Article

Improving N-Best Rescoring in Under-Resourced Code-Switched Speech Recognition Using Pretraining and Data Augmentation

Joshua Jansen van Vüren [1,†,‡] and Thomas Niesler [1,‡]

[1] Stellenbosch University; jjvanvueren@sun.ac.za
[2] Stellenbosch University; trn@sun.ac.za
* Correspondence: jjvanvueren@sun.ac.za
† Current address: Stellenbosch University, Stellenbosch, South Africa

Abstract: We present improvements in n-best rescoring of code-switched speech achieved by n-gram augmentation as well as optimised pretraining of long short-term memory (LSTM) language models with larger corpora of out-of-domain monolingual text. In addition, we consider the application of large pretrained transformer-based architectures. Our experimental evaluation is performed on an under-resourced corpus of code-switched speech comprising four bilingual code-switched sub-corpora, each containing a Bantu language (isiZulu, isiXhosa, Sesotho, or Setswana) and English. We find in our experiments that, by combining n-gram augmentation with the optimised pretraining strategy, speech recognition errors are reduced for each individual bilingual pair by 3.51% absolute on average over the four corpora. Importantly, we find that even speech recognition at language boundaries improves by 1.14% even though the additional data is monolingual. Utilising the augmented n-grams for lattice generation, we then contrast these improvements with those achieved after fine-tuning pretrained transformer-based models such as distilled GPT-2 and M-BERT. We find that, even though these language models have not been trained on any of our target languages, they can improve speech recognition performance even in zero-shot settings. After fine-tuning on in-domain data, these large architectures offer further improvements, achieving a 4.45% absolute decrease in overall speech recognition errors and a 3.52% improvement over language boundaries. Finally, a combination of the optimised LSTM and fine-tuned BERT models achieves a further gain of 0.47% absolute on average for three of the four language pairs compared to M-BERT. We conclude that the careful optimisation of the pretraining strategy used for neural network language models can offer worthwhile improvements in speech recognition accuracy even at language switches, and that much larger state-of-the-art architectures such as GPT-2 and M-BERT promise even further gains.

Keywords: code-switching; automatic speech recognition; low resource languages; language modelling

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Preprints* **2022**, *1*, 0. https://doi.org/

1. Introduction

Language modelling in under-resourced settings, such as those encountered for African languages, is challenging [1,2]. In comparison with the large datasets available for English and other highly resourced languages, very few large corpora are available for African languages, and when they are available, the applicability of their domain is narrow. In addition to this, code-switching, which involves the use of multiple languages within or between utterances, is common in everyday speech in African countries. In the context of language modelling and speech recognition, code-switching results in a high confusion and consequently increased errors around language switches, as shown in this work. Research notes that code-switches vary between speakers and across speech domains, in addition, data for modelling code-switching is under-resourced and therefore modelling the phenomenon statistically is challenging [3]. This makes them challenging to model within a singular linguistic paradigm; such as matrix language frame theory [4] or equivalence constraint theory [5].

In this work we consider the inclusion of larger monolingual corpora in both related and unrelated languages for language model pretraining as a way of alleviating data scarcity. We show the surprising result that, although the additional data is monolingual its inclusion allows us to improve speech recognition accuracies, even across language switches. Additionally, we show that even when language models are pretrained on completely unrelated languages, it is still possible to achieve improvements in speech recognition even across language switches.

We present this work in two parts. Firstly, we optimize the pretraining of an LSTM model utilising available out-of-domain monolingual corpora in our target languages - isiZulu, isiXhosa, Sesotho, Setswana, and English. These language models are then fine-tuned on sets of bilingual soap opera data [6] and used for n-best rescoring. Synthetic code-switched data is also incorporated into our pretraining data. We show that, even in situations where the vocabulary is closed on the types in the training, development and test sets for each respective sub-corpus in the under-resourced dataset, resulting in high out of vocabulary rates (26.13%-70.72%) during pretraining on the respective out-of-domain corpora, that the neural language models improve substantially when pretrained using a curated dataset.

In the second portion of this work we refrain from pretraining the neural language models ourselves and instead utilise a selection of publicly available models that have not been trained on any of the target languages considered in this work[1]. We find that, when applying these models in n-best rescoring after fine-tuning on the same in-domain data used in the LSTM experiments, we again see improved speech recognition even over language switches. In fact, even without fine-tuning (a zero-shot setting) small improvements are possible.

The remainder of this document is organised as follows. Section 2 provides a description of the background and literature. Section 3 describes the corpora utilised in this work, and Section 4 presents our experimental setup, detailing how the corpora and model architectures are utilised in pretraining and augmentation experiments. The augmented n-gram and neural language models are then utilised for lattice generation and n-best rescoring respectively, the results from these experiments are presented in Section 5. This section also presents the rescoring results utilising the publicly available pretrained models. Finally, Section 6 concludes.

2. Background

Research has found that pretraining on synthetic data generated by an adversarial network can improve code-switched language modelling [7]. Additionally, an extensive investigation of TDNN, LSTM, transformer, and other neural language models found that, for the two African languages isiZulu and Sepedi, the inclusion of training data from nine non-European South African languages improves language modelling performance [2]. Improvements in language modelling and speech recognition for two under-resourced South African languages - isiZulu and Setswana - are also achieved by using sub-word and multi-word tokenization techniques respectively [1].

Utilising the same corpus of multilingual code-switched speech considered in this work, Biswas *et al.* [8] investigate the impact of the inclusion of both acoustic and textual data on speech recognition accuracy. Specifically, four balanced bilingual corpora are used to train bilingual speech recognition systems. The subsequent inclusion of more in-domain speech, which leads to an imbalance in the training data, improves performance. Further improvements are afforded by including out-of-domain monolingual speech [9] in each of the considered languages. Finally, speech recognition in the same languages can also be improved by incorporating additional out-of-domain monolingual text as well as synthetically generated code-switched bigrams to the language model [10].

Typically the performance achieved by the current state of the art language models is attributed the their large training corpora, which are far larger even than our monolingual pretraining sets. However, research has shown that competitive multilingual language models can be trained utilising data from under-resourced (<1GB training data) African languages [11], even when compared to large M-BERT [12] models trained on much larger amounts (100GB) of mostly non-African languages.

In Ralethe [13], a monolingual Afrikaans BERT model was shown to outperform a fine-tuned multilingual BERT in named entity recognition and dependency parsing. Utilising an Afrikaans text corpus, a new sub-word encoded vocabulary was created. The monolingual Afrikaans BERT model was then initialised utilising the parameters of mBERT. However, only the overlapping sub-word embeddings were utilised, while all new tokens in the vocabulary were assigned randomly initialised embeddings.

Research aimed specifically to improve code-switched language modelling and speech recognition has employed state of the art architectures for text synthesis. Pretrained BERT models have been employed in adversarial fine-tuning frameworks to generate code-switched text [14], which was shown to improve speech recognition accuracy when the data is incorporated into the language model training data. State-of-the-art transformer models [15] have also been utilised to generate code-switched text which, when used to train an LSTM model, reduced test set perplexities [16].

In this work we demonstrate improvements in language modelling and speech recognition by optimising the pretraining strategy for an LSTM language model utilised in n-best rescoring experiments. We show the surprising result that although the pretraining data is monolingual, we observe improvements even across language switches. We also demonstrate that large publicly available architectures, such as multilingual BERT and distilled GPT-2, are able to improve the speech recognition accuracy for the same code-switched speech. Use of these architectures can be seen as a way of pretraining on very large out-of-domain and out-of-language datasets. To the best of our knowledge, this is the first study which investigates the impact of such bidirectional transformer models on code-switched speech recognition and particularly in African languages [17,18].

3. Datasets

For experimentation, we utilise an under-resourced corpus of code-switched speech collected from South African Soap Operas [6]. We evaluate the performance of our pre-training and augmentation techniques using the development and test set defined in this corpus. The dataset is split into four bilingual corpora, each consisting of a Bantu language and English, as shown in Table 1.

For each of the languages in this corpus, a respective out-of-domain monolingual corpus is available, which has been collected from newspapers and web content [8]. For each of the sub-corpora we define a bilingual vocabulary closed on the types in the training, development, and test sets highlighted in Table 1. In Table 2 we present the out of vocabulary rates (OOV) when utilising these respective vocabularies on the Bantu and English monolingual corpora. We note that in general the out of vocabulary rates are high, in excess of 26%. It is also clear from the table that the out of vocabulary rates are always higher for the Bantu corpora than for English. This is especially true for the Nguni languages (isiZulu and isiXhosa) which are known to have large vocabularies due to their agglutinative orthographies.

Additionally we utilise four synthetic corpora of bilingual code-switched text, generated using an existing LSTM network [19]. The token counts for the monolingual and the synthesized text are given in Table 3.

4. Experimental Setup

In this section we outline the experimental setup of our work. We begin, in Section 4.1, by outlining the evaluation metrics utilised in our language modelling and speech recognition experiments. In Section 4.2 we detail the experimental setup of our baseline speech recognition system, and specify the toolkits we employed to train our n-gram models and neural networks. Section 4.3 describes the architecture of the baseline and pretrained LSTM language models. Subsequently, Section 4.4 presents our LSTM pretraining optimisation

Table 1. The soap opera corpus, showing the four bilingual corpora (English-isiZulu, English-isiXhosa, English-Sesotho, and English-Setswana). For each corpus the total number of word tokens (Tok), word types (Uniq), and code switches (CS) is presented. We denote the number of code-switches from English to a Bantu language as CS_{EB}, while CS_{BE} indicates the number of switches from Bantu to English. The final column (Dur) presents the duration of each corpus in minutes (m) or hours (h). Adapted from [6].

	English		Bantu		**Total**				
Partition	Tok	Uniq	Tok	Uniq	Tok	Uniq	CS_{EB}	CS_{BE}	Dur
				English-isiZulu					
Train	28033	3608	24350	6788	52383	10396	2236	2743	4.81h
Dev	832	414	734	452	1566	866	175	198	8.00m
Test	2457	870	3199	1435	5656	2305	688	776	30.4m
Total	31322	3841	28283	7448	59605	11289	3099	3717	5.45h
				English-isiXhosa					
Train	20324	2630	12215	5086	32539	7716	776	1003	2.68h
Dev	1153	484	1147	762	2300	1246	91	113	13.7m
Test	1149	498	1502	889	2651	1387	328	363	14.3m
Total	22626	2828	14864	5975	37490	8803	1195	1479	3.14h
				English-Sesotho					
Train	15395	2255	19825	2086	35197	4339	1565	1719	2.36h
Dev	843	437	2227	614	3067	1050	156	166	12.8m
Test	1794	659	2265	535	4054	1193	403	396	15.5m
Total	18032	2520	24317	2437	42318	4955	2124	2281	2.83h
				English-Setswana					
Train	16180	2361	19570	1448	35725	3808	1885	1951	2.33h
Dev	1170	514	2539	539	3707	1052	224	251	13.8m
Test	1970	729	2979	526	4939	1254	505	526	17.8m
Total	19320	2607	25088	1625	44371	4231	2614	2728	2.86h

Table 2. Out of vocabulary rates for the monolingual corpora when a vocabulary is closed on the four respective bilingual datasets.

Label	English-isiZulu	English-isiXhosa	English-Sesotho	English-Setswana
Bantu	62.31%	70.72%	36.39%	35.16%
English	26.13%	30.22%	32.45%	31.53%

strategy, Section 4.5 presents the n-gram optimisation strategy, and finally Section 4.6 presents the fine-tuning strategies for larger pretrained transformer architectures.

4.1. Code-switched perplexity (CPP) and bigram error rate (CSBG)

We evaluate the performance of our language models utilising the perplexity calculated for the development and test sets as defined in Table 1. To analyse the model performance specifically over code-switches, we also calculate the average perplexity over only language switches, and refer to this as the code-switched perplexity (CPP). In a similar fashion, we evaluate speech recognition performance at language switches by specifically calculating the speech recognition error rate only at the point of language transition, and refer to this as the code-switched bigram error rate (CSBG).

Table 3. Token counts for the four available monolingual corpora, as well as for the synthetic code-switched corpus, which contains both English and the respective Bantu language.

Label	English	isiZulu	isiXhosa	Sesotho	Setswana
Monolingual	471M	3.25M	0.99M	0.23M	2.84M
Synthetic	-	6.6M	4.18M	10.7M	8.08M

4.2. Speech Recognition

The speech recognition system utilised in our work is trained using the Kaldi toolkit [20]. A CNN-TDNN-F acoustic model is pretrained on the pooled audio data and then fine-tuned on each bilingual corpus, as presented in [21]. All n-gram language models are trained using the SRILM toolkit [22], while the neural language models are trained using Tensorflow [23]. The baseline speech recognition system is denoted as ASR-B in the subsequent sections. Our baseline n-gram language models are trigrams with modified Kneser-Ney smoothing trained separately on each of the four bilingual soap opera training corpora. These are denoted by LM_B.

4.3. Language Model Architecture

The neural language model shown in Figure 1 is an LSTM with 256-dimensional embedding, and 256-dimensional recurrent dimensions. The embedding matrix weights are tied to the final matrix [24]. This means that the embedding matrix weights of dimension $(d_\text{embed}, d_\text{vocab})$ are utilised to form the output likelihood vectors $\mathbf{o}_n$ of dimensionality d_vocab in addition to producing embedded vectors of dimensionality d_embed for each token in the vocabulary. The output likelihood vectors $\mathbf{o}_n$ are calculated by applying the product between the output hidden state vectors of the LSTM ($\mathbf{h}_n$), whose dimensionality is also d_embed, and the embedding matrix. The resulting vector ($\mathbf{o}_n$) has the dimensionality of the vocabulary (d_vocab). In addition, L2 weight regularisation is applied to the LSTM recurrent kernel (the trainable weights which manipulate the input hidden state vector $\mathbf{h}_{n-1}$) and to the weight-tied embedding and dense matrix. These hyperparameters were chosen in correspondence to the best development set performance during preliminary experiments. We found that the incorporation of L2 regularisation improved performance, whilst weight tying allowed us to greatly reduce the number of model parameters as well as improve language model performance.

We train four separate baseline LSTM language models, denoted by N-LM_B, on each of the soap opera bilingual training corpora (Table 1) until convergence on the development set. We utilise a batch size of 32 and the ADAM gradient descent algorithm [25].

4.4. Optimisation of LSTM pretraining

In preliminary experiments we found that training the model separately on the monolingual data (i.e. by for example first training on the English data and then on the respective Bantu data) did not afford improvements in perplexity when the pretrained model is fine-tuned. However, interleaving and sub-sampling the sequences from the different corpora resulted in consistent perplexity improvements. Therefore, we investigate different methods of sequence level interleaving in order to determine empirically which affords the largest improvements in language modelling performance.

In our first experiment we interleave batches ($\mathbb{B}$) of the available sets by sub-sampling the largest corpus (typically English) at regular intervals until the number of sequences is the same as in the smallest set. The second strategy interleaves English and Bantu at the sequence level ($\mathbb{S}$), again by sub-sampling the largest dataset at regular intervals until it is the same size as the Bantu set.

For each of the strategies we close the vocabulary of a word-based LSTM language model on all word types in the soap opera dataset. We then pretrain using a specific interleaving strategy ($\mathbb{B}$ or $\mathbb{S}$) on a subset of the datasets presented in Table 4, after which the model is fine-tuned using the code-switched data (Table 1) until convergence on the development set. These models are indicated as N-LM in Table 4. Specifically, as indicated in this table, we consider three pooled corpora for pretraining. Firstly, utilising only the synthetic data (S), then combining the respective monolingual Bantu data as well as the monolingual English data (M), and finally utilising a combination of the synthetic, Bantu and English data (S+M).

In Section 5.1, we begin by optimising the number of pretraining epochs over the range $n_e = \{1, 2, \ldots, 5\}$.

Figure 1. Structure of the LSTM language model. Each batch of the respective training corpus is first preprocessed and tokenized, and then presented as input to the embedding layer, which computes the embedded vector $\mathbf{x}_n$. This embedded vector is presented as input to the LSTM network along with the hidden ($\mathbf{h}_{n-1}$) and cell state ($\mathbf{c}_{n-1}$) vectors. The updated hidden state vector is presented to the output dense layer, whose weights are shared with the embedding layer, to form a likelihood vector of the next possible words ($\mathbf{o}_n$). This process is repeated for each token in the current sequence. L2 weight regularisation is specifically applied to the weight-tied embedding and dense layer, as well as the recurrent kernel in the LSTM.

Table 4. Corpora considered for the different pretraining or augmentation strategies.

Model	Label	Soap Opera	Synthetic	Monolingual English	Monolingual Bantu
N-LM	S		×		
	M			×	×
	S+M		×	×	×
LM	B	×			
	B+S	×	×		
	B+M	×		×	×
	B+S+M	×	×	×	×

4.5. N-gram Optimisation

We would like to contrast the improvements in speech recognition when rescoring n-best lists with the pretrained language model to improvements seen when re-running the speech recognizer after augmenting the training sets of an n-gram model with the same data used for pretraining. If the improvements afforded by rescoring are comparable to those afforded by n-gram augmentation, the latter is preferable due to its reduced computational requirements and greater simplicity, especially in computationally-constrained settings. Additionally, we investigate whether the improvements afforded by n-gram augmentation and n-best rescoring are complementary.

The augmentation process is accomplished by training separate n-gram language models on each of the available corpora (soap opera, synthetic, and monolingual) and interpolating these three individual n-gram models according to the four different combinations of corpora (B,B+S,B+M,B+S+M) as indicated in Table 4. In all cases these interpolated n-gram language models, denoted by LM, are optimised according to the development set perplexity.

4.6. Large Pretrained Language Models

Large pretrained language models have become the de facto standard in industry. These models are trained on billions of tokens and achieve the state-of-the-art in many language modelling tasks. Here we investigate the application of these models in n-best rescoring of code-switched speech in both zero-shot settings as well as after fine-tuning on the code-switched data.

We note that many of these models are bidirectional and are trained using the masked language modelling (MLM) objective as first presented in Devlin *et al.* [12]. This training strategy randomly selects 15% of the input sequence tokens, and either replaces them with a specific mask token `[MSK]`, a random token, or the same token with 80%, 10%, and 10% probability respectively. The average cross-entropy loss is calculated over the masked tokens and utilised to calculate gradients via backpropagation for model weight training.

During rescoring, an autoregressive model is typically employed to calculate the likelihood of the next token given a sequence of previous tokens. However, because bidirectional models make use of context from both the left and the right of the current token, a different strategy must be employed. Specifically, when applying these models, the negative log likelihood is calculated for each token in a sequence given the remainder of the sequence as a context. This is accomplished by applying the masking token `[MSK]` to the nth position in a sequence and calculating the negative log-likelihood of the target token at the same position. This process is repeated for each position in the hypothesised sequence, in order to generate a negative log-likelihood score for the sequence as a whole. The scores for each sequence are then interpolated with the respective n-gram and acoustic model scores, in order to rescore the development and test set n-best lists.

To our knowledge these are the first investigations applying bidirectional transformer language model architectures to code-switched speech recognition. Although the structures of the transformers and LSTM language models are not directly comparable - they differ for example in pretraining data, architecture, and vocabulary size - we can nevertheless contrast their impact on speech recognition performance.

We consider five different bidirectional architectures: a distilled multilingual BERT model (D-M-BERT) [26], a multilingual BERT model (M-BERT) [12], a first BERT architecture trained on South African languages (afriBERTa-S) [11], another distilled multilingual transformer model distilled from XLM-R (mMiniLMv2) [27], and finally a distilled GPT-2 (GPT-2) model which is an autoregressive transformer [28].

Each of the considered models are applied in the following three ways. Firstly, in a zero-shot setting ($\mathbb{Z}$), where the baseline language model is applied with no additional training. Secondly, after fine-tuning the model for between one and ten epochs on each of the four bilingual soap opera corpora ($\mathbb{F}$-$\mathbb{B}$). Finally, by pooling the four bilingual language pairs and again fine-tuning the model for between one and ten epochs ($\mathbb{F}$-$\mathbb{P}$). Each of these resulting models is then evaluated by means of n-best rescoring experiments.

5. Results

First, in Section 5.1, we present the language modelling and n-best list rescoring results achieved by the optimisation of the monolingual pretraining of the LSTM language models as described in Section 4.4. In Section 5.2, we show the improvements afforded by optimising the augmented n-gram language model, and additional performance afforded by utilising both augmented n-grams as well as the pretrained LSTM for n-best rescoring as described in Section 4.5. Finally in Section 5.3, the results afforded by applying fine-tuned publicly available pretrained models, as described in Section 4.6, in n-best rescoring experiments is presented.

5.1. Optimisation of LSTM pretraining

In this section we present the results of the experiments that investigate the best pretraining strategy for the LSTM language model presented in Section 4.4. These results will be both in terms of perplexity and word error rate after n-best rescoring. We compare the performance of the pretrained model with the baseline LSTM language model (N-LM$_{\mathrm{B}}$) which is trained solely on the soap opera training data as described in Section 4.3. In each experiment, the LSTM is pretrained using data interleaved at the batch ($\mathbb{B}$) or sequence ($\mathbb{S}$) level as described in Table 4 for between one and five epochs. Table 5 shows that, on average over all four language pairs and five training epochs, the model pretrained using only the out-of-domain monolingual data (N-LM$_{\mathrm{M}}$) affords the largest improvements in perplexity of 42.87% relative to the baseline (N-LM$_{\mathrm{B}}$). Additionally, we find that interleaving at the sequence level ($\mathbb{S}$) is better than at the batch level ($\mathbb{B}$), affording a 45.4% average relative improvement in perplexity compared to a 40.33% average improvement relative to the baseline (N-LM$_{\mathrm{B}}$).

When considering the application of these same language models in 50-best list rescoring, a similar trend in terms of speech recognition improvements is seen. In Table 6, we find that interleaving the monolingual sets at the sequence level again affords the largest improvement, both in overall word error rate and in code-switched bigram error. We find that, on average over all four language pairs and over all the pretraining epochs, this strategy leads to an absolute improvement of the average test set word error rate and code-switched bigram error of 1.65% and 1.26% compared to the baseline (ASR$_{\mathrm{B}}$) respectively,

Table 5. Development set perplexity (PP) and perplexity over code-switches (CPP) during the optimisation of the pretraining strategy for each of the respective language pairs: English-isiZulu (EZ), English-isiXhosa (EX), English-Sesotho (ES), and English-Setswana (ET).

Label	Interleaving Strategy	Pretrain Epochs	EZ		EX		ES		ET	
			PP	CPP	PP	CPP	PP	CPP	PP	CPP
N-LM$_B$		baseline	1128.03	7096.29	931.15	8304.73	382.63	3125.48	217.98	1540.83
N-LM$_S$	-	1	**1850.37**	13311.93	**1614.64**	59474.81	**561.98**	4650.30	**297.12**	2417.28
		2	2048.55	17436.57	1650.34	54905.99	610.43	5420.13	300.00	2737.39
		3	1998.27	16713.30	1652.17	45977.33	618.98	4383.20	322.15	3208.09
		4	2098.93	17396.86	1699.18	36098.41	602.49	4594.89	324.74	2482.70
		5	2187.36	19880.80	1851.82	59169.59	631.43	5092.63	325.94	2681.45
		avg	2036.696	16947.892	1693.63	51125.226	605.062	4828.23	313.99	2705.382
N-LM$_M$	$\mathbb{S}$	1	509.02	5074.23	**518.32**	25010.68	291.01	2539.42	**151.55**	1177.82
		2	497.80	5601.93	533.98	25526.28	272.05	2502.47	155.16	1267.82
		3	**475.92**	5721.63	524.00	31449.95	**255.68**	2492.32	156.67	1529.83
		4	483.95	6755.50	549.62	33120.62	257.26	2640.20	157.40	1434.55
		5	481.83	6394.88	564.92	29463.09	266.17	2514.81	158.58	1547.93
		avg	489.704	5909.63	538.17	28914.12	268.43	2537.84	155.87	1391.59
N-LM$_M$	$\mathbb{B}$	1	571.18	5494.01	**547.90**	19960.04	322.25	3082.60	167.14	1689.05
		2	553.30	6428.99	565.68	24943.19	274.48	2902.02	**164.13**	1556.71
		3	**532.67**	6391.08	593.91	21960.06	273.58	2370.09	166.29	1620.73
		4	557.46	7651.09	588.64	28318.79	271.80	2674.29	166.61	1857.61
		5	573.03	8382.29	609.99	24683.29	**264.45**	2371.42	171.15	2053.57
		avg	557.53	6869.49	581.22	23973.07	281.31	2680.08	167.06	1755.53
N-LM$_{S+M}$	$\mathbb{S}$	1	**781.01**	8943.00	**689.57**	36081.04	302.86	2770.62	**176.52**	1657.77
		2	859.46	11458.10	802.73	51184.45	**291.72**	2885.21	192.29	1881.91
		3	926.41	12613.78	869.72	64164.98	311.49	3526.64	197.82	2039.33
		4	928.10	13214.15	930.73	60961.38	325.22	4223.84	208.19	2422.76
		5	973.04	11998.12	971.92	80515.01	347.04	4047.01	213.72	2544.30
		avg	893.60	11645.43	852.93	58581.37	315.67	3490.66	197.71	2109.21
N-LM$_{S+M}$	$\mathbb{B}$	1	**836.48**	9139.47	**846.31**	49030.77	321.34	2940.73	**180.66**	1688.67
		2	944.16	10014.31	939.02	56308.11	**308.64**	3171.25	194.83	2143.43
		3	958.63	10209.21	964.50	61227.18	310.26	3493.94	203.00	2370.01
		4	1013.20	11586.97	1063.56	78132.84	315.36	3376.50	211.92	2505.06
		5	1081.88	13051.23	1088.47	66280.58	335.33	4088.15	226.31	2847.04
		avg	966.87	10800.24	980.37	62195.90	318.19	3414.11	203.34	2310.84

outperforming the batch level interleaving, which afforded average improvements of 1.56% and 1.13%.

When pretraining on the synthetic code-switched data, it appears from the results in both Table 5 and Figure 2 that subsequent fine-tuning on the soap opera data is not successful. In fact, columns S and S_M of the figure make it clear that for all four languages the code-switched losses immediately diverge when fine tuning begins. We believe this may be caused by over-fitting on the synthetic data, therefore we investigated pretraining for fewer batches (1,100,500, and 1000). However, we found that models incorporating the synthetic data are still outperformed by those pretrained on the monolingual data. When pretraining on the monolingual data, it is clear that the performance over code-switches is poor. However, during fine-tuning, performance improves. In fact, these models (M) exhibit better performance over code-switches than the models which are exposed to synthetic code-switched data during pretraining (S and S_M).

We therefore conclude that the best pretraining strategy in terms of speech recognition performance is to mix the monolingual datasets at the sequence level, and pretrain for three

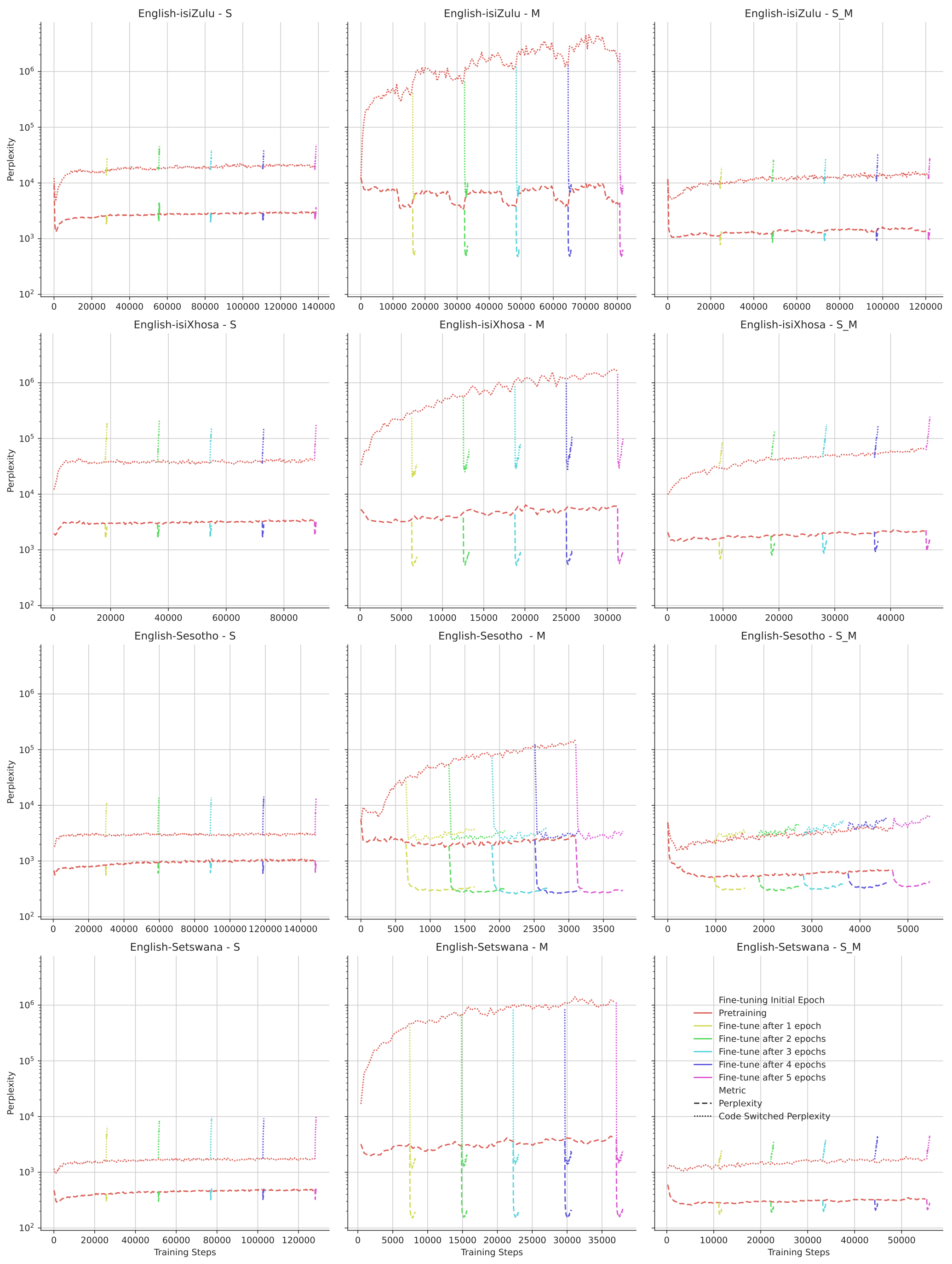

Figure 2. Development set perplexity and code-switched perplexity for each language pair (in rows: English-isiZulu, English-isiXhosa, English-Sesotho, and English-Setswana) and for each of the combinations of corpora used for pretraining as outlined in Table 4 (in columns: S: Synthesized, M: Monolingual, S_M: Synthesized and Monolingual). The red plot indicates the loss throughout pretraining, and each alternate colour corresponds to the loss when fine-tuning is started after pretraining for between one and five epochs.

Table 6. Test set word error rates (WER) and code-switched bigram error rates (CSBG) during the optimisation of the pretraining strategy for each of the respective language pairs: English-isiZulu (EZ), English-isiXhosa (EX), English-Sesotho (ES), and English-Setswana (ET). The epochs which afforded the lowest development set word error rate are highlighted. The best average performance over the five pretraining epochs across the different strategies on the test set is denoted in bold.

Label	Interleaving Strategy	Pretrain Epochs	EZ		EX		ES		ET	
			WER	CSBG	WER	CSBG	WER	CSBG	WER	CSBG
ASR_B		baseline	41.75	63.73	42.89	69.46	50.64	67.17	41.89	57.37
$N\text{-}LM_B$		baseline	40.84	61.20	43.04	69.32	50.07	68.30	41.00	55.22
$N\text{-}LM_S$	-	1	42.03	62.91	43.04	69.90	51.13	69.17	41.49	56.29
		2	41.38	62.43	43.27	70.19	51.70	69.55	42.26	57.85
		3	41.85	62.30	43.68	71.35	51.48	69.30	41.36	56.20
		4	41.53	62.50	44.10	71.49	51.78	69.42	41.67	56.39
		5	41.46	62.50	44.44	71.64	51.31	69.05	41.63	55.32
		avg	41.65	62.53	43.71	70.91	51.48	69.30	41.68	56.41
$N\text{-}LM_M$	$\mathbb{S}$	1	39.25	61.27	43.30	72.07	50.07	67.79	39.12	52.68
		2	39.15	61.00	43.04	71.64	49.38	67.54	39.36	53.76
		3	38.58	59.70	42.78	69.90	49.48	68.30	38.73	51.90
		4	39.08	60.93	42.47	71.20	49.51	67.42	39.42	54.05
		5	39.18	61.68	42.21	69.90	49.48	67.42	39.32	53.37
		avg	**39.05**	**60.92**	42.76	70.94	49.58	67.69	**39.19**	**53.15**
$N\text{-}LM_M$	$\mathbb{B}$	1	39.75	60.86	43.30	70.91	49.06	67.54	39.93	55.90
		2	39.38	61.41	42.81	70.04	49.36	67.29	39.22	53.37
		3	39.40	61.27	42.47	69.75	48.99	67.29	39.28	54.73
		4	39.38	60.52	42.74	70.62	49.06	66.79	39.62	54.15
		5	39.31	60.59	42.66	70.62	49.58	68.05	39.44	54.34
		avg	39.44	60.93	42.80	70.40	**49.21**	67.39	39.50	54.50
$N\text{-}LM_{S+M}$	$\mathbb{S}$	1	40.35	61.48	42.14	70.04	50.15	67.42	39.18	54.54
		2	40.65	62.30	42.66	70.04	49.48	66.92	39.14	53.27
		3	40.53	62.02	42.44	69.75	49.38	68.05	40.11	53.95
		4	40.92	62.16	42.89	70.33	49.70	67.42	40.03	54.93
		5	40.58	61.82	43.08	71.49	50.30	69.05	40.23	54.83
		avg	40.61	61.96	**42.64**	70.33	49.80	67.77	39.74	54.30
$N\text{-}LM_{S+M}$	$\mathbb{B}$	1	40.46	62.30	42.78	69.90	49.43	66.92	39.20	53.27
		2	40.44	61.95	42.70	70.62	49.04	67.54	40.49	55.12
		3	40.51	62.23	42.93	69.75	49.65	67.29	40.51	54.63
		4	40.67	62.23	43.04	70.19	49.70	67.04	40.66	55.41
		5	40.16	61.68	43.08	70.62	49.63	67.04	40.45	54.24
		avg	40.45	62.08	42.91	**70.22**	49.49	**67.17**	40.26	54.53

or four epochs. Given this strategy, selecting the model with the best development set word error rate over the five pretraining epochs (highlighted in Table 6) affords test set absolute word error rate improvements compared to the baseline speech recognition system (ASR-B), as outlined in Section 4.2, of 3.17%, 0.42%, 1.16%, 2.47% for isiZulu, isiXhosa, Sesotho, and Setswana respectively. We note deterioration in speech recognition at code-switches (CSBG) for isiXhosa and Sesotho of 1.74% and 1.13% respectively, while isiZulu and Setswana are improved by 4.03% and 3.32%.

5.2. N-gram Augmentation

In Table 7, we present both the language model perplexities and speech recognition word error rates when utilising the interpolated n-gram language models trained on the respective corpora outlined in Table 4. We find that the interpolated models which incorporate n-grams from the soap opera data, the synthetic code switched data, and the monolingual data (LM_{B+S+M}) on average afford the largest improvement in development set word error rate as well as perplexity, and improve the test set word error rate by between 1.97-3.24% absolute compared to the baseline (ASR_B). Additionally, absolute improvements in code-switched bigram error compared to the baseline of 1.23% and 3.19% are achieved for isiZulu and Setswana respectively, while isiXhosa and Sesotho deteriorate by 0.29% and 0.75% respectively. By incorporating the additional monolingual and synthetic data to train an interpolated n-gram model we are able to consistently improve absolute recognition accuracy on average by 2.53% compared to the baseline model.

Table 7 also shows the results of rescoring the n-best lists with the best performing pretraining strategy (N-LM$_M$) identified in Section 5.1. Specifically, the table presents the test set results corresponding to the best development set word error rate from the five pretrained and fine-tuned models - N-LM$_M$-$\mathbb{S}$ in Table 6. We rescore the n-best lists generated by both the n-gram models trained using the soap opera and monolingual data (LM$_{B+M}$), as well as those that incorporated the soap opera, monolingual and synthetic data (LM$_{B+S+M}$). On average, over the four language pairs, we find that rescoring the hypotheses generated using the n-gram incorporating the soap opera, monolingual and synthetic data (LM$_{B+S+M}$) lead to the largest improvements in the development set word error rate compared to the baseline (ASR$_B$). These language models achieved corresponding improvements in test set word error rate of 3.5% on average compared to the baseline, and 0.98% over the n-gram trained using the soap opera, synthetic and monolingual data. We find that including the synthetic data improves both language modelling and speech recognition when achieved by n-gram augmentation. This is in contrast to the lack of improvement seen when the same data is used in LSTM pretraining. This is consistent with our expectation, as the data was optimised specifically to improve speech recognition when incorporated by n-gram augmentation.

It is also clear that the improvements afforded by rerunning speech recognition experiments after n-gram augmentation are similar to those afforded by the rescoring experiments, which suggests that especially in computationally constrained settings, the augmentation of only the n-gram models for lattice generation should be favoured since it is far less computationally expensive to implement. More specifically, when comparing the results achieved when the optimally pretrained model is used to rescore the baseline n-best hypotheses (ASR$_B$ + N-LM$_M$ in Table 7) to those achieved by utilising the same data for n-gram augmentation (LM$_{B+M}$) we find that on average over the four language pairs, the n-gram augmentation outperforms the rescoring by 0.69% absolute in overall speech recognition accuracy. However, the rescoring method outperforms the n-gram augmentation in terms of code-switched recognition accuracy by 1.38% absolute on average. This suggests that the neural language models are better able to model the code-switching phenomenon, while the augmented n-grams improve the modelling of monolingual stretches of speech.

Overall, rescoring the n-best lists produced by the augmented n-gram LM$_{B+S+M}$ produces the lowest development set word error rate for isiZulu and Setswana, with corresponding test set improvements of 4.23% and 4.45% absolute compared to the baseline (ASR$_B$). Additionally, we improve the speech recognition accuracy over code-switches for the same languages by 2.05% and 3.13% absolute compared to the baseline. The best development set word error rate for isiXhosa and Sesotho is achieved by rescoring the n-best lists produced by n-gram LM$_{B+M}$, leading to test set improvements of 1.81% and 2.69% absolute compared to the baseline. We find, however, that speech recognition at code-switches is worse for both of these languages than the baseline. We conclude that utilising the additional data for both n-gram augmentation (LM$_{B+S+M}$) and LSTM pretraining (N-LM$_M$) for n-best rescoring offers the largest consistent improvements in speech recognition accuracy, and outperforms either strategy employed alone.

5.3. Large Pretrained Language Models

In Table 8 we present the overall test set speech recognition error rate (WER) as well as the speech recognition error rate specifically over code-switches (CSBG) for the five considered pretrained architectures, as discussed in Section 4.6. Each pretrained model is used in rescoring experiments in either a zero shot setting ($\mathbb{Z}$) or after fine-tuning for between one to ten epochs on either the respective bilingual corpus ($\mathbb{F}$-$\mathbb{B}$) or the pooled data

Table 7. Development set perplexity (PP) and code-switched perplexity (CPP) of the augmented n-gram language models (LM) for each of the four respective language pairs: English-isiZulu (EZ), English-isiXhosa (EX), English-Sesotho (ES), and English-Setswana (ET). The test set speech recognition error rates (WER) and code switched bigram error rates (CSBG) when utilising the augmented n-gram language models and additional rescoring by the pretrained neural language model (+N-LM) or multilingual language model (+ GPT-2 (𝔽-ℙ) or + M-BERT (𝔽-ℙ)) are also shown. The model which afforded the best average speech recognition performance on the development set is highlighted. The best performance achieved on the test set is denoted in bold.

Label	EZ		EX		ES		ET	
	PP	CPP	PP	CPP	PP	CPP	PP	CPP
LM_B	565.44	2606.7	418.04	3433.36	235.99	1143.3	174.51	836.91
LM_{B+S}	515.07	**2127.7**	409.94	3356.73	227.39	1024.28	160.81	**724.57**
LM_{B+M}	432.83	2702.94	330.12	3298.88	207.68	1086.17	150.75	864.44
LM_{B+S+M}	**407.78**	2220.83	**328.69**	**3236.44**	**203.21**	**998.71**	**142.38**	730.85
	WER	CSBG	WER	CSBG	WER	CSBG	WER	CSBG
ASR_B	41.75	63.73	42.89	69.46	50.64	67.17	41.89	57.37
+ $N\text{-}LM_M$	38.58	59.70	42.47	71.20	49.48	68.30	39.42	54.05
LM_{B+S}	41.41	62.64	42.66	68.60	50.69	67.92	41.24	56.29
LM_{B+M}	38.12	63.11	40.21	70.04	48.79	68.92	40.07	56.68
+ $N\text{-}LM_M$	37.15	61.54	41.08	71.35	47.95	67.67	37.92	54.54
LM_{B+S+M}	38.51	62.5	40.44	69.75	48.67	67.92	39.42	54.18
+ $N\text{-}LM_M$	37.52	61.68	41.19	71.20	46.99	66.04	37.44	54.24
+ GPT-2 (𝔽-ℙ)	36.59	59.84	39.08	69.03	48.00	67.04	37.94	53.37
+ M-BERT (𝔽-ℙ)	37.36	59.08	**38.82**	**67.44**	46.15	64.54	37.05	**52.59**
+ M-BERT (𝔽-ℙ) + $N\text{-}LM_M$	**36.32**	**59.02**	40.32	69.90	**45.98**	**64.16**	**36.85**	53.56

Table 8. Test set word error rates (WER) and code-switched bigram error rates (CSBG) when utilising transformer models for n-best rescoring. Results are presented for each of the four respective language pairs: English-isiZulu (EZ), English-isiXhosa (EX), English-Sesotho (ES), and English-Setswana (ET). The model which afforded the best average speech recognition performance on the development set is highlighted. The best performance achieved on the test set amongst the multilingual models is presented in bold.

Label	Setting	EZ		EX		ES		ET	
		WER	CSBG	WER	CSBG	WER	CSBG	WER	CSBG
ASR_B		41.75	63.73	42.89	69.46	50.64	67.17	41.89	57.37
$N\text{-}LM_B$		40.84	61.20	43.04	69.32	50.07	68.30	41.00	55.22
$N\text{-}LM_M$		38.58	59.70	42.47	71.20	49.48	68.30	39.42	54.05
D-M-BERT	ℤ	41.41	64.14	42.21	69.61	50.35	69.55	41.43	56.78
	𝔽-𝔹	40.84	62.57	41.34	67.87	49.61	67.67	40.72	55.71
	𝔽-ℙ	40.62	62.16	40.81	66.86	48.52	66.54	40.35	54.34
M-BERT	ℤ	41.41	64.14	42.21	69.61	50.35	69.55	41.43	56.78
	𝔽-𝔹	40.40	61.27	40.29	**66.86**	49.31	66.42	39.93	**52.98**
	𝔽-ℙ	39.57	60.38	40.81	67.58	**48.50**	**65.04**	**39.54**	54.05
afriBERTa-S	ℤ	41.23	63.11	41.72	68.89	50.54	67.67	41.67	56.98
	𝔽-𝔹	40.74	62.02	41.98	70.48	49.38	66.17	40.66	55.71
	𝔽-ℙ	40.01	60.52	41.57	67.58	48.82	65.54	40.21	54.05
mMiniLMv2	ℤ	41.57	63.80	42.25	68.31	50.79	68.92	41.73	56.59
	𝔽-𝔹	40.30	60.59	41.95	68.60	50.72	68.67	41.79	56.39
	𝔽-ℙ	40.30	61.41	41.64	67.73	50.42	66.92	40.76	55.61
GPT-2	ℤ	44.95	67.55	45.83	73.95	52.91	69.92	45.41	61.37
	𝔽-𝔹	39.70	60.31	40.89	67.73	50.05	67.04	40.62	55.41
	𝔽-ℙ	**38.97**	**60.11**	**40.06**	68.60	48.91	65.91	39.62	53.27

from all four sub-corpora ($\mathbb{F}$-$\mathbb{P}$). After fine-tuning, the model which afforded the largest development set word error rate improvements over the ten fine-tuning epochs is selected and used to rescore the corresponding test set n-best list. The resulting test set word error rate is listed in the table.

It is clear from Table 8 that marginal average speech recognition improvements in zero-shot rescoring ($\mathbb{Z}$) are possible for all the bidirectional models except GPT-2. On average over the four language pairs, an absolute improvement of 0.4% is achieved compared to the baseline (ASR_B). While better results were achieved by rescoring with the LSTMs trained only on the in domain data (N-LM_B), it is nevertheless interesting that these models, even in zero-shot settings, are able to improve the recognition accuracy. We believe that this might be due to the language agnostic sub-word encoding strategy used by all the large models in Table 8. These encodings allow the models to learn rich embeddings across languages. Unlike our own LSTM language model, which distinguishes words between languages by appended language tags, the sub-word encoding strategies used by the multilingual models do not. We hypothesise that this allows the model to benefit from the additional training data in related target languages. We aim to explore this hypothesis in further research.

When fine-tuning the pretrained models on the pooled bilingual sets of in domain data ($\mathbb{F}$-$\mathbb{P}$), we find that absolute speech recognition accuracy is improved by 1.79% on average compared to the baseline over all the models and the four language pairs, while models fine-tuned on only the bilingual data afford improvements of 1.23% compared to the baseline.

Interestingly, we note that the BERT model trained on 11 African languages (afriBERTa-S) is outperformed by the BERT model (M-BERT) trained on much more text in unrelated languages. Preliminary experiments using the larger afriBERTa-base, which is comparable in model size to M-BERT, indicated even higher word error rates than those achieved with the smaller afriBERTa-S model.

Over all the large transformer models and language pairs, we find that distilled GPT-2 and multilingual BERT, both fine-tuned on the pooled ($\mathbb{F}$-$\mathbb{P}$) soap opera training data, afford the largest improvements in development set speech recognition, both overall and over language switches. In fact, the best fine-tuned GPT-2 model outperforms our own LSTM rescoring model pretrained on the out-of-domain bilingual corpora as outlined in Section 4 by an average of 0.60% absolute on the test set over the four language pairs. Similar trends are seen in terms of code-switched speech recognition error rate, where GPT-2 outperforms the LSTM model by 1.34% absolute on average over the four language pairs. These results are remarkable, given that the data on which GPT-2 was pretrained did not include any of the target languages considered in this work, and uses a vocabulary that is not optimised to represent those same languages.

Furthermore, the multilingual BERT model also affords consistent improvements over all four language pairs in both overall speech recognition and recognition across code-switches. On average over the four language pairs, this model affords overall improvements of 2.18% in test set word error rate, and 2.67% in code-switched bigram error compared to the baseline system (ASR_B).

When rescoring the n-best lists produced using the augmented n-grams ($\text{LM}_\text{B+S+M}$) the performance of M-BERT surpasses the performance of GPT-2. As shown in Table 7, the multilingual BERT model affords, on average over the four language pairs, improvements of 4.66% and 4.45% in development and test set speech recognition accuracy respectively. The improvement in code-switched performance achieved by the bidirectional model is further strengthened, improving the test set baseline recognition performance by 3.52%, outperforming the GPT-2 model by 1.41%.

We conclude that fine-tuning large transformer models pretrained on unrelated languages can improve speech recognition accuracy more effectively than carefully fine-tuned LSTM models pretrained on data in the target languages. In terms of effective use of computational resources this is an encouraging result, because it means that under-resourced

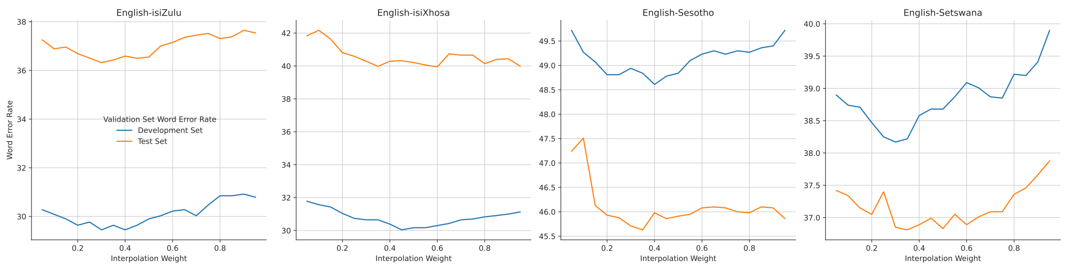

Figure 3. Development and test set word error rates (WER) for each of the respective language pairs (English-isiZulu, English-isiXhosa, English-Sesotho, and English-Setswana) when optimising the interpolation weight between n-best scores produced by the optimal pre-trained LSTM (N-LM$_M$ from Section 5.1) and BERT (M-BERT ($\mathbb{F}$-$\mathbb{P}$) from Section 5.3).

languages can benefit from large models pretrained on well-resourced languages, even when the under-resourced languages are completely unrelated to those used to train the larger models.

5.4. LSTM + BERT Rescoring

In a final set of experiments, we interpolated the n-best scores obtained by the best LSTM (Section 5.1) and best M-BERT (Section 5.3) models. We optimise the interpolation weight over the range 0.05 to 0.95, as shown in Figure 3. In the figure, interpolation weights closer to one assign more weight to the BERT scores, and conversely interpolation weights closer to zero assign more weight to the LSTM scores.

It is clear from Table 7 that utilising a combination of both architectures for rescoring is able to marginally improve (0.4% absolute) overall speech recognition performance for all language pairs except English-isiXhosa. However, recognition performance over code-switches is not improved. Additionally, utilising both models incurs the severe computational overhead of training both architectures, as well as requires each model to rescore the n-best lists.

In future work, we aim to train a large multilingual transformer (comparable to M-BERT) using our South African language data, in order to better assess the performance of the fine-tuned transformer architectures explored here. The LSTM models we have considered receive word-level tokens with language dependent and closed vocabularies, while the transformer models utilise language agnostic sub-word encoding strategies. By closing this gap, we hope to achieve further benefits.

6. Conclusions

In this work we have presented and compared several strategies for pretraining a code-switched neural language model. We found that interleaving distinct pretraining corpora at the sequence level outperformed interleaving at the batch level. Additionally we found that incorporating synthetic data in the pretraining corpora did not afford improvements in speech recognition when the language models are employed in n-best rescoring. We presented the surprising result that, although the data utilised for pretraining our LSTM language model is monolingual, its inclusion allows us to improve speech recognition accuracies, even across language switches.

In contrast, we found that when augmenting n-gram models used for lattice generation with the monolingual and synthetic data we could achieve consistent and comparable improvements, at a fraction of the computational cost of training the neural language model. A combination of n-best rescoring and n-gram augmentation lead to larger improvements in speech recognition accuracies than each approach individually, achieving gains of between 1.81% and 4.45% absolute compared to the baseline.

Finally, we contrasted the improvements afforded by our pretraining strategies to those achieved by fine-tuning large publicly available language models, such as M-BERT.

We found that, even when not specifically trained on data that include the target languages, these models also improve both overall test set speech recognition (by between 4.07%-4.84%) and recognition over code-switches (by between 2.02%-4.78%) compared to the baseline. This result is encouraging, since it represents a means of taking advantage of huge out-of-domain datasets without the need for costly pretraining. Further marginal improvements in speech recognition were achieved by interpolating the n-best scores produced by the best pre-trained LSTM and BERT models. In future work we would like to investigate if these results can be improved upon by direct training of large transformer architectures utilising the monolingual text in the target languages which are available.

Author Contributions: Conceptualization, J.J.v.V. and T.N.; methodology, J.J.v.V. and T.N.; software, J.J.v.V.; validation, J.J.v.V. and T.N.; formal analysis, J.J.v.V.; investigation, J.J.v.V. and T.N.; resources, T.N.; data curation, J.J.v.V.; writing—original draft preparation, J.J.v.V. and T.N.; writing—review and editing, J.J.v.V. and T.N.; visualization, J.J.v.V.; supervision, T.N.; project administration, T.N.; funding acquisition, T.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: We would like to thank the South African Centre for High Performance Computing (CHPC) for providing computational resources on their Lengau cluster for this research. We gratefully acknowledge the support of Telkom South Africa. We also are grateful to the SABC and Human Stark at Generations: The Legacy, as well as e.tv and Yula Quinn at Rhythm City, for assistance with data compilation.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LSTM	Long-short term memory
WER	Word error rate
CSBG	Code-switched bigram error rate
OOV	Out of vocabulary
PP	Perplexity
CPP	Code-switched perplexity
BERT	Bidirectional encoder representations from transformers
GPT	Generative pretrained transformer
LM	Language model
N-LM	Neural language model
ASR	Automatic Speech Recognition

Notes

[1] huggingface.co

References

1. Wills, S.; Uys, P.; van Heerden, C.; Barnard, E. Language Modeling for Speech Analytics in Under-Resourced Languages. Proc. Interspeech; , 2020.
2. Mesham, S.; Hayward, L.; Shapiro, J.; Buys, J. Low-Resource Language Modelling of South African Languages. Proc. 2nd Workshop on African Natural Language Processing, 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL); , 2021.
3. Chang, C.T.; Chuang, S.P.; Lee, H.Y. Code-Switching Sentence Generation by Generative Adversarial Networks and its Application to Data Augmentation. Proc. Interspeech; , 2019.
4. Myers-Scotton, C. *Duelling Languages: Grammatical structure in codeswitching*; Oxford University Press, 1997.
5. Poplack, S. Sometimes I'll start a sentence in Spanish y termino en español: Toward a typology of code-switching. *The Bilingualism Reader* **2000**, *18*, 221–256.

6. van der Westhuizen, E.; Niesler, T. A First South African Corpus of Multilingual Code-switched Soap Opera Speech. Proc. Eleventh International Conference on Language Resources and Evaluation (LREC); , 2018.
7. Garg, S.; Parekh, T.; Jyothi, P. Code-switched Language Models Using Dual RNNs and Same-Source Pretraining. Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP); , 2018.
8. Biswas, A.; Yılmaz, E.; van der Westhuizen, E.; de Wet, F.; Niesler, T. Code-switched automatic speech recognition in five South African languages. *Computer Speech & Language* **2022**, *71*, 101262.
9. Barnard, E.; Davel, M.H.; van Heerden, C.; de Wet, F.; Badenhorst, J. The NCHLT speech corpus of the South African languages. Proc. 4th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU); , 2014.
10. van der Westhuizen, E.; Niesler, T.R. Synthesised bigrams using word embeddings for code-switched ASR of four South African language pairs. *Computer Speech & Language* **2019**, *54*, 151–175.
11. Ogueji, K.; Zhu, Y.; Lin, J. Small Data? No Problem! Exploring the Viability of Pretrained Multilingual Language Models for Low-resourced Languages. Proceedings of the 1st Workshop on Multilingual Representation Learning; , 2021.
12. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NACL); , 2019.
13. Ralethe, S. Adaptation of Deep Bidirectional Transformers for Afrikaans Language. Proc. of the 12th Language Resources and Evaluation Conference; European Language Resources Association: Marseille, France, 2020.
14. Gao, Y.; Feng, J.; Liu, Y.; Hou, L.; Pan, X.; Ma, Y. Code-Switching Sentence Generation by BERT and Generative Adversarial Networks. Proc. Interspeech; , 2019.
15. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. Proc. Advances in neural information processing systems (NIPS); , 2017.
16. Tarunesh, I.; Kumar, S.; Jyothi, P. From Machine Translation to Code-Switching: Generating High-Quality Code-Switched Text. *arXiv preprint arXiv:2107.06483* **2021**.
17. Shin, J.; Lee, Y.; Jung, K. Effective Sentence Scoring Method Using BERT for Speech Recognition. Proc. of the Eleventh Asian Conference on Machine Learning; , 2019.
18. Ortiz, P.; Burud, S. Disambiguation-BERT for N-best Rescoring in Low-Resource Conversational ASR. *arXiv preprint arXiv:2110.02267* **2021**.
19. Jansen van Vueren, J.; Niesler, T. Optimised Code-Switched Language Model Data Augmentation in Four Under-Resourced South African Languages. Proc. SPECOM; , 2021.
20. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. The Kaldi Speech Recognition Toolkit. Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU); , 2011.
21. Biswas, A.; Yilmaz, E.; de Wet, F.; van der Westhuizen, E.; Niesler, T. Semi-supervised Development of ASR Systems for Multilingual Code-switched Speech in Under-resourced Languages. Proc. 12th Language Resources and Evaluation Conference (LREC); , 2020.
22. Stolcke, A. SRILM-an extensible language modeling toolkit. Proc. Seventh International Conference on Spoken Language Processing (ICSLP); , 2002.
23. Martín, A.; Ashish, A.; Paul, B.; Eugene, B.; Zhifeng, C.; Craig, C.; Greg, S.C.; Andy, D.; Jeffrey, D.; Matthieu, D.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. White Paper, 2015.
24. Press, O.; Wolf, L. Using the Output Embedding to Improve Language Models. Proc. of the 15th Conference of the European Chapter of the Association for Computational Linguistics; , 2017.
25. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. Proc. 3rd International Conference on Learning Representations (ICLR); , 2015.
26. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* **2019**.
27. Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; Zhou, M. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. *arXiv preprint arXiv:2002.10957* **2020**.
28. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners **2019**.