Ice-numbers: Experimentation with a spoken numeral recognizer for Icelandic

Hinrik Hafsteinsson
University of Iceland / Reykjavík University
hih43@hi.is

Abstract

This report describes a small-scale experiment in automatic speech recognition for Icelandic.[1] The aim of the project was to create software for practical applications, Ice-numbers, that could understand spoken numerals as input, along with basic mathematical functions. Using the Kaldi toolkit and manually collected speech data, a monophone ASR system was produced that returned a WER of 6% when tested on unfamiliar data. These results were promising, although the practical implementation of the project remains unfinished.

1 Introduction

Automatic speech recognition (ASR) in the context of Icelandic is a burgeoning field. Recent innovations in the production of large spoken corpora (e.g. Guðnason et al., 2012; Steingrímsson et al., 2017; Helgadóttir et al. 2017) have enabled the production of high performance ASR software tailored to the language, e.g. the Reykjavík University (RU) text-to-speech decoder.[2] In the case of the RU text-to-speech decoder, the free ASR toolkit Kaldi[3] (Povey et al., 2011) is used to process the relevant speech data in a state-of-the-art manner and produce the back-end of the software.

The purpose of this project is to test the functionality of Kaldi on a simple dataset in Icelandic; spoken numerals and mathematical functions. The project, which will here be called Ice-numbers, entails editing Kaldi's various scripts and parameters to fit the dataset along with evaluating the results and, if possible, implement the software into a practical demo.

This report is structured as follows. 2 covers the recording and preparation of the speech data used for training and testing. 3 covers the technical aspect of using Kaldi, the methods used in the training process and results produced. 4 concludes.

2 Collection and Preparation of Speech Data

The speech recordings used for training the Ice-numbers system were gathered manually from volunteer participants. In total, 6 volunteers took part, three males and three women, all aged 23-26. Each participant was given an individual ID (kk1-3 for men, kv1-3 for women) and each provided 20 recordings in .wav format, giving 120 utterances in total.

Each utterance consisted of 10 digits (0-9) and five mathematical symbols in random order. An example utterance can be seen in table 1, in the format the participants read.

Filename: kk1	
1.	4 + 6 2 8 7 0 9 = - 5 1 x 3 ÷
2.	5 = 6 1 3 2 7 9 8 4 - ÷ 0 + x
3.	1 = + x 5 4 3 ÷ 9 2 8 7 - 6 0

Table 1: Format of speaker instructions

For standardization purposes, participants were given specific instructions on how to pronounce each number and mathematical symbol, as Icelandic allows variation in gender, number and case in numerals depending on context. Thus, numbers were always to be read in *masculine* gender, *nominative* case and *singular* number. For the mathematical symbols, participants were asked to use only the word forms provided in table 2.

[1] This report is part of the open project assignment in the course Automatic Speech Recognition (T-718-ATSR), taught by Jón Guðnason, PhD, and Michal Borský, PhD, at Reykjavík University during the spring semester of 2019.

[2] https://tal.ru.is

[3] http://kaldi-asr.org

Symbol		Pronunciation
+	→	„plús“
-	→	„mínus“
x	→	„sinnum“
÷	→	„deilt með“
=	→	„samasem“

Table 2: Standardized pronunciation for mathematical symbols

Each participant recorded his utterances on his/her own, with segmentation being done manually by the researcher. In preparation of using Kaldi for processing the recordings, each .wav was renamed according to the respective speaker's ID and utterance context, separated by underscores as shown in table 3.

Participant: kk1
kk1_+_6_÷_=_5_9_4_2_3_1_0_-_x_8_7.wav
kk1_+_8_x_0_4_2_3_-_1_9_÷_5_=_7_6.wav
kk1_-_0_x_6_8_+_2_4_3_1_7_9_5_÷_=.wav
kk1_-_2_6_=_9_5_8_0_4_x_3_1_+_7_÷.wav

Table 3: Filenames ready for use in kaldi

3 Methods and results

The Kaldi toolkit contains both the code for its own processes and various example projects to run with the toolkit, most of which are based on existing Corpora. One of these, based on the "Yesno" dataset[4] is a popular starting point for those learning to use Kaldi. Because its project's simplicity and functionality[5], it was used as a template for the Ice-numbers project. The scripts and basic work flow of the Yesno project were kept mostly unchanged, unless the need for modification arose.

3.1 Language data - Lexicon

Kaldi requires an input lexicon that maps sequences of phones to words. Larger projects, that e.g. use Kaldi to map whole languages, need immense datasets of phonetic transcriptions in their lexicon to achieve their goal. However, our "language" (the contents of the recorded speech data) only has 15 "words" in total. Thus a small set of 25 "phones" was chosen to represent the words in the whole lexicon. These phones and their IPA[6] representations are listed in table 4. The phones were arbitrarily approximated from Icelandic phonemes, with variables in pronunciation such as length ignored. As well as the lexicon described above, a variant lexicon was produced which contains no segmentation between phones, i.e. each word is its own phone. This was based on a similar method applied to the Yesno corpus and was mainly done for comparison's sake. This will be touched upon in section 3.4.

Kaldi	Phone	Kaldi	Phone	Kaldi	Phone
A	a [a]	D	d [t]	P	p [p^h]
I	i [ɪ]	DH	ð [ð]	R	[r]
U	u [ʏ]	F	f [f]	S	s [s]
E	e [ɛ]	H	h [h]	T	t [t^h]
O	o [ʊ]	J	j [j]	V	v [v]
AU	á [au]	K	k [k^h]		
OE	ö [œ]	L	l [l]		
EI	ei [ei]	M	m [m]		
UU	ú [u]	N	n [m]		

Table 4: Phones designated for the Kaldi lexicon and their IPA approximations

3.2 Training and testing

For comparison's sake, this project used two different methods of splitting the testing material from the training material, listed in Table 5. The first approach aimed to ensure that no speaker in the training set would appear in the testing set. The simplest way to do accomplish this is to isolate one speaker from the rest. As the total number of speakers is only 6, one speaker accounts to a substantial amount of the total recordings. By default the test speaker is "kk1" but this can be changed at will.

The second setup used 10% of the dataset (2 recordings from each speaker) to create the test set. Such a setup introduces speaker contamination but serves as a good comparison during evaluation of the system.

3.3 Utterance information

Aside from the language input described in section 3.1, Kaldi requires various data on the utterances being processed, most importantly transcriptions

[4] http://www.openslr.org/1/

[5] The Yesno example project creates a monophone acoustic model, which should suit the numerals being looked at here well.

[6] International phonetic alphabet. See: https://www.internationalphoneticassociation.org.

	Training	Testing	Note
Setup 1	100	20	Speaker specific
Setup 2	108	12	Intra-speaker

Table 5: Number of files in training/testing setups

and segmentation information on the audio recordings. For bigger projects this might be supplied in it's own designated file or other formats. For this project all the relevant information about each recording is stored in its file name, which, like each utterance, is unique in the dataset. This step of the project saw the heaviest editing of the existing Kaldi scripts.

In ASR literature, a complete ASR system like the one Kaldi creates is referred to as a finite state transducer (FST). With the information on word content in individual utterances, Kaldi puts together a grammar for the input language in the form of the FST "G". Kaldi uses this word-level grammar information in tandem with a FST generated from the lexicon information, "L", to create a bigger FST, "LG", which essentially calculates both word and grammar information.

The rest of the information Kaldi needs to create a complete ASR system comes down to phone contexts ("C") and the acoustic model ("H"). As Kaldi makes a monophone model, the phone context is irrelevant in this project. The acoustic model is calculated through scripts with the mel-frequency cepstrum coefficients (MFCC) of the audio recordings. Finally Kaldi calculates all these FSTs together into a final FST, "HCFG", made up of 4 sup-parts which all play a role in the ASR system.

3.4 Results

As described in section 5 the testing setup was twofold. *Setup 1* returned a consistent word error rate (WER) of $\approx 6\%$. This is relatively good, but at first glance might be improved on given such a small lexicon size. Interestingly the results from setup 2 showed little variation in WER from setup 1, giving WER of $\approx 6.5\%$.

The "variant lexicon" described in section 3.1, where each word has only one corresponding phone, was also tested for comparison, both with train/test setup 1 and setup 2. Here WER deteriorated considerably, with setup 1 giving a 30% WER and setup 2 giving a 19.35% WER.

From these results it can be inferred that (a) even thought it may be small, a one-phone-per-word lexicon does not suit this dataset and (b) speaker cross-contamination in the training and testing sets does not seem to have a predictable effect on WER in this dataset.

4 Discussion

This section discusses various points in the project that might be improved or changed in further development.

Initial preparation of the project assumed the total number of speakers would be as big as 8, with 160 individual recordings. This was scaled down to 6 speakers to avoid unnecessary complexity in the project but might have been beneficiary for the project results in the long run.

The lexicon described in section 3.1 might also be improved on. As mentioned, the choice of phoneme representation in the phones was arbitrary, with no real effort being put into testing different setups (other than the single phone per word model). Here there is much room for testing, e.g. seeing if representing diphthongs (vowels with two phonetic values) as two phones instead of one would improve the results. This would make the total number of distinct phones smaller which has an effect on Kaldi's calculations.

Finally, the splitting of the training and testing data may be improved on. Although separating one speaker from the rest is a good way to define testing material, a better solution would be to test on every individual speaker after training on the other speakers in an automatic fashion. The average of the WER from these testings would give a reliable estimation on the accuracy of the ASR system. This was not implemented in the project and although it could have been done manually, the idea was dropped due to time constraints.

This project covered the various steps needed to use the Kaldi toolkit on a custom dataset. In this sense the project was a success. Unfortunately due to technical limitations and time constraints the final step of using the ASR system created with Kaldi for practical applications remains unfinished.

References

Jón Guðnason, Oddur Kjartansson, Jökull Jóhannsson, Elín Carstensdóttir, Hannes Högni Vilhjálmsson, Hrafn Loftsson, Sigrún Helgadóttir, Kristín M Jóhannsdóttir, and Eiríkur Rögnvaldsson. 2012. Almannaromur: An open icelandic speech corpus. In *Spoken Language Technologies for Under-Resourced Languages*.

Inga Rún Helgadóttir, Róbert Kjaran, Anna Björk Nikulásdóttir, and Jón Guðnason. 2017. Building an asr corpus using althingi's parliamentary speeches. In *Proc. Interspeech 2017*, pages 2163–2167.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December. IEEE Catalog No.: CFP11SRW-USB.

Steinþór Steingrímsson, Jón Guðnason, Sigrún Helgadóttir, and Eiríkur Rögnvaldsson. 2017. Málrómur: A manually verified corpus of recorded icelandic speech. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 237–240.