drawback, reliance on Internet connectivity was eliminated by employing the external data storage card on smart phones as the data exchange mechanism. Since field workers are always in close proximity to respondents, the utilization of a laptop to upload data onto a hard drive and download prompts to the device solved this problem.

2.2. Developing open, customisable source code

Given the general scarcity of resources in developing-world environments, and in order to stimulate the development of speech technologies for under-resourced languages, it was essential to develop an open, customisable tool. Android as a free and open-source mobile operating system, provided an excellent framework for developing such a tool. Having a tool that is freely available at no cost to end users, and at the same time completely open and customisable for different contexts also allows for adaptation to novel challenges that may occur in other developing-world environments.

2.3. Providing support for semi-trained field workers

While crowd-sourcing presents a number of great opportunities in both the developing and developed world, recruiting of first language speakers that are sparsely distributed across large geographic areas is not a trivial task. To aid field workers in establishing networks of same language speakers, basic contact information of each speaker was requested and captured by the application as part of the enrolment process. This information could then be subsequently used to re-establish contact with speakers of a specific language in order to help recruit additional speakers spread over wide geographic areas. (Previously, field workers would maintain a separate database of first language speakers for further recruitment purposes.)

This information can also be useful for other reasons: for example, in South Africa there is a legal requirement that even part-time workers must be above the age of 16 in order to be re-

Copyright © 2011 ISCA

munerated for their data recording services. Simply requesting the respondent's South African identity number provided the field worker with confirmation of the person's actual age, while also circumventing embarrassing situations that may arise from asking a person's age to ascertain such legal compliance.

As the above information (contact information, identity numbers) are of a personal nature, additional steps were taken to provide sufficient privacy protection. The information collected as part of the respondent's profile was kept completely separate from the data that the person recorded by generating a unique key from the information provided using a Message-Digest algorithm (MD5), in order to maintain a clear separation of any personal information from the resulting speech corpora compiled.

2.4. Verifying data quality in semi-real-time

The primary goal of recording speech data is *to ensure that a certain target number of "acceptable" recordings are collected for each speaker*. By guiding field workers towards a fixed number of acceptable recordings per speaker, rather than a fixed number of *total recordings* per speaker, we can ensure that more usable speech data is collected.

If significantly less data than the target is recorded, e.g. the subset of data that is of sufficient quality is too small, the data will not be representative of the speaker. On the other hand, if significantly more acceptable data than required is recorded per speaker, it may skew the resulting acoustic model used for speaker-independent speech recognition. By closing the loop on the quality of recordings as quickly as possible (on the mobile device), a waste of various resources is thus avoided. Although Hughes *et al.* [4] attempted to estimate the rate of a specific number of errors to be expected when using such a tool, in practice certain types of errors were found to be much more frequent as the later analysis will indicate. We call this method of implementing quality control (QC) in semi-real-time on the phones "QC-on-the-go."

3. QC-on-the-go

Traditional methods of checking the quality of data during post-processing to ascertain the quality and sufficiency per speaker, introduces unnecessary losses when collecting speech data for under-resourced languages and in resource-constrained environments. *Quality* depends, of course, on the particular application for which the data is intended; we have implemented a number of measures that are appropriate for developing ASR systems, but the open nature of *Woefzela* allows for easy modification or replacement of these measures. Since Internet connectivity can not be assumed, this semi-real-time QC can not be performed on back-end servers in time to change the target number of prompts based on the quality of recorded prompts. The quality criteria used in this paper as well as some of the implementation details are briefly described next.

3.1. QC flags

The following quality checks are performed on the audio file and the subsequent results stored in an associated XML file:

3.1.1. Volume level

A basic check of the volume of the recording formed part of the quality check criteria. 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, and the distance between the speaker's mouth and the microphone is also highly variable.

DataHound provided a form of visual feedback to the user by indicating on a volume level bar what the current volume is. In the newly developed application we chose to monitor the peak volume during a recording in order to stop a user from moving to the next prompt if the volume was deemed insufficient. Should clipping of the audio signal be detected, the user is also barred from moving to the next prompt.

This first-level check flags some basic errors (such as covering the microphone completely); a more advanced volume check is employed for more subtle problems. Once the user has finished recording a prompt, the audio file is submitted to a service running in the background on the device for further quality checks. With regard to the volume, a more thorough check is done during this stage, causing a file to be flagged as having too low or too high volume should the set criteria not be met.

3.1.2. 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, a specified 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. To further inform the user when the hardware is ready, the prompt text was made to change colour to indicate the various states. In this aspect, DataHound has a superior implementation by maintaining a rotating buffer for recordings and subsequently writing the buffer to a file while including 0.5 seconds of recorded audio before initiation and after termination of the recording.

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), our quality check feature caused the recordings to fail the quality criteria, seemingly indicating a start/stop truncation error, when the actual problem involved too much energy being present in the surrounding ambient noise. This is in fact a desirable conclusion, as such high ambient noise levels indicate that the environment is unsuitable for speech data collection.

3.2. Respondent data collection process

Once a speaker has finished recording a prompt presented to him/her, and moves on to the subsequent prompt, the audio file is queued for quality checks. As soon as the processor of the phone becomes available, a series of quality checks are performed on the next file in line for QC. The results of these quality checks are written to an XML file along with the prompt presented and the audio recorded.

Should an audio file fail any of the quality criteria, the XML file would indicate this fact and the application would *load additional prompts* for the user to complete prior to ending the recording session. The user is also informed that a certain quality criterion has been failed by means of statistics presented on the screen of the device, to serve as a passive form of training/feedback – potentially helping to avoid further failures.

4. Case study: NCHLT Project

One of the projects under the auspices of the National Centre for Human Language Technology (NCHLT) [5], funded by the South African Department of Arts and Culture, set out to collect 50-60 hours of broadband speech data for each of the eleven official languages in South Africa, spanning six of the nine provinces over a period of 10 months. This forms part of an initiative to encourage the development of speech technologies in all of these official languages. In order to develop a balanced corpus two hundred speakers (100 male and 100 female) of each of the languages are to be recorded, with around 500 utterances per respondent. The resulting corpus will consist of more than 1.1 million utterances from more than 2,000 individuals in a developing-world environment. Therefore it is a rigorous test bed for our tools.

This case study analyses relevant statistics of four of the languages for which speech data has been collected, namely Afrikaans (af), South African English (en), Sepedi (Northern Sotho) (ns) and Zulu (zu). As a brief overview of the actual recording process, the steps followed were:

1. Screening: Assessment of the language ability of each respondent is ascertained by a qualified mother-tongue speaker of the target language.
2. Registration: A profile is created for the respondent including information such as identity number, age, gender and data collection agreement consent.
3. Training: The respondent is trained on the use of Woefzela and records at least 10 prompts in order to be comfortable with the general functioning of the application.
4. Recording: The respondent is presented with the target number of prompts, with the application loading additional prompts as required by quality criteria.
5. Reward: Upon completion of the target number of quality prompts, the session is automatically terminated by the application and the respondent is rewarded as per prior agreement.

4.1. Analysis

Although common sense suggests that the pursuit of a desired number of *good* recordings (as opposed to having a fixed number of recordings per speaker) will lead to a better distribution of usable recordings, the question that arises is whether an even simpler solution would not be sufficient. Will increasing the fixed target with a certain number of prompts not compensate for errors to be made? Also, if a moving target of good recordings is pursued, how close can one expect to get to the set target? Both these questions are explored below.

4.1.1. Data set selection

In order to investigate these questions, a subset of the complete data set was selected for each language. This subset consisted of all the sessions for which additional prompts have been loaded due to audio files failing any of the quality criteria listed above. Eliminated from this list were sessions that *terminated* before reaching the initial target of 500 prompts. For Afrikaans, of the 130 *potential* sessions, 11 terminated before reaching 50% of the prompts and 19 terminated with between 50% and 75% of the prompts. The remaining 100 sessions (amounting to just under 27 hours of speech data excluding inter and intra-word silences), formed the dataset for Afrikaans for our analyses below.

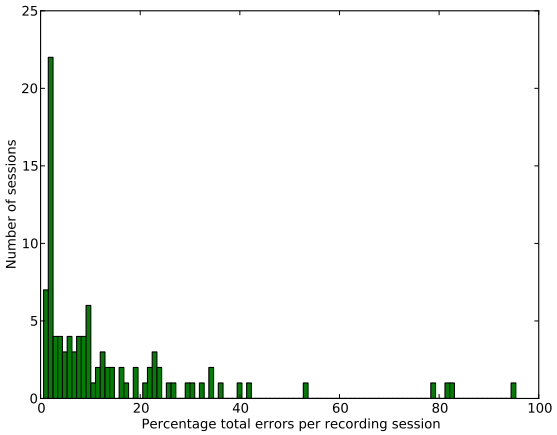

Figure 1: *Histogram of percentage total errors made per recording session for Afrikaans.*

Similar information for the four languages analysed is summarized in Table 1.

Table 1: *Breakdown of the number of sessions per category to arrive at the analysis data set.*

Language	Potential	Terminated	Analysis	Hours
af	130	30	100	26.9
en	122	33	89	20.4
ns	178	73	105	28.8
zu	176	55	121	42.4

4.1.2. Percentage failed recordings

Figure 1 shows the distribution of the percentage errors that Afrikaans users made per session i.e. failed files versus total number of files, for the first 500 prompts; where QC has not yet had the effect of loading additional prompts. Table 2 summarises some statistics of this typical distribution for the four languages under investigation.

Table 2: *Summary of percentage total errors made per recording session for four languages.*

Language	Mean	Standard Deviation
af	13.9%	18.1%
en	10.6%	11.6%
ns	11.7%	16.4%
zu	11.0%	12.7%

A number of interesting observations can be made from Figure 1 and Table 2. It seems that some users do not correct their behaviour based on the statistics presented to them on the device in semi-real-time. This suggests that a more disruptive method could be used to guide the behaviour of respondents, but for now we simply accept that the variability in success rates will require some respondents to make a substantial number of additional recordings. Keep in mind, however, that a failure percentage of (say) 20% implies that for that specific session 80%

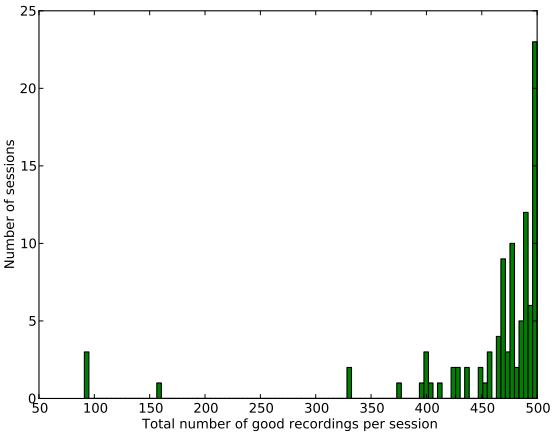

Figure 2: *Histogram showing the number of acceptable recordings made per session using a moving target approach for Afrikaans.*

of the data is of sufficient quality at the moment that the session completes.

The substantial inter-speaker variability observed in our experiments suggests that our concerns related to a fixed number of recordings (without QC) are justified; targeting the number of acceptable recordings (i.e. a "moving target" of total recordings) is indeed beneficial.

4.1.3. Comparison between fixed and moving target

The effectiveness of targeting the number of acceptable recordings during the recording process can be evaluated by computing the number of good quality recordings at the time that the session completes. Figure 2 shows a distribution of the number of recordings made per session that passed the QC criteria for Afrikaans.

Since the distributions of the other three languages are again similar to that of Afrikaans, the overall results are summarised in Table 3.

Table 3: *Summary of number of acceptable recordings made per session for four languages.*

Language	Mean	Standard Deviation
af	456	79
en	475	27
ns	461	72
zu	471	52

A few brief comments are in order. It is important to note that although the actual target number of good recordings is still not achieved, the bulk of the data now lies well within 10% of the actual target as can be seen from Figure 2. One possible reason - in most of the cases - for not achieving the actual target is the fact that the semi-real-time QC running as processor time becomes available, will inherently lag behind the actual data. By employing faster CPUs or optimising the QC algorithms this lag could be reduced, but unless real-time QC can be achieved, there will always be a small lag.

Equally important as the target number of samples, is the observation that the variance is substantially less than in the uncontrolled case. This observation therefore provides the opportunity to add a fixed number of recordings to the initial target without incurring great losses. We have effectively decreased the variance on the number of acceptable recordings made, by adding more prompts for some users and fewer prompts for others.

5. Conclusion

Challenges faced in collecting data for under-resourced languages called for an automated quality control method in order to optimise the recording opportunity per respondent session. We recommend a near real-time quality control process deployed on the mobile phone that monitors the quality of recordings and adds additional prompts to the session if the quality of recorded prompts falls outside a predefined set of criteria. Our toolkit for doing so is freely available as open-source software for the Android platform at https://sites.google.com/site/woefzela/, and will hopefully stimulate further speech resource development for resource-scarce languages.

6. Acknowledgements

This project was made possible through the support of the South African National Centre for Human Language Technology, an initiative of the South African Department of Arts and Culture.

The authors would also like to thank Pedro Moreno, Thad Hughes and Ravindran Rajakumar of Google Research for valuable inputs at various stages of this work.

7. References

[1] E. Barnard, J. Schalkwyk, C. van Heerden and P.J. Moreno, "Voice search for development," in *Proc. Interspeech*, Makuhari, Japan, September 2010, pp. 282-285.

[2] A. Kumar, A. Tewari, S. Horrigan, M. Kam, F. Metze and J. Canny, "Rethinking Speech Recognition on Mobile Devices," in *Proc. 2nd International Workshop on Intelligent User Interfaces for Developing Regions*, Palo Alto, CA, February 2011, pp. 10-15.

[3] T. Hazen, E. Weinstein, R. Kabir, A. Park and B. Heisele, "Multi-Modal Face and Speaker Identification on a Handheld," in *Proc. Workshop on Multimodal User Authentication*, Santa Barbara, CA, December 2003, pp. 113-120.

[4] T. Hughes, K. Nakajima, L. Ha, A. Vasu, P. Moreno and M. LeBeau, "Building transcribed speech corpora quickly and cheaply for many languages," in *Proc. Interspeech*, Makuhari, Japan, September 2010, pp. 1914-1917.

[5] "Human Language Technology Projects," Meraka Institute, CSIR, Pretoria, South Africa. Online: http://www.meraka.csir.co.za/hlt_projects_nchlt.htm, accessed on 29 March 2011.