SLTU-2014, St. Petersburg, Russia, 14-16 May 2014

Speech data collection in an under-resourced language within a multilingual context

Raymond Molapo[1,2], *Etienne Barnard*[2], *Febe de Wet*[1]

[1]Human Language Technologies Research Group Meraka Institute, CSIR, South Africa
[2]North-West University, Vanderbijlpark, South Africa
rmolapo@csir.co.za, etienne.barnard@nwu.ac.za, fdwet@csir.co.za

Abstract

In this paper, we present an end-to-end solution to the development of an automatic speech recognition (ASR) system in typical under-resourced languages, where the target language is likely to be influenced by one more embedded foreign languages. We first describe the collection and processing of the text corpus crawled from the World Wide Web using the Rapid Language Adaptation Toolkit. In particular, we highlight the challenges faced when foreign languages are embedded within the matrix language. Thereafter, we discuss our speech data collection efforts in under-resourced environments. We finally report on a strategy called transliteration that aids to improve recognition results of our grapheme-based automatic speech recognition system in the presence of embedded-language words.

Index Terms: under-resourced languages, matrix language, transliteration, grapheme-based ASR

1. Introduction

The development of speech systems requires a significant amount of transcribed speech data, which in turn requires text data for the construction of prompts, language models and pronunciation dictionaries. For languages that are regarded as under-resourced, it is often a difficult task to gather all these components. We have therefore set about developing a Web-based framework to assist in various steps of the process. In our research, for a speech system to be developed for a certain language, it is required that the language of interest must at least have a standardised orthography and some presence on the World Wide Web. Our goal is to collect text data of sufficient quality to produce accurate overall system performance. which implies that the data does not need to be perfectly grammatical or monolingual.

However, most of the pages on the World Wide Web are diluted by embedded languages such as English or French. These languages are often the lingua franca of the countries in which the target language is spoken, and appear in various guises such as advertising, loan words, code-switched speech, etc. Embedded languages are languages that are found within the matrix or language of interest. In such cases, the text data extracted from these web sites may be found to contain more of the embedded language than the matrix language [1]. This scenario leads to large amounts of text data being filtered out during post processing, which is an undesirable outcome for resource-scarce languages.

The approach we explore in this paper is to limit the amount of text data lost during post processing. Since there is significant code switching between native dialects and embedded languages, removing some of the text may take away the context of the sentences [2]. Also, grapheme-based automatic speech recognition (ASR) systems – which are our focus, for reasons discussed below – trained with either of the languages mentioned above result in invariably poor recognition results [3]. This is more evident for the case where the majority of the speech data are written in the orthography of another language. To limit the loss of data and still obtain high-quality grapheme-based ASR systems, we use a method called transliteration. The concept of transliteration is to regularize the words from an irregular embedded language to match that of the language of interest (dominant language).

2. Background

Among the many different languages spoken around the world, only a small number can be classified as well-resourced. For our purposes, the languages that do not have widely-available transcribed speech data are classified as under-resourced, despite the fact that some of these languages have millions of native speakers. The reasons for this can range from most native speakers having no interest in speech technology to accessibility problems because the native speakers live in remote areas; most commonly, however, economic issues determine the extent of resources available in a given language. Corpus development is typically quite expensive and these expenses generally prevent resource collection unless there are sufficient commercial reasons to justify the development of language technologies using the collected resources. For the reasons mentioned, most African languages can be classified as under-resourced.

However, with the emergence of mobile, inexpensive and relatively fast computer technology, acquiring text and speech data has become an achievable goal. Various speech tools have been developed to take advantage of these new technological advances. These tools drastically reduce the amount of time and effort speech system developers require to develop ASR or text-to-speech (TTS) systems. Although some of these tools may work as standalone entities, some need to be combined to create an efficient end-to-end system that is fast, intuitive and cost effective.

For the purposes of the current research, a language must have data on the internet, as mentioned in the Introduction. Fortunately, a substantial number of the under-resourced languages

Table 1: Shona vowel: orthography and pronunciation

Vowel	IPA
a	/a/
e	/e/
i	/i/
o	/o/
u	/u/

do have a significant presence on the internet. Such internet sites can be crawled to retrieve the contents of the web pages, and the data can then be cleaned through suitable pre-processing stages to serve as general text corpora. The Rapid Language Adaptation Toolkit (RLAT) [4] includes such pre-processing amongst its numerous capabilities. RLAT permits speech system developers to rapidly collect text data from the internet using web crawlers and web robots. For speech data recording, we used a tool called Woefzela [5] to record prompts read out by carefully selected respondents. The process described above is referred to as end-to-end due the fact that it provides a complete semi automated way to develop a speech system from scratch.

3. The Shona Language

The Shona language is a language on the Bantu branch of the Niger-Congo language family, native to the Shona people of Zimbabwe, southern Zambia, eastern Botswana and parts of Mozambique. Shona is used as an umbrella term to identify people who speak one of the Shona language dialects, namely Zezuru, Karanga, Manyika, Ndau, and Korekore. Zezuru, mainly spoken in Mashonaland, is regarded as standard Shona dialect [6]. Shona is also spoken unofficially in South Africa and is closely related to the Venda language (one of the official languages of South Africa). The language has about 11 million first-language speakers across Southern Africa. Shona is a tonal language with two tones (high and low); the tones are not indicated in the script form of the language, which uses Roman alphabet with a fairly regular relationship between orthography and pronunciation. The Shona language consists of five vowels and thirty five consonants. Table 1 lists the phonetic pronunciations of the vowels and Table 2 lists the consonant pronunciation.

4. Text data collection

For the purposes of text data collection, a tool called Rapid Language adaptation Toolkit (RLAT) was incorporated into our end-to-end system. RLAT was developed at Carnegie Mellon University (CMU) and Karlsruhe Institute of Technology (KIT). It permits system developers to quickly crawl and clean text data from the internet. Even though we use RLAT for text data collection, it can also be used for speech data collection.

To start the crawling process, we compiled a list of the 100 most frequently used Shona words and sent it to the RLAT team at the Karlsruhe Institute of Technology, in order to create a place holder for the target language (Shona in our case) on the RLAT web site. Subsequently, a list of universal resource locators (URLs) pointing to Shona websites is uploaded to the site. RLAT then crawled the internet, starting from those URLs and collecting documents that contain a sufficient concentration of the 100 common words.

Table 2: Shona consonant: orthography and pronunciation

Consonant	IPA	Consonant	IPA
b	/b/	bh	/ɓ/
ch	/tʃ/	d	/d/
dh	/ɗ/	dzv	/d͡ʒβz/
dy	/dg/	f	/f/
g	/ɠ/	h	/h/
j	/dz/	k	/k/
l	/l/	m	/m/
mbw	/mbeg/	mh	/m̈̊/
n	/n/	ng	/ŋ/
p	/p/	r	/r/
s	/s/	sv	/s͡ɸs/
sw	/skw/	t	/t/
ty	/tk/	tsv	/t͡ɸs/
v	/β/	vh	/v/
w	/w/	y	/j/
z	/z/	zv	/z͡βz/

The crawling process may take several days or weeks to retrieve all the sites. For a more direct and robust web crawling, RLAT permits developers to upload a text file with a list of URLs to initiate the crawling process. Starting from the domains shown in Table 3, we have managed to collect a total of 19 Megabytes of text data using this process. The data contained approximately 267 000 sentences, which included over 2.6 million tokens. The crawled text data was found to contain numerous characters and words that needed to be cleaned and normalized.

Table 3: Shona URLs used to initiate crawling

Order	URL
1	http://mudararatinashemuchuri.blogspot.com
2	http://vashona.com/shona-news
3	http://www.watchtower.org/ca/jt/
4	http://www.kwayedza.co.zw/
5	http://www.voanews.com/shona
6	http://www.viva.org/downloads/pdf/wwp2012/
7	http://faraitose.wordpress.com
8	http://16dayscwgl.rutgers.edu

4.1. Text normalization

RLAT provides a mechanism that cleans the collected raw data by removing HTML tags and punctuation marks and converting the text to lower case. This is termed language independent text normalization [7]. RLAT also provides the capability to perform language dependent text normalization. This process involves the removal of characters not occurring in the target language, digit normalization. and refined punctuation mark removal. The process requires linguistic input from a native speaker of the language.

4.2. Finding and managing English words in the text corpus

In order to evaluate how prominent the language-mixing phenomenon occurs in our data, we counted the words that occur in any of the CMU [8], Lwazi [9] and NCHLT English [5] pronunciation dictionaries. (The numerals were left unchanged, to enable us to hear how native speakers call them out - we have previously found that numeric quantities in Southern African languages are often pronounced in English [10].) Even though the text was crawled from Shona web sites, the data was found to have a large portion of English content: for both word types (i.e. each unique word is counted separately) and word tokens (i.e. each word counted regardless of repetition) the ratio of English to Shona was approximately 50/50. Although some English data would be acceptable for our Shona development process, this ratio is too high - we therefore needed to perform additional processing. To control the amount of English text in our corpus, we removed sentences that contain English words only.

The rejection list consisted of 65 thousand words, mostly in the South African dialect of English. Sentences that had a mix of English and Shona were included in the corpus, since such code-switched speech is commonly found in ASR applications in under-resourced languages. After the complete English sentences were removed, around 14% of the word tokens and 13.4% of the word types are in English, which is a more acceptable starting point for corpus development.

The process described allowed us to collect clean text data from the internet for a typically under-resourced African language. The process was efficient and cost effective. The main motivation for choosing RLAT was the ease of use and less reliance on local internet connectivity when acquiring text data (the Karlsruhe server did, of course, not have connectivity issues). However the data needed a fair amount of post processing due to the amount of the embedded language (English) found in the text. The process managed to collect sufficient text data to generate prompts that were used for recording which is the next stage of our ASR development.

4.3. Prompt design and generation

The process of prompt design is an important step when creating an ASR system: the manner in which prompts are generated can greatly influence the accuracy of the system. Important factors that need to be kept in mind are the domain in which the prompts will be used, acoustic patterns in a language and phonetic coverage of the prompts.

To match the statistics of the target language, the prompts are generally required to cover the most frequently used words in that domain. We achieved this by crawling text data and performing a word frequency count. A Perl script running a greedy algorithm was used to generate prompts. Since the Shona language has a conjunctive writing style (resulting in long words), the prompts were constrained to word tri-grams.

4.4. Prompt verification

Our prompt selection process uses statistical algorithms that do not perform any other analysis. For this reason the prompts had to be verified before the recording process could take place. This is to ensure that they do not contain spelling errors or inappropriate content such as abusive or obscene phrases. For under-resourced languages the luxury of a spell checker to correct the text in the prompts is not available. The verification process under these conditions requires manual verification from linguists or native speakers of the particular language.

After the prompts were generated, the prompt text file was uploaded to the Google App Engine (GAE) through a web interface [1]. The prompts could then be verified on line by selecting a checkbox corresponding to the valid prompts. The selected prompts are saved on the database and ready for download.

5. Speech data collection

The final stage of our data collection efforts was the recording of speech data from several native Shona speakers. For our typically under-resourced conditions, we opted to incorporate a mobile phone application called Woefzela [11] to facilitate the speech data collection process. Woefzela is an open-source tool that runs on the Android operating system. It does not rely on internet connection to perform audio data collection, but does require that text prompts be loaded on the phone manually. The main reason for using the application is due to its open-source nature and ease of use.

Previously verified prompts are downloaded from GAE via an Android application called WDownload. WDownload works in conjunction with Woefzela and loads the downloaded file when the application is activated. However, before the recording process could commence, there were several measures that were taken to ensure that we collected high quality speech data.

5.1. Respondent processing

These measures included respondent canvassing and screening, where a respondent is required to read out fifteen randomly selected Shona sentences in the presence of a language screener. Subsequent to the screening process, respondents have to be registered and sign a consent form to allow their voices to be used for our project.

5.2. Prompt recording

The last step of our speech data collection process was to get respondents to read out 500 randomly selected prompts. The number of prompts to be recorded was later reduced to 300 due to fatigue and loss of concentration. For this process we employed six smart phones running the Android operating system. The phones were running Woefzela as an added application. It provides a practical manner to collect speech data, especially in under-resourced environments.

The recordings could be performed in multiple sessions under quiet conditions. Depending on how quickly the respondent could read the prompts, the recording sessions could take between 45 minutes to an hour. After the recording process was completed, the audio files and the associated meta-data were automatically uploaded to GAE. Through our recording efforts, we managed to collect over 7 hours of speech from 22 speakers, of which 8 were female and 14 were male.

6. Automatic Speech Recognition

In order to evaluate how useful our data is for the purposes of ASR (and to create a basic Shona recognizer for further development), we have carried out several experiments. Most of the

results reported below employ three-fold cross validation, with the test and training folds selected to have no speaker overlap and to ensure that the three test folds have approximately the same duration of speech.

6.1. Pronunciation dictionary

The development of phone-based ASR systems for under-resourced languages normally requires the development of an appropriate pronunciation dictionary (lexicon). Whereas any literate native speaker of the target language can perform the tasks to this point, the development of such a lexicon requires more specialized linguistic knowledge. Since such knowledge is often hard to come by for under-resourced languages, there is a growing awareness that grapheme-based ASR is an attractive alternative for such languages [3]. Since Shona is characterized by a very regular relationship between its written and spoken forms, it is a good candidate for this approach. The dictionary was assembled by representing the pronunciation of a word by its sequence of letters. (A more sophisticated grapheme representation could also be considered - we discuss this possibility in the Conclusion, but in the remainder of this section we use "letter" and "grapheme" interchangeably.) All the words in the word list are acquired from the crawled text corpus.

6.2. Feature extraction

The recogniser employed a standard Hidden Markov Model (HMM) based system. For feature extraction, 39 (13 static, 13 delta and 13 delta-delta) dimensional Mel Frequency Cepstral Coefficient (MFCC) features were generated using HTK [12]. The MFCCs were extracted from a 25 milliseconds frame every 10 milliseconds. Eight Gaussian mixtures per HMM state were incorporated to model the cepstral densities. A flat grapheme-based language model was used for grapheme recognition.

6.3. Experiment 1: English + Shona

This subsection presents the results for the overall corpus. The independent training and test sets contain both English and Shona content. Table 4 shows the overall amount of data used and the accuracy of the grapheme-based three-fold cross validation system with English and Shona-Only.

Table 4: *Overall English + Shona results: grapheme accuracy with a flat language model.*

Language	% Correct	% Accuracy	Data
English + Shona	70.64	59.95	7.7 hours
Shona-only	75.53	66.33	6.5 hours

Table 4 illustrates that the accuracy results improved after English was omitted from the test and training sets. This is due to the fact that English has a highly irregular mapping between graphemes and phonemes. The grapheme-based recognition results for such languages are invariably poor [13] – especially for the case where the majority of the speech data are written in the orthography of another language.

The Shona-Only grapheme-based system also produces accuracies in the range presented by [14]. Although our corpus was limited in size and speaker variability, the grapheme accuracy achieved is therefore acceptable. However, as Table 4 indicates, about 1.2 hours of data was lost after removing the English content, and our recognition of embedded English content was significantly worse than that of native Shona words. These issues can be addressed by performing transliteration on the English data to make it as regular as Shona [3]. The process is accomplished by mapping the phonetic representation of the English words to Shona graphemes ("P-to-G" mapping).
The results that were obtained for the English + Shona data after performing P2G mapping are shown in Table 5. In comparison with Table 4, recognition accuracy for the same amount of data clearly benefits from P2G mapping.

Table 5: *Overall English + Shona results after transliteration of English words*

Language	% Correct	% Accuracy	Data
Transliterated	72.87	61.42	7.7 hours

To further analyse the results from Table 5, we evaluated the transliterated corpus quality by using Phone-based Dynamic Programming (PDP) scores [15]. The scores were obtained by first training a grapheme-based ASR system, then decoding with a phone-loop grammar and also aligning the utterances with the known orthographic transcriptions at a grapheme level. Thereafter, the two grapheme strings corresponding to an utterance are aligned using dynamic programming. Finally, the alignment score obtained is utilised as a measure of both audio and transcription quality [2]. The experiments were conducted using a flat scoring matrix. Figure 1 depicts sorted DP scores per utterance, where a score of 1 indicates a perfect match between the two grapheme strings. The transliteration process has proven to improve the quality of the utterance as shown by the transliterated graph moving above the raw data graph.

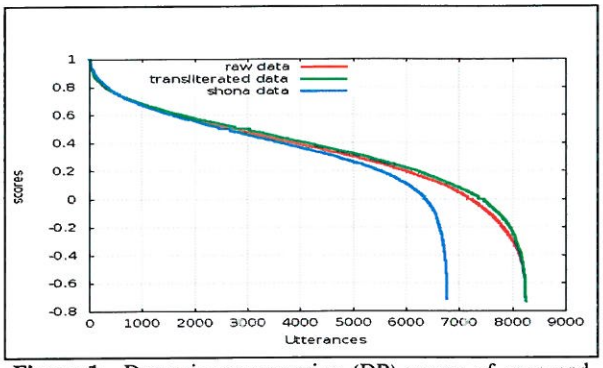

Figure 1: Dynamic programming (DP) scores of pure and mixed Shona utterances

The figure also shows a sharp decline of the Shona graph below the score of zero, which indicates the small number of bad Shona utterances. Comparing the confusion matrices before and after transliteration, we see that the most significant change is a diffuse improvement in the accuracy of acoustic models, leading to a substantial reduction in falsely inserted graphemic units. Utterances containing digits make a small contribution to the total number of errors, but are very poorly recognised (accuracy less than 40% both before and after transliteration). The usual challenge of digit normalisation is aggravated by poorly-understood patterns of code-switching when numbers are read out loud. Further research is required to determine which digit

normalisation approach is most appropriate for the Shona language and whether digit sequences are most often pronounced in English.

7. Conclusion

We have demonstrated the use of an end-to-end system for data collection, consisting of a set of widely-available or open-source tools. We selected an under-resourced language, Shona, for which no speech technology had previously been available, and collected relevant text data via the World Wide Web. The crawling and clean-up processes were accomplished through a web-based tool called RLAT. Although the crawled data contained an embedded language (English) that had equal presence as the matrix data, we used a selection process in order to reach a 86 % to 14% Shona-to-English ratio. The text was used to generate prompts which could be verified on line through a web based interface housed on GAE.

Our ASR system was trained with 7.7 hours of data that contained both English and Shona. The grapheme-based system was proven to be significantly more accurate after the English content was removed from both the test and training sets. However, the presence of English content in Shona speech is a practical reality, which prompted us to explore a mechanism that would permit us to increase the system's accuracy in the presence of such content. We used transliteration to perform a phoneme-to-grapheme mapping of English phones to Shona graphemes. The resulting system was still somewhat less accurate than the Shona-only system, but the observed gain in accuracy indicates that transliteration is a promising approach in this context.

Our approach to transliteration is extremely simple: we simply replace each English phoneme in a standard lexicon with the Shona grapheme that most closely corresponds to it. It would be interesting to see whether a more sophisticated strategy can be used to achieve further gains in accuracy. Similarly, we have treated all letters as though they are distinct graphemes; this is known to be untrue for Shona, and it would be interesting to see whether a linguistically motivated grapheme set will produce improved recognition accuracies. To perform such a comparison would, however, require word recognition, which was not included in the scope of the current research.

The end-to-end system was designed to be as generic as possible and can be employed to develop ASR systems for all languages that have text data on the World Wide Web. RLAT has very little reliance on internet connectivity, which makes it ideal for both well-resourced and under-resourced conditions. Woefzela, which is our open-source software together with inexpensive mobile devices can be used in all environments. However, many under-resourced languages exist in the presence of an often dominant language, which results in a lot of data loss during post-processing. Our method of transliteration, which can be applied to any embedded language, has proved to improve the accuracy of ASR systems without any data loss. However, users require reliable internet connectivity to perform prompt verification and download prompts, which may prove undesirable for some under-resource environments.

8. Acknowledgment

The authors would like to thank Pedro Moreno for proposing the development of an end-to-end ASR collection toolkit. Tim Schlippe, Ngoc Thang Vu, Charl van Heerden, Willem D. Basson, Nic de Vries, Neil Kleynhans, and the HLT Research Group, Meraka Institute, CSIR contributed to this project in various ways. Financial support from a Google Research Award is gratefully acknowledged.

9. References

[1] R. Molapo, E. Barnard, and F. de Wet, "A distributed approach to speech resource collection," in *24th Annual Symposium of the Pattern Recognition Association of South Africa*. PRASA 2013, Dec 2013, pp. 70–75.

[2] T. I. Modipa, M. H. Davel, and F. de Wet, "Implications of Sepedi/English code switching for ASR systems," in *24th Annual Symposium of the Pattern Recognition Association of South Africa*. PRASA 2013, Dec 2013, pp. 64–69.

[3] W. D. Basson and M. H. Davel, "Category-based phoneme-to-grapheme transliteration." in *Proc. INTERSPEECH*, 2013, pp. 1956–1960.

[4] T. Schlippe, S. Ochs, and T. Schultz, "Wiktionary as a source for automatic pronunciation extraction." in *INTERSPEECH*, Makuhari, Japan, Sept 2010, pp. 2290–2293.

[5] N. J. de Vries, M. H. Davel, J. Badenhorst, W. D. Basson, E. Barnard *et al.*, "A smartphone-based ASR data collection tool for under-resourced languages." Elsevier, 2013.

[6] C. Mudzingwa, "Shona morphophonemics: Repair strategies in Karanga and Zezuru," Ph.D. dissertation, University of British Columbia, 2010.

[7] T. Schlippe, C. Zhu, J. Gebhardt, and T. Schultz, "Text normalization based on statistical machine translation and internet user support." in *INTERSPEECH*. Makuhari, Japan: Citeseer, Sept 2010, pp. 1816–1819.

[8] R. Weide, "The CMU pronunciation dictionary, release 0.6." Carnegie Mellon University, 1998.

[9] L. Loots, M. Davel, E. Barnard, and T. Niesler, "Comparing manually-developed and data-driven rules for p2p learning," in *20th Annual Symposium of the Pattern Recognition Association of South Africa*. PRASA 2009, Nov 2009, pp. 35–39.

[10] T. Ndwe, E. Barnard, and M. De Villiers, "Admixture practises in South African languages: Impact on speech-enabled technology design," in *IST-Africa Conference Proceedings*. IEEE, 2011, pp. 1–8.

[11] N. J. De Vries, J. Badenhorst, M. H. Davel, E. Barnard, and A. De Waal, "Woefzela-an open-source platform for ASR data collection in the developing world," in *INTERSPEECH*, Florence, Italy, Aug 2011, pp. 3177–3180.

[12] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The HTK book," vol. 3, 2002, p. 175.

[13] M. Davel and E. Barnard, "Default-and-refinement approach to pronunciation prediction." in *15th Annual Symposium of the Pattern Recognition Association of South Africa*. Grabouw, South Africa: PRASA 2004, Nov 2004, pp. 374–393.

[14] Meraka-Institute. (2009) Lwazi ASR corpus. [Online]. Available: http://www.meraka.org.za/lwazi

[15] M. H. Davel, C. J. van Heerden, and E. Barnard, "Validating smartphonecollected speech corpora," in *3rd workshop on Spoken Languages Technologies for Under-resourced languages*, 2012, pp. 68–75.