Semi-supervised Acoustic Modelling for Five-lingual Code-switched ASR using Automatically-segmented Soap Opera Speech

Nick Wilkinson[1], Astik Biswas[1], Emre Yılmaz[2], Febe de Wet[1], Ewald van der Westhuizen[1], Thomas Niesler[1]

[1] Department of Electrical and Electronic Engineering, Stellenbosch University, South Africa
[2] Department of Electrical and Computer Engineering, National University of Singapore, Singapore
{nwilkinson, abiswas, fdw, ewaldvdw, trn}@sun.ac.za, emre@nus.edu.sg

Abstract

This paper considers the impact of automatic segmentation on the fully-automatic, semi-supervised training of automatic speech recognition (ASR) systems for five-lingual code-switched (CS) speech. Four automatic segmentation techniques were evaluated in terms of the recognition performance of an ASR system trained on the resulting segments in a semi-supervised manner. The systems' output was compared with the recognition rates achieved by a semi-supervised system trained on manually assigned segments. Three of the automatic techniques use a newly proposed convolutional neural network (CNN) model for framewise classification, and include a novel form of HMM smoothing of the CNN outputs. Automatic segmentation was applied in combination with automatic speaker diarization. The best-performing segmentation technique was also tested without speaker diarization. An evaluation based on 248 unsegmented soap opera episodes indicated that voice activity detection (VAD) based on a CNN followed by Gaussian mixture model-hidden Markov model smoothing (CNN-GMM-HMM) yields the best ASR performance. The semi-supervised system trained with the resulting segments achieved an overall WER improvement of 1.1% absolute over the system trained with manually created segments. Furthermore, we found that system performance improved even further when the automatic segmentation was used in conjunction with speaker diarization.

Keywords: code-switched speech, under-resourced languages, automatic segmentation, semi-supervised training

1. Introduction

Code-switching is the alternation between two or more languages by a single speaker during discourse, and is a common phenomenon in multilingual societies. In South Africa, for example, 11 official and geographically co-located languages are in use, including English which serves as the lingua franca. Here, speakers frequently code-switch between English, a highly-resourced language, and their Bantu mother tongue, which is in comparison highly under-resourced.

The automatic recognition of code-switched speech has become a topic of growing research interest, as reflected by the increasing number of language pairs that have recently been studied. While English-Mandarin has received extensive attention (Li and Fung, 2013; Zeng et al., 2018; Vu et al., 2012; Taneja et al., 2019), other language pairs such as Frisian-Dutch (Yılmaz et al., 2016; Yılmaz et al., 2018), Hindi-English (Pandey et al., 2018; Emond et al., 2018; Ganji et al., 2019), English-Malay (Ahmed and Tan, 2012), Japanese-English (Nakayama et al., 2018) and French-Arabic (Amazouz et al., 2017) have also attracted interest. We have introduced the first South African corpus of multilingual code-switched soap opera speech in (van der Westhuizen and Niesler, 2018).

For code-switched speech, the development of robust acoustic and language models that are able to extend across language switches is a challenging task. When one or more of the languages are under-resourced, as it is in our case, data sparsity limits modelling capacity and this challenge is amplified. Acoustic data that includes code-switching is extremely hard to find, because it usually does not occur in formal conversation, such as broadcast news, and also because it requires skilled multilingual language practitioners for its annotation. The result is that manually-prepared datasets including code-switched speech in Africa are destined to remain rare and small.

In previous work, we have demonstrated that multilingual training using in-domain soap opera code-switched speech and poorly matched monolingual South African speech improves the performance of both bilingual and five-lingual automatic speech recognition (ASR) systems when the additional training data is from a closely-related language (Biswas et al., 2018a; Biswas et al., 2018b). Specifically, isiZulu, isiXhosa, Sesotho and Setswana belong to the same Bantu language family and were found to complement each other when combined into a multilingual training set for acoustic modelling. Hence, increasing the amount of in-domain code-switched speech data is a reliable way to achieve more robust ASR. However, the development of such in-domain data is a time-consuming and costly endeavour as it requires highly skilled human annotators and transcribers.

To address this lack of annotated data, automatically transcribed training material has been shown to be useful in under-resourced scenarios using semi-supervised training (Thomas et al., 2013; Yılmaz et al., 2018; Guo et al., 2018). This strategy was successfully implemented on South African code-switched speech to obtain bilingual and five-lingual ASR systems using 11.5 hours of manually segmented but untranscribed soap opera speech (Biswas et al., 2019). Recently a study has analyzed the performance of batch-wise semi-supervised training on South African code-switched ASR (Biswas et al., 2020). However, manual segmentation of the raw soap opera audio by skilled annotators was still required to identify the speech that is useful for ASR. Therefore, this approach is not fully automatic which remains an impediment in resource-scare settings.

In this study, we apply four automated approaches to the

segmentation of soap opera speech and investigate the effect on ASR performance. A conventional energy-based voiced activity detector (VAD) (Povey et al., 2011), as well as CNN-HHM and CNN-GMM-HMM systems that we have developed are used to distinguish between speech, music and noise. In addition, an X-vector DNN embedding system is used for speaker diarization to obtain speaker specific metadata for three of the segmentation approaches (Snyder et al., 2018). For the experiments, 248 complete soap opera episodes, each approximately 22 minutes in length, were used. It is important to note that we also have the manual segmentation (approximately 24 hours of speech) of these 248 episodes and can therefore perform a comparative evaluation with the automated approaches. Semi-supervised systems trained using the manually-segmented speech were used as baselines and compared with systems trained on speech identified by the automatic approaches.

Pseudo-labels or transcriptions of automatically segmented speech were generated using our best baseline systems trained on 21 hours manually transcribed speech and 11 hours of manually segmented but automatically transcribed speech. Given the multilingual nature of the data, the transcription systems must not only provide the orthography, but also the language(s) present at each location in each segment. To achieve this, each segment was presented to four individual code-switching systems as well as to a five-lingual system.

2. Data

For experimentation, we use a corpus of multilingual, code-switched speech compiled from South African soap opera episodes. This corpus contains both manually and automatically-annotated speech divided into four language pairs: English-isiZulu (EZ), English-isiXhosa (EX), English-Setswana (ET), and English-Sesotho (ES). Of the Bantu languages, isiZulu and isiXhosa belong to the Nguni language family while Setswana and Sesotho are Sotho-Tswana languages.

The corpus contains 8 275, 11 352, 6 169, 1 902 and 2 792 unique English, isiZulu, isiXhosa, Setswana and Sesotho words, respectively. IsiZulu and isiXhosa have relatively large vocabularies due their agglutinative nature and conjunctive writing system. Although Setswana and Sesotho are also agglutinative, they use disjunctive writing systems which result in smaller vocabularies than isiZulu and isiXhosa. The speech in the soap opera episodes is also typically fast and often expresses emotion. These aspects of the data in combination with the high prevalence of code-switching makes it a challenging corpus for conducting ASR experiments.

2.1. Manually Segmented and Transcribed Data (ManT)

Our first code-switching ASR systems were developed and evaluated on 14.3 hours of speech divided into four language-balanced sets, as described in (van der Westhuizen and Niesler, 2018). In addition to the language-balanced sets, approximately another nine hours of manually transcribed speech was available. This additional data

is dominated by English and was initially excluded from our training set to avoid bias. However, pilot experiments indicated that, counter to expectations, its inclusion enhanced recognition performance in all languages. The additional data was therefore merged with the balanced sets for the experiments described here. Of this, 21.1 hours is used as a training set, 48 minutes as a development set, and 1.3 hours as a test set. The composition of the unbalanced training set is shown in Table 1.

Language	Mono (m)	CS (m)	Total (h)	Total (%)	Word tokens	Word types
English	755.0	121.8	14.6	69.3	194 426	7 908
isiZulu	92.8	57.4	2.5	11.9	24 412	6 789
isiXhosa	65.1	23.8	1.5	7.0	13 825	5 630
Setswana	36.9	34.5	1.2	5.6	21 409	1 525
Sesotho	44.7	34.0	1.3	6.2	22 226	2 321
Total	994.5	271.5	21.1	100.0	276 290	24 170

Table 1: Duration in minutes (m) and hours (h) as well as word type and token counts for the unbalanced training set.

An overview of the composition of the development (Dev) and test (Test) sets for each language pair is given in Table 2. The table includes values for the total duration as well as the duration of the monolingual and code-switched segments. The test sets contain no monolingual data and a total of approximately 4 000 language switches (English-to-Bantu and Bantu-to-English).

	English-isiZulu				
	emdur	zmdur	ecdur	zcdur	**Total**
Dev	0.0	0.0	4.0	4.0	8.0
Test	0.0	0.0	12.8	17.9	30.4
	English-isiXhosa				
	emdur	xmdur	ecdur	xcdur	**Total**
Dev	2.9	6.5	2.2	2.1	13.7
Test	0.0	0.0	5.6	8.8	14.3
	English-Setswana				
	emdur	tmdur	ecdur	tcdur	**Total**
Dev	0.8	4.3	4.5	4.3	13.8
Test	0.0	0.0	8.9	9.0	17.8
	English-Sesotho				
	emdur	smdur	ecdur	scdur	**Total**
Dev	1.1	5.1	3.0	3.6	12.8
Test	0.0	0.0	7.8	7.7	15.5

Table 2: Duration (minutes) of English, isiZulu, isiXhosa, Sesotho, Setswana monolingual (mdur) and code-switched (cdur) segments in the code-switching development and test sets.

2.2. Manually Segmented Automatically Transcribed Data: Expert Segmentation (AutoT$_{Exp}$)

During corpus development, approximately 11 hours of manually segmented speech (representing 127 different speakers) was produced in addition to the manually transcribed data described in the previous section. Segmentation was performed manually by experienced language

practitioners. This dataset (AutoT$_{Exp}$) was automatically transcribed during our initial investigations into semi-supervised acoustic model training, resulting in 7 951 EZ, 3 796 EX, 11 415 ES and 128 ET segments (Biswas et al., 2019).

2.3. Manually Segmented Automatically Transcribed Data: Non-expert Segmentation (AutoT$_{NonE}$)

A subsequent phase of corpus development, currently still underway, has produced manual segmentations for a further 248 soap opera episodes. These 248 episodes amount to 89 hours of audio data before segmentation, and 23 hours of speech data (AutoT$_{NonE}$) after segmentation. The segmentation was not performed by language experts and is therefore expected to be less accurate than that of the AutoT$_{Exp}$ data. Furthermore South African languages other than the five present in the transcribed data are known to occur in this batch, but to a limited extent.

This set of 248 episodes was used in the automatic segmentation experiments described in the next section because the manually assigned segment labels were available as a reference in the form of AutoT$_{NonE}$.

3. Automatic Segmentation

A number of automatic segmentation techniques were considered as alternatives to the labour-intensive process of manually segmenting the soap operas. Different voice activity detection (VAD) approaches were combined with the X-vector DNN embedding-based speaker diarization system introduced in (Snyder et al., 2018) to obtain speaker labels. In subsequent ASR experiments, the best performing VAD technique was also evaluated without speaker diarization.

3.1. VAD$_1$: Energy-based

In our first experiment, the X-vector diarization recipe provided in the Kaldi toolkit was applied using an X-vector DNN model pre-trained on wide-band VoxCeleb data (Povey et al., 2011; Nagrani et al., 2017; Chung et al., 2018). This system uses 24-dimensional filterbank features based on 25ms frames. Speech frames are identified using a simple energy threshold and are subsequently passed to the pre-trained DNN which extracts the X-vectors. Finally, probabilistic linear discriminant analysis (PLDA) is applied to the X-vectors, and agglomerative hierarchical clustering is used to assign speaker labels.

A difficulty observed when using this approach was that, while a simple energy VAD works reasonably well under low noise conditions where most frames are speech, it performs poorly when confronted with our soap opera data in which extensive non-speech segments containing music and other sounds are common. Post-diarization listening tests revealed that many non-speech segments were still present in the data classified as speech. Adjustment of the VAD threshold to more aggressively remove non-speech segments resulted in the loss of many speech segments.

3.2. VAD$_2$: CNN-HMM

At the time of writing, the X-vector based system achieved state-of-the-art performance in diarization tasks. However,

Layer	Kernels/Nodes	Activation Function
Convolutional_1	32 (3x3 kernel)	ReLU
Max_Pooling_2	-	-
Convolutional_3	64 (3x3 kernel)	ReLU
Max_Pooling_4	-	-
Convolutional_5	64 (3x3 kernel)	ReLU
Flatten_6	1024	-
Fully_Connected_7	64	ReLU
Fully_Connected_8	2	Sigmoid

Table 3: The CNN architecture used in the VAD systems.

the energy based VAD it uses limits performance. For this reason, efforts to improve automatic segmentation focused on developing improved VAD. Recently, CNNs have been successfully applied to the task of VAD (Thomas et al., 2014), and both large and small architectures have been found to perform well (Sehgal and Kehtarnavaz, 2018; Hershey et al., 2017). In our resource-constrained setting, computational efficiency is important since VAD will most likely occur on a mobile device.

We introduce a small CNN architecture ($\approx$120 000 parameters) implemented in Python, using Tensorflow (v2.0.0) and Keras (v2.2.4-tf), to create a fast, lightweight VAD system whose architecture is shown in Table 3. This system computes 32-dimensional log-mel filterbank energies using a frame length of 10ms and then stacks these over 320ms to form 32x32 spectrogram features as input to the CNN. The CNN was trained on the balanced subset ($\approx$ 53 hours) of Audio Set (Gemmeke et al., 2017) to classify frames as containing "speech" and/or "non-speech".

While our CNN on its own performs well at the VAD task, it fails to capture temporal patterns in the data and was observed to often mislabel single frames within extended sections of speech or non-speech. In an initial attempt to address this, we introduce a HMM for smoothing. The AVA-Speech dataset (Chaudhuri et al., 2018) was used to train our HMMs, and for testing of the final VAD. The full dataset contains $\approx$ 46 hours of densely labeled, multilingual movie data, with the following class labels: "NoSpeech","CleanSpeech", "Speech+Music" and "Speech+Noise". AVA-Speech (train), a randomly selected $\approx$ 23 hour subset of the dataset, was used to train the HMM. Table 4 provides a description of the dataset used for training and testing of the automatic segmentation systems.

For training the "CleanSpeech", "Speech+Music" and "Speech+Noise" classes were treated as a single "speech" class. A two state HMM was defined, with states representing ground truth "speech" and "no-speech" labels respectively. The HMM observations are the binary output of the "speech" neuron in the CNN, which indicates "speech" or "no-speech". Note, the "no-speech" label differs slightly from the "non-speech" label, since the "speech" and "non-speech" sounds can co-occur, whereas "no-speech" implies "speech" does not occur in the signal.

Transition and emission probabilities were trained in a supervised manner, by passing AVA-Speech (train) though the CNN, then using the labels predicted by the CNN and corresponding ground truth labels as observations and hid-

Dataset	Size (hours)	Use
Audio Set	≈ 53	CNN training
AVA-Speech (train)	≈ 23	HMM/GMM-HMM training
AVA-Speech (test)	≈ 23	VAD testing

Table 4: Datasets used for training and testing of automatic segmentation systems.

den state sequences respectively. Viterbi decoding was then used to find the most likely underlying label sequence, given CNN predicted labels. Finally the VAD segments are used as input to the X-vector diarization system.

3.3. VAD$_3$: CNN-GMM-HMM

While the CNN-HMM approach yields a large improvement over the energy-based VAD, it may be possible to improve it further by making use of the CNN soft label outputs, rather than the hard labels obtained by taking the argmax of the CNN outputs. In this case the HMM observation sequence is chosen to consist of the output probabilities computed by the CNN speech neuron, rather than the binary labels. Where the observations were previously modelled as repeated Bernoulli trials, they are now continuous and can therefore be modelled by a more complex distribution function. A 3-mixture GMM for each of the two HMM states was found to be an effective choice. Fewer mixtures led to deteriorated performance, while more mixtures did not result in further improvement. As before, the GMM-HMM is trained on AVA-Speech (train) in a supervised manner and the resulting segments used as input for the X-vector diarization system.

3.4. VAD$_4$: VAD$_3$ Without Speaker Diarization

While the X-vector diarization system is useful for obtaining speaker labels for each segment, it is computationally expensive and represents only a pre-processing step for downstream ASR. To determine its importance, our final experiment used the segments produced by our best performing VAD system directly, without diarization. Hence each segment was treated as being from a different speaker.

4. Automatic Transcription

Recent studies demonstrated that semi-supervised training can improve the performance of Frisian-Dutch code-switched ASR (Yılmaz et al., 2018) as well as South African code-switched ASR (Biswas et al., 2019). The approach taken in this study is illustrated in Figure 1. The figure shows the two phases of semi-supervised training for the parallel bilingual as well as five-lingual configurations: automatic transcription followed by bilingual semi-supervised acoustic model retraining. The five-lingual system was not retrained with the automatically transcribed data for this set of experiments as our primary motive was to study the effect of automatic segmented speech on bilingual semi-supervised ASR.

4.1. Parallel Bilingual Transcription

This system (System A) consists of four subsystems, each corresponding to a language pair for which code-switching

occurs. Acoustic models were trained on the manually segmented and transcribed soap opera data (ManT, described in Section 2.1) pooled with the manually segmented but automatically transcribed speech (AutoT$_{\text{Exp}}$, introduced in Section 2.2). Because the languages spoken in the untranscribed data were unknown, each segment was decoded in parallel by each of the bilingual decoders. The output with the highest confidence score provided both the transcription and a language pair label for each segment.

4.2. Five-lingual Transcription

Some of our previous experiments indicated that the automatic transcriptions generated by the five-lingual baseline model enhanced the performance of the bilingual semi-supervised systems (Biswas et al., 2019). Five-lingual transcriptions were therefore also included in this study. The five-lingual system (System H) is based on a single acoustic model trained on all five languages. It was trained on the same data as the bilingual systems, except for the fact that the AutoT$_{\text{Exp}}$ data was transcribed using a five-lingual baseline model.

Since the five-lingual system is not restricted to bilingual output, its output allows Bantu-to-Bantu language switching. Examples of such switches were indeed observed in the transcriptions. Moreover, the automatically generated transcriptions sometimes contained more than two languages. Although the use of more than two languages within a single segment is not common, we have observed such cases during the compilation of the manually transcribed dataset. For our fast, continuous speech, the automatically generated segments have been observed to produce longer segments than manual segmentation of the data. This increases the likelihood of multiple language switches within the segment. Unfortunately, since the automatic segments are generated from untranscribed data, the degree to which multiple languages occur within a single automatic segment is difficult to quantify.

5. Automatic Speech Recognition

5.1. Acoustic Modelling

All acoustic models were trained using the Kaldi ASR toolkit (Povey et al., 2011) and the data described in Section 2. The models were trained on a multilingual dataset that included all the data in Table 1. In addition, three-fold data augmentation (Ko et al., 2015) was applied prior to feature extraction. The feature set included standard 40-dimensional MFCCs (no derivatives), 3- dimensional pitch and 100 dimensional i-vectors.

The models were trained with lattice free MMI (Povey et al., 2016) using the standard Kaldi CNN-TDNN-F (Povey et al., 2018) Librispeech recipe (6 CNN layers and 10 time-delay layers followed by a rank reduction layer) and the default hyperparameters. All acoustic models consist of a single shared softmax layer for all languages, as in general there is more than one target language in a segment.

No phone merging was performed between languages and the acoustic models were all language dependent. For the bilingual experiments, the multilingual acoustic models were adapted to each of the four target language pairs.
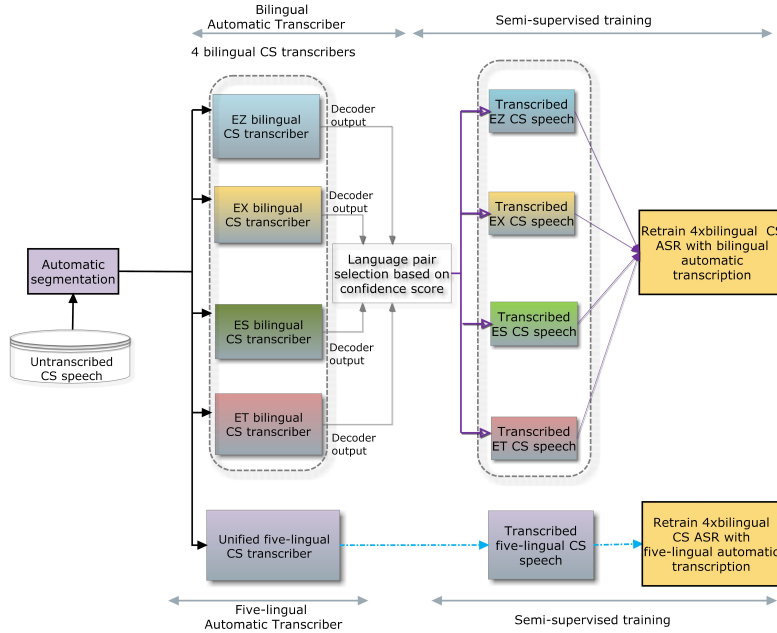
Figure 1: Semi-supervised training framework for bilingual code-switch (CS) ASR. EZ, EX, ES and ET refer to Engish-isiZulu, English-isiXhosa, English-Sesotho and English-Setswana language pairs respectively.

5.2. Language Modelling

The EZ, EX, ES, ET vocabularies contained 11 292, 8 805, 4 233, 4 957 word types respectively and were closed with respect to the training, development and test sets. The vocabularies were closed since the small datasets and the agglutinative character of the Bantu languages would otherwise lead to very high out-of-vocabulary rates. The SRILM toolkit (Stolcke, 2002) was used to train and evaluate all language models (LMs).

Transcriptions of the balanced subset of the ManT dataset as well as monolingual English and Bantu out-of-domain text were used to develop trigram language models. Four bilingual and one five-lingual trigram language model were used for the transcription systems as well as for semi-supervised training (Yılmaz et al., 2018; Biswas et al., 2019). Table 5 summarises the development and test set perplexities for the bilingual LMs. Details on the monolingual and code-switch perplexities are only provided for the test set (columns 3 to 6 in Table 5). The test set perplexities of the five-lingual LM are 1007.1, 1881.8, 345.3, and 277.5 for EZ, EX, ES and ET respectively. Further details regarding the five-lingual perplexities can be found in (Biswas et al., 2019).

Much more monolingual English text was available for language model development than text in the Bantu languages (471M vs 8M words). Therefore, the monolingual perplexity (MPP) is much higher for the Bantu languages than for English for each language pair.

Code-switch perplexities (CPP) for language switches indicate the uncertainty of the first word following a language switch. EB corresponds to switches from English to a Bantu language and BE indicates a switch in the other direction. Table 5 shows that the CPP for switching from English to isiZulu and isiXhosa is much higher than switching from these languages to English. This can be ascribed to the much larger isiZulu and isiXhosa vocabularies, which

are, in turn, due to the high degree of agglutination and the use of conjunctive orthography in these languages. The CPP for switching from English to Sesotho and Setswana is found to be lower than switching from those languages to English. We believe that this difference is due to the much larger English training set. The CPP values are even higher for the five-lingual language model. This is because the five-lingual trigrams allow language switches not permitted by the bilingual models.

6. Semi-supervised Training

For semi-supervised ASR, lattice-based supervision was combined with the lattice-free MMI objective function (Manohar et al., 2018; Carmantini et al., 2019). Conventionally, semi-supervised training only considers the best path while lattice-based supervision uses the entire decoding lattice. Hence, the latter approach allows the model to learn from alternative hypotheses when the best path is not accurate.

Table 6 gives an overview of the bilingual ASR systems that were trained using the manually segmented data (System B) as well as five different versions of the automatically segmented data (Systems C-G & I). In addition to manually-transcribed speech, ManT, the $AutoT_{Exp}$ data was also included in all the training sets.

Also defined in Table 6 are systems A and H, which are the bilingual and five-lingual baseline systems respectively, trained only on the ManT and $AutoT_{Exp}$ data. These baseline systems were used to obtain automatic transcriptions, $AutoT_A$ and $AutoT_H$, for each version of the additional data shown in Table 6. These automatic transcriptions were subsequently used to train new acoustic models.

VAD_{2Sub} was included to enable a fair comparison between automatic and manual segmentation. This is a 21-hour, randomly selected subset of the VAD_2 data which is comparable in size to the manually-segmented dataset ($AutoT_{NonE}$).

	Dev	Test	all CPP	CPP_{EB}	CPP_{BE}	all MPP	MPP_E	MPP_B
EZ	425.8	601.7	3 291.9	3 835.0	2 865.4	358.1	121.1	777.8
EX	352.9	788.8	4 914.4	6 549.6	3 785.6	459.0	96.8	1 355.6
ES	151.5	180.5	959.0	208.6	4 059.1	121.2	126.9	117.8
ET	213.3	224.5	70.2	317.3	3 798.1	160.4	142.1	176.1

Table 5: Development and test set perplexities. CPP: code-switch perplexity. MPP: monolingual perplexity.

| System | Type | Training segments | | | | | |
		$AutoT_{NonE}$ (23h)	VAD_1 (83.6h)	VAD_2 (47h)	VAD_{2Sub} (20.9h)	VAD_3 (37.0h)	VAD_4 (45.63h)
A	Bilingual baseline						
B		✓					
C			✓				
D	Bilingual system trained with $AutoT_A$			✓			
E					✓		
F						✓	
G							✓
H	Five-lingual baseline						
I	Bilingual system trained with $AutoT_H$					✓	

Table 6: ASR systems trained on different versions of the automatically segmented data. The duration of each of these datasets is given in parentheses.

7. Results & Discussion

The next three subsections concern results of the systems described in Sections 3., 4. and 5. Finally, ASR results are presented for specific languages, as well as at code-switching points.

7.1. Automatic Segmentation

AVA-Speech (test), which is the ≈ 23 hour subset of AVA-Speech not used for HMM training, was used as a test set to evaluate VAD performance. This dataset provides similar conditions to our target domain of soap opera data, as well as dense voice activity labels. Furthermore, it is accompanied by baseline results for the WebRTC project VAD (WebRTC.org, 2011) as well as two CNN-based systems based on the architecture proposed in (Hershey et al., 2017). The smaller of these two CNN-based systems, *tiny320*, is similar in size to our CNN, also containing three convolutional layers, while the other, *resnet960*, is based on the much larger ResNet-50 architecture (He et al., 2016).

Frame-based true positive rates (TPR) for a fixed false positive rate (FPR), scored over 10ms frames are shown for all VAD systems in Table 7. To allow comparison, all VAD systems were tuned to achieve a FPR of 0.315, as described in (Chaudhuri et al., 2018). TPR is reported for each individual speech condition (clean speech, speech with noise and speech with music) as well as for all conditions combined.

As expected, VAD_1 performs poorly. However, it is interesting to note that it is the only system that performs better for the "Noise" and "Music" conditions than for the "Clean" condition. This is because noisy signals tend to have more energy than their clean counterparts, making noisy signals more likely to exceed an energy threshold.

A large performance improvement is seen for VAD_2 which uses the CNN-HMM. In particular, this system already outperforms *tiny320*. A smaller performance increase is re-

| Model | True positive rate | | | |
	Clean	Noise	Music	All
RTCvad	0.786	0.706	0.733	0.722
tiny320	0.965	0.826	0.623	0.810
resnet960	0.992	0.944	0.787	0.917
VAD_1	0.564	0.662	0.693	0.646
VAD_2	0.972	0.898	0.778	0.886
VAD_3	0.985	0.917	0.811	0.907

Table 7: The true positive rate reported at a false positive rate of 0.315 for various VAD systems tested on AVA-Speech. The first three systems are baselines from (Chaudhuri et al., 2018), tested on the full dataset. The final three systems are tested on a ≈ 23 hour test set split.

ported for VAD_3 which uses the CNN-GMM-HMM. However, this increase brings its performance to a level comparable to the much larger *resnet960* system. In the case of "Music", VAD_3 outperforms *resnet960*, whilst for "All" the TPR of VAD_3 is within 1% absolute.

In terms of computational complexity, the energy VAD is about 30 times faster than the CNN based VADs. However, the speaker diarization system is two orders of magnitude slower than the slowest VAD, making the compute times of VAD_1, VAD_2 and VAD_3 are roughly equivalent. VAD_4, which removes the speaker diarization, is much faster.

7.2. Automatic Transcription

The automatic transcription outputs of the bilingual (System A) and five-lingual (System H) baseline systems are summarised in Table 8. The first five rows of the table correspond to segments that were classified as monolingual while the last row shows the number of segments that contain code-switching. The values in this row reveal a high number of code-switched segments in the additional data.

In terms of the number of segments per category, the output of the automatic segmentation systems agree with the manual segmentation process. The only exception is the number of English segments identified by the five-lingual system, which is higher than for the other systems. We believe that this is because the five-lingual language model was trained on more in-domain English text (Biswas et al., 2019).

The table also shows that including speaker diarization in the segmentation process produces smaller chunks of words than using only the VAD. Due to the varying duration of each set, comparisons are difficult to make. For this reason, the set VAD_{2Sub} is included, which is of a similar duration to $AutoT_{NonE}$, allowing comparison between the non-expert manual segmentation and automatic segmentation. It

System	A: Bilingual						H: Five-lingual
Language	$\text{AutoT}_{\text{NonE}}$	VAD_1	VAD_2	VAD_{2Sub}	VAD_3	VAD_4	VAD_3
English	8 570	12 155	7 608	4 721	11 686	4 754	23 973
IsiZulu	5 955	4 084	3 583	2 065	7 995	2 122	7 315
IsiXhosa	302	154	116	57	443	236	831
Sesotho	1 317	2 267	1 695	759	3 457	719	1 942
Setswana	2 598	6 272	4 341	2 241	6 691	2 196	1 973
Code-switched	25 824	39 562	30 904	12 572	36 475	17 911	30 616

Table 8: Number of segments per language identified by the baseline bilingual (A) and baseline five-lingual (H) ASR systems for different segmentation approaches.

can be seen that for the same duration of data, the automatic segmentation produces fewer segments.

7.3. Automatic Speech Recognition

The performance of the ASR systems introduced in Table 6 were measured in terms of the word error rate (WER) achieved after semi-supervised training. Results for the different training configurations are reported in Table 9. The values in the table indicate that including the additional 23 hours of non-expert manually segmented data ($\text{AutoT}_{\text{NonE}}$) in the training set (System B) yields absolute improvements of 1.5% and 1.4% over the baseline (System A) for the development and test sets respectively.

The results for System C show that using 83 hours of automatically-segmented speech results in an absolute improvement of 1.7% and 1.6% for the development and test sets relative to System A. Although System C's performance is on par with that of System B, its training set was much larger which means that the computational cost of developing the system is also much higher.

According to Table 6, VAD_2 reduced the additional training data from 83.6 to 47 hours. Furthermore, including this reduced additional data in System D's training set resulted in lower WERs for both the development and test sets, when compared with Systems A, B and C.

The semi-supervised system trained on the 21-hour subset of VAD_2 (System E) achieved results that are comparable to those of System B. The two additional dataset seem to have had almost the same impact on the accuracy of the resulting acoustic models. This result seems to indicate that, in terms of ASR performance, manually and automatically-produced segmentations are equally well suited for system development. However, it should be kept in mind that the segment labels used by System B were not assigned by experts.

System F, trained on the segments generated by VAD_3, yielded better performance than system D, despite the fact that System D's training set contained 10 more hours of data. The improvement in WER was found to be statistically significant at the 95% confidence level using bootstrap interval estimation (Bisani and Ney, 2004).

The results for System G show that automatic segments that do not take speaker identity into account (VAD_4) do not achieve the accuracy levels as those that do (System F). Therefore, the inclusion of speaker diarization does tend to improve ASR performance.

The performance of System H (five-lingual baseline system) is included in Table 9 but should not be directly com-
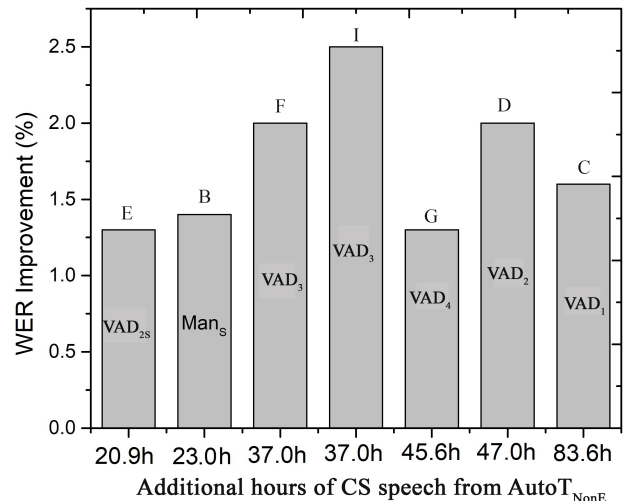

Figure 2: Improvement (% in comparison with the baseline) in test set WER for different semi-supervised systems incorporating additional soap opera training data.

pared with the bilingual systems because the recognition task is inherently more complex. However, as has been observed before (Biswas et al., 2019), the bilingual System I, trained on automatic transcriptions generated by System H, shows the best overall performance of all the evaluated systems. The improvement on the test set over its closest competitor (System F) is 0.5% absolute and this was found to be statistically significant above the 90% confidence level using bootstrap interval estimation. This improvement may be due to the ability of the five-lingual system to transcribe more than two languages, as well as Bantu-to-Bantu switches. The untranscribed soap opera speech is known to contain at least some segments that do not conform to the four considered bilingual language groupings. The degree to which such language switches do occur is unfortunately difficult to quantify without manual transcriptions. However, since the key difference between the bilingual and five-lingual systems is the ability to handle a greater variety of language switches, we speculate that this is a likely cause for the superior performance of System I.

The improvement in WER achieved by the semi-supervised ASR systems incorporating different versions of the additional data is summarised in Figure 2. The figure confirms that the largest gain in recognition accuracy was achieved by System I. It also affirms the observation that an equal amount of manually and automatically segmented

CS Pair	A (baseline)		B		C		D		E		F		G		H		I	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
EZ	34.5	40.8	33.1	39.6	33.3	39.2	33.1	39.0	33.8	39.1	32.7	38.6	33.3	38.9	37.6	43.6	33.2	38.5
EX	35.8	42.7	35.3	42.0	34.8	41.9	34.7	41.8	35.2	42.1	34.7	41.4	34.7	42.3	40.6	54.5	33.8	41.0
ES	51.7	48.7	48.7	46.5	49.8	47.3	48.7	47.7	49.0	47.1	49.2	46.8	49.1	47.9	54.5	49.3	49.6	45.7
ET	44.3	41.3	42.7	39.7	41.1	38.9	40.7	38.5	41.7	40.0	40.1	38.7	40.8	39.3	47.2	43.9	39.9	38.4
Overall	41.5	**43.4**	40.0	**42.0**	39.8	**41.8**	39.3	**41.7**	39.9	**42.1**	39.2	**41.4**	39.5	**42.1**	46.5	**46.7**	39.1	**40.9**

Table 9: Mixed WERs (%) for the four code-switched language pairs.

System	English-isiZulu			English-isiXhosa			English-Sesotho			English-Setswana		
	E	Z	Bi_{CS}	E	X	Bi_{CS}	E	S	Bi_{CS}	E	T	Bi_{CS}
A (baseline)	37.9	48.7	33.3	37.8	54.5	25.8	43.7	61.4	25.2	36.2	51.8	35.6
B	32.3	45.2	36.8	32.7	49.1	32.1	32.9	57.2	33.7	28.1	48.3	40.5
F	31.6	43.9	37.8	31.5	48.3	34.2	32.5	56.8	33.8	27.4	46.4	42.2
I	31.7	43.7	37.3	31.6	47.6	34.4	32.0	56.4	34.2	26.8	45.7	42.0

Table 10: Language specific WER (%) (lowest is best) for English (E), isiZulu (Z), isiXhosa (X), Sesotho (S), Setswana (T) and code-switched bigram correct (Bi_{CS}) (%) (highest is best) for the test set.

data yields an equal improvement in recognition accuracy in a semi-supervised set-up.

7.4. Language Specific WER Analysis

For code-switched ASR, the performance of the recogniser at code-switch points is of particular interest. Language specific WERs and code-switched bigram correct (Bi_{CS}) values for the different semi-supervised systems are presented in Table 10. Code-switch bigram correct is defined as the percentage of words correctly recognised immediately after code-switch points. All values are percentages. The table reveals that both the English and Bantu WERs for all the semi-supervised systems are substantially lower than the corresponding values for the baseline system. The accuracy at the code-switch points is also substantially higher for the semi-supervised systems. Hence, adding the additional training data enhances system performance at code-switch points. Moreover, there are no substantial differences between the gains achieved by adding the manually (System B) or automatically (Systems F, I) segmented data.

8. Conclusions

In this study, we have evaluated the impact of using automatically-segmented instead of manually-segmented speech data for semi-supervised training of a code-switched automatic speech recognition system. Four different automatic segmentation approaches were evaluated, based respectively on simple energy thresholding with diarization, a CNN classification with two variants of HMM smoothing and diarization, and CNN classification with GMM-HMM smoothing and no diarization. It was found that applying our new CNN-GMM-HMM based VAD followed by X-vector speaker diarization resulted in the best ASR performance. The results also showed that the performance of systems that used automatically and manually-segmented data were comparable. We conclude that automatic-segmentation in combination with semi-supervised training is a viable approach to enhancing the recognition accuracy of a challenging five-language code-switched speech recognition task. This is a very positive outcome, since the difficulty in providing a manual segmentation of new broadcast material has remained an impediment to the development of speech technology in severely under resourced settings such as the one we describe. Future work will focus on improving the VAD and speaker diarization techniques as well as incorporating language identification into the automatic segmentation process.

10. Bibliographical References

Ahmed, B. H. and Tan, T.-P. (2012). Automatic speech recognition of code switching speech using 1-best rescoring. In *Proc. IALP*, Hanoi, Vietnam.

Amazouz, D., Adda-Decker, M., and Lamel, L. (2017). Addressing code-switching in French/Algerian Arabic speech. In *Proc. Interspeech*, Stockhom, Sweden.

Bisani, M. and Ney, H. (2004). Bootstrap estimates for confidence intervals in ASR performance evaluation. In *Proc. ICASSP*, Montreal, Canada.

Biswas, A., de Wet, F., van der Westhuizen, E., Yılmaz, E., and Niesler, T. R. (2018a). Multilingual neural network acoustic modelling for ASR of under-resourced English-isiZulu code-switched speech. In *Proc. Interspeech*, Hyderabad, India.

Biswas, A., van der Westhuizen, E., Niesler, T. R., and de Wet, F. (2018b). Improving ASR for code-switched speech in under-resourced languages using out-of-domain data. In *Proc. SLTU*, Gurugram, India.

Biswas, A., Yılmaz, E., de Wet, F., van der Westhuizen, E., and Niesler, T. R. (2019). Semi-supervised acoustic model training for five-lingual code-switched ASR. In *Proc. Interspeech*, Graz, Austria.

Biswas, A., Yılmaz, E., de Wet, F., van der Westhuizen, E., and Niesler, T. R. (2020). Semi-supervised development of ASR systems for multilingual code-switched speech in under-resourced languages. In *Proc. LREC*, Marseille, France.

Carmantini, A., Bell, P., and Renals, S. (2019). Untranscribed web audio for low resource speech recognition. In *Proc. Interspeech*, Graz, Austria.

Chaudhuri, S., Roth, J., Ellis, D., Gallagher, A. C., Kaver, L., Marvin, R., Pantofaru, C., Reale, N. C., Reid, L. G., Wilson, K., and Xi, Z. (2018). AVA-speech: A densely labeled dataset of speech activity in movies. In *Proc. Interspeech*, Hyderabad, India.

Chung, J. S., Nagrani, A., and Zisserman, A. (2018). Voxceleb2: Deep speaker recognition. In *Proc. Interspeech*, Hyderabad, India.

Emond, J., Ramabhadran, B., Roark, B., Moreno, P., and Ma, M. (2018). Transliteration based approaches to improve code-switched speech recognition performance. In *Proc. SLT*, Athens, Greece.

Ganji, S., Dhawan, K., and Sinha, R. (2019). IITG-HingCoS corpus: A Hinglish code-switching database for automatic speech recognition. *Speech Communication*, 110:76–89.

Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. (2017). Audio set: An ontology and human-labeled dataset for audio events. In *Proc. ICASSP*, New Orleans, USA.

Guo, P., Xu, H., Xie, L., and Chng, E. S. (2018). Study of semi-supervised approaches to improving English-Mandarin code-switching speech recognition. In *Proc. Interspeech*, Hyderabad, India.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proc. CVPR*, Las Vegas, USA.

Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R. J., and Wilson, K. (2017). CNN architectures for large-scale audio classification. In *Proc. ICASSP*, New Orleans, USA.

Ko, T., Peddinti, V., Povey, D., and Khudanpur, S. (2015). Audio augmentation for speech recognition. In *Proc. Interspeech*, Dresden, Germany.

Li, Y. and Fung, P. (2013). Improved mixed language speech recognition using asymmetric acoustic model and language model with code-switch inversion constraints. In *Proc. ICASSP*, Vancouver, Canada.

Manohar, V., Hadian, H., Povey, D., and Khudanpur, S. (2018). Semi-supervised training of acoustic models using lattice-free MMI. In *Proc. ICASSP*, Calgary, Canada.

Nagrani, A., Chung, J. S., and Zisserman, A. (2017). Voxceleb: a large-scale speaker identification dataset. In *Proc. Interspeech*, Stockholm, Sweden.

Nakayama, S., Tjandra, A., Sakti, S., and Nakamura, S. (2018). Speech chain for semi-supervised learning of Japanese-English code-switching ASR and TTS. In *Proc. SLT*, Athens, Greece.

Pandey, A., Srivastava, B. M. L., Kumar, R., Nellore, B. T.,

Teja, K. S., and Gangashetty, S. V. (2018). Phonetically balanced code-mixed speech corpus for Hindi-English automatic speech recognition. In *Proc. LREC*, Miyazaki, Japan.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al. (2011). The Kaldi speech recognition toolkit. In *Proc. ASRU*, Hawaii.

Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., and Khudanpur, S. (2016). Purely sequence-trained neural networks for ASR based on lattice-free MMI. In *Proc. Interspeech*, San Francisco, USA.

Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohammadi, M., and Khudanpur, S. (2018). Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Proc. Interspeech*, Hyderabad, India.

Sehgal, A. and Kehtarnavaz, N. (2018). A convolutional neural network smartphone app for real-time voice activity detection. *IEEE Access*, 6:9017–9026.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (2018). X-vectors: Robust DNN embeddings for speaker recognition. In *Proc. ICASSP*, Calgary, Canada.

Stolcke, A. (2002). SRILM – An extensible language modeling toolkit. In *Proc. ICSLP*, Denver, USA.

Taneja, K., Guha, S., Jyothi, P., and Abraham, B. (2019). Exploiting monolingual speech corpora for code-mixed speech recognition. In *Proc. Interspeech*, Graz, Austria.

Thomas, S., Seltzer, M. L., Church, K., and Hermansky, H. (2013). Deep neural network features and semi-supervised training for low resource speech recognition. In *Proc. ICASSP*, Vancouver, Canada.

Thomas, S., Ganapathy, S., Saon, G., and Soltau, H. (2014). Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions. In *Proc. ICASSP*, Florence, Italy.

van der Westhuizen, E. and Niesler, T. R. (2018). A first South African corpus of multilingual code-switched soap opera speech. In *Proc. LREC*, Miyazaki, Japan.

Vu, N. T., Lyu, D.-C., Weiner, J., Telaar, D., Schlippe, T., Blaicher, F., Chng, E.-S., Schultz, T., and Li, H. (2012). A first speech recognition system for Mandarin-English code-switch conversational speech. In *Proc. ICASSP*, Kyoto, Japan.

WebRTC.org. (2011). The WebRTC project. [Online]. Available: https://webrtc.org.

Yılmaz, E., van den Heuvel, H., and van Leeuwen, D. (2016). Investigating bilingual deep neural networks for automatic recognition of code-switching Frisian speech. *Procedia Computer Science*, 81:159–166.

Yılmaz, E., McLaren, M., van den Heuvel, H., and van Leeuwen, D. (2018). Semi-supervised acoustic model training for speech with code-switching. *Speech Communication*, 105:12–22.

Zeng, Z., Khassanov, Y., Pham, V. T., Xu, H., Chng, E. S., and Li, H. (2018). On the end-to-end solution to Mandarin-English code-switching speech recognition. *arXiv preprint arXiv:1811.00241*.