IDIAP RESEARCH REPORT

A New Speech Recognition Baseline System for Numbers 95 Version 1.3 Based on Torch

Johnny Mariéthoz [1] Samy Bengio [2]

IDIAP–RR 04-16

April 15, 2004

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

1 IDIAP, CP 592, 1920 Martigny, Switzerland, marietho@idiap.ch
2 IDIAP, CP 592, 1920 Martigny, Switzerland, bengio@idiap.ch

IDIAP Research Report 04-16

A New Speech Recognition Baseline System for Numbers 95 Version 1.3 Based on Torch

Johnny Mariéthoz   Samy Bengio

April 15, 2004

Abstract. This report describes a complete baseline system for the last version (1.3) of the Numbers95 database for speech recognition. The goal of this report is to provide a speech recognition system for all researchers at IDIAP which performance corresponds to the state-of-the-art, and from which everyone is encouraged to improve using their own research idea.

Contents

1 Introduction

Since several years, several researchers at IDIAP have used the Number95 database for their research on speech recognition. The version most often used is in general quite old (1.0) as compared to newer (and larger) version available. A new release of this database does exist and is available at IDIAP, but have never been used for experimental research. This new version contains about four times more sentences. In this report we propose to re-create a new baseline system on the new version of the Numbers95 databases using the Torch library.

This report is organized as follows. In section 2, we give motivations to re-do a baseline system at IDIAP. In section 3 the new version of the database is described. The system and the installation procedure is given in section 4. The complete experiments using the Numbers95 database are given in section 5. In section 6, we present some conclusions.

2 Speech Recognition System at IDIAP

A speech recognition system tries to obtain the correct transcription from a sentence pronounced by a speaker. State-of-the-art systems are based on Hidden Markov Models (HMMs), either using Gaussian Mixture Models (GMMs) or Multi-Layer Perceptrons (MLPs) for emission distributions.

When a researcher wants to propose a new model or approach, he normally needs to compare his system to a baseline in order to show why his approach is so good. Comparing systems is not always easy as we normally need to keep invariant most of variables such as data, protocols, etc...

The actual situation at IDIAP is that each researcher uses his own version of the database, the protocol and the software. In order to save time for new comers and to make the results more comparable between researchers work, we would like to propose to the speech group, a common baseline system on the latest version of the Numbers95 database.

Since more and more people use the Torch library, this baseline has been designed using this C++ library, together with the use of some perl scripts in order to perform speech recognition.

3 Numbers95 Database Version 1.3

This database contains sentences of several words, essentially numbers. We have retain the sentences containing only the 30 most frequent words and removed the sentences containing truncated words. After removing these files and following the protocol given by the database (modulo 5 rule), we have 10441 sentences for training, 3582 sentences for the validation set and 3621 sentences for the test set. Table 1 gives more information on this protocol, detailing the number of sentences, words, phonemes and frames for each set of data.

set	#sentences	#words	#phonemes	#frames
train	10441	50358	215963	2086138
valid	3582	17597	75507	728030
test	3621	17835	76281	693096

Table 1: Statistics of Numbers95 version 1.3

4 Baseline System Description

This system is based on HMM models. The grammar is the simplest: every words can follow all the other words. The models are trained using the Viterbi training algorithm (which can be seen as a hard case of the EM algorithm).

In a first step, monophone models are trained, based on a linear segmentation of the training set, using Table 3 to relate words and monophones. Then a forced alignment is performed on the training set and the resulting segmentation is used to initialize triphone models (using Table 2 to relate words and triphones).

4.1 Installation

First the *CVS* version of Torch should be installed in your account[1] including the *speech* package. You should then update your *CVS* version of Torch as often as possible to make sure you are always using the latest version:

```
cvs update -P -d
```

You also need to install new *Signal Processing* package, used to compute MFCC features:

```
cvs -d:ext:${USER}@cvs.idiap.ch:/home/learning/norman/cvs checkout torch_sigpro
```

Of course, you also need to add it to your Torch configuration file. Finally install the Numbers95 package:

```
cvs -d:ext:${USER}@cvs.idiap.ch:/home/learning/marietho/cvs checkout speech_n95
```

You can now compile the C++ *mains* used to perform speech recognition; go to the `src` directory, and compile the following files:

```
make speech_hmm_simple_decode speech_hmm_train speech_hmm_init cepstrum amat2htk
or
xmake speech_hmm_simple_decode speech_hmm_train speech_hmm_init cepstrum amat2htk
```

if you are using the xmake package.

4.2 Tools and Scripts

If all programs and scripts are installed, you should have severals directories:

scripts: contains all perl scripts to perform speech recognition:

- **speech_reco:** this script is used to perform a complete speech recognition experiment.
- **wav2torch:** this script is used to compute MFCC features from wav files.
- **mono2tri:** this script is used to convert monophone to triphones transcriptions.

config_files: contains all ASCII configuration files.

- **dict_mono, dict_tri:** corresponding phoneme transcriptions for each word (see Table 3 and 2).
- **phonemes_mono, phonemes_tri:** list of all phonemes used in the database.
- **train:** list of files used to train the models.
- **valid:** list of files used to select the hyper-parameters.
- **test:** list of files used to estimate the expected performance.
- **sil.wav:** audio file containing some silence that can be put at the beginning of each access to ensure a minimum length for each access to suit the size of the corresponding HMM models.

src: contains the Torch3 main programs:

- **speech_hmm_init.cc:** this program is used to initialize simple speech recognition models.
- **speech_hmm_train.cc:** this program is used to train simple speech recognition models.
- **speech_hmm_simple_decode.cc:** this program is used to decode test speech sentences.

[1] see: `/home/learning/common/torch3_cvs_version`.

5 Experiments

All the experiments described here have followed the same methodology. First, the original waveforms were sampled every 12ms and then parameterized into 12 MFCC coefficients and their first and second derivative, as well as the energy together with its first and second derivative, for a total of 39 features. Cepstral Mean Subtraction is performed as well. 100ms of silence can be added at the beginning and at the end of each file using an special option of wav2torch script. To create feature files use:

```
perl ./scripts/wav2torch /com/databases/numbers_v1.3/cds/speech data/features
```

In a first step, monophone models are trained with 3 emitting states and 10 Gaussians per state on the whole training dataset. A forced alignment is then performed on the training set:

```
perl ./scripts/speech_reco  --test --valid --col -phoneme mono --force data/mono_align
     --no -word-target /com/databases/numbers_v1.3/cds/htk_trans results_mono
```

The resulting transcriptions are converted into triphone transcription:

```
perl ./scripts/mono2tri data/mono_align data/mono2tri
```

New triphone models are trained using this provided segmentation. Again each triphone model has 3 emitting states and 10 Gaussians per state:

```
perl ./scripts/speech_reco --test --valid --col --phone-tar data/mono2tri
     -word-target /com/databases/numbers_v1.3/cds/htk_trans results_tri
```

Note that you can run a script containing all the previous commands:

```
bash scripts/run_baseline
```

5.1 Results on Numbers95 Database

Table 4 shows the results, given in terms of *WRR* (Word Recognition Rate) for both validation and test sets, for protocol version 1.0 (the one used by most people at IDIAP) and 1.3 (the new and extended protocol):

	Version 1.0		Version1.3	
	Validation	Test	Validation	Test
Mono+sil	91.55	91.80	91.87	92.48
Mono	92.07	91.09	91.39	91.9
Tri+sil	94.89	94.54	94.9	94.94
Tri	94.22	94.20	94.67	94.92

Table 4: Results on Numbers95 databases

In order to compare the results, Table 4 also gives the results on version 1.0 of the database. The obtained results show that the baseline system performs as well as state-of-the-art speech recognition system, for both versions of the database.

6 Conclusion

We have proposed a new baseline system on the latest version of Numbers95. This includes Torch3 programs, configuration files and scripts. This is a baseline system and can (should!) be improved by researcher. It would probably be a good idea to keep the best results obtained on this database (and the corresponding methodology) throught the use of a web page maintained by the speech group.

Words	Triphones
sil	h#
<s>	h#
</s>	h#
eight	ey+tcl ey-tcl+t tcl-t h#
eighteen	ey+tcl ey-tcl+t tcl-t+iy t-iy+n iy-n h#
eighty	ey+tcl ey-tcl+t tcl-t+iy t-iy h#
eleven	ih+l ih-l+eh l-eh+v eh-v+ah v-ah+n ah-n h#
fifteen	f+ih f-ih+f ih-f+tcl f-tcl+t tcl-t+iy t-iy+n iy-n h#
fifty	f+ih f-ih+f ih-f+tcl f-tcl+t tcl-t+iy t-iy h#
five	f+ay f-ay+v ay-v h#
forty	f+ao f-ao+r ao-r+tcl r-tcl+t tcl-t+iy t-iy h#
four	f+ao f-ao+r ao-r h#
fourteen	f+ao f-ao+r ao-r+tcl r-tcl+t tcl-t+iy t-iy+n iy-n h#
hundred	hh+ah hh-ah+n ah-n+dcl n-dcl+d dcl-d+r d-r+ah r-ah+dcl ah-dcl+d dcl-d h#
nine	n+ay n-ay+n ay-n h#
nineteen	n+ay n-ay+n ay-n+tcl n-tcl+t tcl-t+iy t-iy+n iy-n h#
ninety	n+ay n-ay+n ay-n+tcl n-tcl+t tcl-t+iy t-iy h#
oh	ow h#
one	w+ah w-ah+n ah-n h#
seven	s+eh s-eh+v eh-v+ah v-ah+n ah-n h#
seventeen	s+eh s-eh+v eh-v+ah v-ah+n ah-n+tcl n-tcl+t tcl-t+iy t-iy+n iy-n h#
seventy	s+eh s-eh+v eh-v+ah v-ah+n ah-n+tcl n-tcl+t tcl-t+iy t-iy h#
six	s+ih s-ih+kcl ih-kcl+k kcl-k+s k-s h#
sixteen	s+ih s-ih+kcl ih-kcl+k kcl-k+s k-s+tcl s-tcl+t tcl-t+iy t-iy+n iy-n h#
sixty	s+ih s-ih+kcl ih-kcl+k kcl-k+s k-s+tcl s-tcl+t tcl-t+iy t-iy h#
ten	tcl+t tcl-t+eh t-eh+n eh-n h#
thirteen	th+er th-er+tcl er-tcl+t tcl-t+iy t-iy+n iy-n h#
thirty	th+er th-er+dcl er-dcl+d dcl-d+iy d-iy h#
three	th+r th-r+iy r-iy h#
twelve	tcl+t tcl-t+w t-w+eh w-eh+l eh-l+v l-v h#
twenty	tcl+t tcl-t+w t-w+eh w-eh+n eh-n+tcl n-tcl+t tcl-t+iy t-iy h#
two	tcl+t tcl-t+uw t-uw h#
zero	z+ih z-ih+r ih-r+ow r-ow h#

Table 2: Triphones transcription

Words	Monophones
sil	h#
<s>	h#
</s>	h#
eight	ey tcl t h#
eighteen	ey tcl t iy n h#
eighty	ey tcl t iy h#
eleven	ih l eh v ah n h#
fifteen	f ih f tcl t iy n h#
fifty	f ih f tcl t iy h#
five	f ay v h#
forty	f ao r tcl t iy h#
four	f ao r h#
fourteen	f ao r tcl t iy n h#
hundred	hh ah n dcl d r ah dcl d h#
nine	n ay n h#
nineteen	n ay n tcl t iy n h#
ninety	n ay n tcl t iy h#
oh	ow h#
one	w ah n h#
seven	s eh v ah n h#
seventeen	s eh v ah n tcl t iy n h#
seventy	s eh v ah n tcl t iy h#
six	s ih kcl k s h#
sixteen	s ih kcl k s tcl t iy n h#
sixty	s ih kcl k s tcl t iy h#
ten	tcl t eh n h#
thirteen	th er tcl t iy n h#
thirty	th er dcl d iy h#
three	th r iy h#
twelve	tcl t w eh l v h#
twenty	tcl t w eh n tcl t iy h#
two	tcl t uw h#
zero	z ih r ow h#

Table 3: Monophone transcription