Exploration and Experiment of Mandarin-English Code-Switching Automatic Speech Recognition Model

Chunling Song, Wen-Tseng Chang

Institute for Natural Language Processing

University of Stuttgart
SS 2019

Abstract

Code-Switching has always been an interesting language phenomena, especially for ASR system. In this paper, we will present an experiment for the development and optimization of a Mandarin-English code-switching Automatic-Speech-Recognition (ASR) System using Kaldi, an open-source toolkit. A baseline of code-switch ASR model is developed based on the database SEAME, a Mandarin-English Code-Switching Corpus in South-East Asia. Monolingual Mandarin and English Corpus AISHELL2 and LibriSpeech are interpolated for the optimization of the ASR model. Both WER and TER is adopted in the evaluation for the model. In the exploration, challenges of the code-switching ASR system as well as the results of the optimization are observed and analyzed.

Index Terms: speech recognition, code-switching, kaldi, ASR, SEAME

1. Introduction

Code-switching is a special linguistic phenomenon where one speaker uses two or more languages in the same conversation and switch consistently among the languages. This phenomenon often happens with multilingual speakers. In some region of the world, code-switching is very common, such as in south-east Asian countries like Malaysia, Philippines and Singapore.

Common phenomenon as it is, for Automatic Speech Recognition (ASR), code-switching is especially interesting and challenging. Most of the traditional ASR models are built with monolingual data such as English or Mandarin for the recognition of one certain speech. For the recognition of two or more languages, there may be two approaches, one of which is that multiple models can be built and combined to process the different models, and second of which is that one single model can be built. Since code-switching is a very complex linguistic and social phenomenon influenced by many factors such as communication, politics and identity,generation and social relationship etc, it is hard to observe the pattern of the switching of the two languages. In most cases, the switching line between the multiple languages is very vague and there is no clear boundaries in-between (Firqin Sumartono and Ying-Ying Tan, 111).Therefore the second approach of building one ASR system is more reasonable.

In this paper, we adapted the Mandarin-English Code-switching corpus SEAME of data from South-East Asia and attempt to develop one ASR code-switching system. The baseline of ASR model for the Mandarin-English code-switching model is firstly developed with the open-source Kaldi toolkit. In the second part of the experiment, we put our effort in the attempt to optimize the performance of the Model and explore the influence on the code-switch ASR baseline by monolingual corpora of English and Chinese. In the third part, we compared and analyzed the data obtained in the experiment. Finally, a brief conclusion and discussion for further questions is also presented.

2. Resources

In this experiment, for the development and optimization of the code-switching ASR system, three corpora are used, namely the baseline training data set SEAME (South-East Asia Mandarin-English code-switching corpus), AISHELL2 (open source Mandarin speech monolingual corpus) and LibriSpeech (an English monolingual corpus).

2.1. Baseline corpus: SEAME

SEAME corpus is the Mandarin-English code-switching corpus recorded with 156 speakers in Singapore (36.8%) and Malaysia (63.2%) of balanced gender aging from 19 to 33.

SEAME has the code-switching recording of 100 hours. According to the speaking style, The recordings are separated into daily conversation and interview. Each of them has the audio records and transcripts. Audio records are all in flac format with a duration from 20-120 minutes. Transcripts contains all recorded audio texts, including code-switching, Mandarin, and English texts. Content of the recording is casual daily-life dialogues.

2.2. Optimization Mandarin corpus: AISHELL2

AISHELL2 corpus is a monolingual Mandarin corpus, with 1991 speakers of both northern and southern accents and balanced gender of 42.4% male and 57.6% female. Most of the speakers (85.7%) are in the age between 16 to 25.

The recordings are transcribed in Mandarin Chinese. Unlike the more conversational casual topics of SEAME, the speech utterance contains 12 domains, including keywords, voice command, smart home, autonomous driving, industrial production,etc.

2.3. Optimization English corpus: LibriSpeech

The LibriSpeech corpus is derived from audio books that are part of the LibriVox project, which includes large-scale (1000 hours) of English-reading speech.

Since both AISHELL2 and LibriSpeech corpora are only used in optimizing language model. This project only use part of the transcript, which is no longer than the baseline training text.

3. Baseline Development

For the baseline development of the SEAME corpus with Kaldi, we followed the steps of data preparation and processing, Kaldi

dictionary generation, acoustic model, language model and decoding.

3.1. Data Preparation and Processing

For the development of the baseline of SEAME ASR model, We firstly prepared data in order to generate the five data files in the format required by kaldi. The files are text, utt2spk, spk2utt, wav.scp and segments. The optional segments file is included, because in the SEAME Corpus the wav. files (in flac format in SEAME) are not cut line by line. The processing details of the files will be introduced below.

The SEAME corpus provided two different recorded content styles, one is conversation and the other one is Interview. Each context style also includes two different phases: phaseI only includes code-switching format, while phaseII includes monolingual and code-switching format. We choose to use phase II as our data pool since it has more diversity and larger data set.

The original transcript files include 5 columns: audio file name, start time, end time of its wav file, format of the transcript line and one sentence of the transcript. There are three formats of the transcript line, which are English, Mandarin, and code-switching between the former two languages. There is a small sample in the Figure 1:

```
NI67MBQ_0101  14510  16700  ZH  我是 个 大 四 学生 是
NI67MBQ_0101  17300  18620  CS  就是 in mass comm
NI67MBQ_0101  19060  20180  CS  then 我是
NI67MBQ_0101  20760  22970  EN  specialize in electronic broadcast
NI67MBQ_0101  23640  26060  ZH  就是 电子 传 媒 [嘛] 应该 这样 讲
```

Figure 1: *Small Sample from Seame Corpus Transpcript.*

Since the segment file is essential for the SEAME corpus, the start and end time should also be included in the utterance ID. Therefore, the utterance ID shall follow the pattern as <speaker ID>, <audio file name>and the <start and end time>. According to the corpus document, the speaker ID in the interview file is the 3rd to the 6th digit from the audio file name, while in the conversation file the 5th to the 8th digit. Therefore, the utterance ID for the SEAME corpus is the combination of the generated speaker ID combined with the first three columns of the transcript text (see Figure1). Besides, we also set all the time format into 6 digits by adding 0 before the time, as the start and end time are all recorded in mili-seconds.This transformation is realized by turning the strings into digits. An example of the utterance ID and utt2spk is shown in Figure2 and Figure3 respectively.

The text file follows the Kaldi format as <utterance ID>and <normalized sentence>. For the normalization of the sentence, the brackets, and non-speech tags are processed (See Figure 3). The brackets pairs are cleaned and the words are kept for the discourse particles, hesitation and filled pause marks, such as Mandarin segments[ah], [oh], English segments like [oh], [ah] or English sounds like (er), (erm). For the non-speech sound in the original file, the tags such as (ppb) for paralinguistic phenomena breathing,(ppc) for paralinguistic phenomena coughing, (ppl) for the paralinguistic phenomena laughing and (ppo) for others are all generalized into the tag of <pp>, the corresponding tag is also latter added in the lexicon file. Besides, special punctuation is also cleaned. It shall be noticed that due to the mixture of Mandarin and English, both half-width and full-width punctuation are included in the original transcript and the normalization is also conducted accordingly.

```
04NC07FBX_0101  577004  579697  EN      then in the end right what
04NC07FBX_0101  580033  596610  CS      then in the end right (呃) 我 下面 讲 说 [诶] jason
你 应该 那个 戴 dorous 的 眼镜 [啦] then i <unk> specticles for him to 戴 戴 看 [哇] okay [咧
] then 我 讲 改次 你跟 那个 眼镜 店 老板 讲说 (呃) 我 要 小号 小号 一点 的
04NC07FBX_0101  597367  602480  CS      他 的 他 的 很大 spectacles 他 的 这个 框 框 这个
width 很大
04NC07FBX_0101  603118  604577  EN      (oops)
04NC07FBX_0101  607949  614317  EN      his very big [eh] than than mine right on him right
is like just nice [lo] his really is like so big
04NC07FBX_0101  615102  615579  EN      yup
04NC07FBX_0101  618076  624816  EN      (ppb) (ppl) why are we talking about spectacles
(ppb)
04NC07FBX_0101  626660  627501  EN      (erm)
04NC07FBX_0101  628238  630240  ZH      我 有 一个 朋友 她 叫 婉玲 [咯]
04NC07FBX_0101  630240  633320  CS      她 有 short-sightedness 她 戴 contacts [诶]
```

Figure 2: *Phase II transcription in text formatted file*

For the wav.scp files. Since the original audio files are in .flac format. We use ffmpeg to convert them into wav. file for Kaldi to process. The wav.scp is thus prepared in the format of <recording-id><extended-filename>.

The segments file follows the format of <utterance-id><recording-id><segment-begin><segment-end>. The utt2spk follows the format of <utterance-id><speaker-id>, and the spk2utt file is generated from the the spk2utt with the kaldi tool.

After the data preparation, we fix these data without feats.scp and make sure they are all in order and can accepted by Kaldi. The purpose of processing feature extraction is to identify the sound of human speech and discard unnecessary noise. File mfcc and cmvn (feats.scp) are generated because these two features are used for representing the content of each audio utterance. Finally, we fix and validate these data with feats.scp all over again.

	spaker ID	Audio file	Start time	End time	Utterance ID
Conversation	41MB	46NC41MBP_0101	35795	40445	41MB_ 46NC41MBP_0101_ 035795-040445
Interview	01MA	NI01MAX_0101	510440	512186	01MA_ NI01MAX_0101_510440-512186

Figure 3: *Utterance ID.*

Utterance ID	Speaker ID
02FA_NI02FAX_0101_0065144-0068813	02FA

Figure 4: *An Example from utt2spk file.*

3.2. Data Sets

In order to control the variable for the comparison of the model performance in the optimization and also to avoid the data overfitting, the normalized SEAME data were divided into three data sets according to the speaker-ID list after the preparation and processing of the Kaldi files. The three sets are the train sets for language model training, development sets to avoid the data overfitting and the evaluation data sets for the evaluation of the overall performance of the ASR baseline model as well as the optimized models. The same development sets and evaluation sets are also used in the optimization steps in order to have a better comparison. For the acoustic model, a combination of the data set and the development set is used as the train set, in order to make full use of the data available. Therefore in total four data sets are prepared for the acoustic model, language model and the decoding of the SEAME corpus data.

Utterance ID	Text
02FA_NI02FAX_0101_0014985-0018188	又 会 弹钢琴 oh got dancing eh
02FA_NI02FAX_0101_0019545-0025359	muscle 又 很 够力 一下 哇 哦 又 很 大 哦 很 健壮
02FA_NI02FAX_0101_0025894-0028079	又 很 会 跳 他 会 break dance 你 知道 吗

Figure 5: *An Example from text file.*

Utterance ID (without time)	Convert command	File resource
02FA_NI02FAX_0101	ffmpeg -loglevel -8 -i	/resources/asr-data/ LDC2015S04/seame/data/ interview/audio/ NI02FAX_0101.flac -f wav - |

Figure 6: *An Example from wav.scp file.*

3.3. Pronunciation dictionary

Pronunciation dictionary is used to build the acoustic model. It contains words (lexicon) and corresponding pronunciation (phoneme sequence). The dictionary includes 16011 English words and 11383 Chinese words with five tones. In the baseline system, we deleted the tone feature in our pronunciation dictionary in order to avoid the complication with English data.

The baseline pronunciation dictionary is generated by five files: normalized_lexicon, optional_silence, silence_phones, nonsilence_phones, and extra_questions.

The dictionary is cleaned by the following rules:

- The five tones labels in Chinese pronunciation are deleted.
- The punctuation, such as #, %, _ or curly or double curly brackets such as {, {{, }, }} are deleted.
- The unnecessary labels such as 'ME', 'ENG', 'WB' are deleted.
- One non-SAMPA and other two special phones are defined in another label. In former example, schwa is defined to SAMPA label @. The other examples are front_a is defined to fA, open_o is defined to O.

Next, we extract the lexicon list from the clean dictionary. This is our normalized_lexicon.txt

According to the corresponding pronunciation (phoneme sequence) given from the dictionary, the phone set looks like the table below. This phone set is included in nonsilence_phones.txt

Besides, we also add silence phone(sil/sil), unknown word(unk/spn), and noise symbol(pp/nos) in the silence phone set, which is also silence_phones.txt

Table 1: *Phone Set Table*

Phone symbols
a ae ao aw ay b c C ch d dh e E eh er f fA g h i I ih j jh k l m n N o O oy p q r R rI s S sh t th u U uh v w W x y z Z zh @

In optional_silence, it only contains sil, which means it can optionally appear between words. In extra_questions, it simply contains a set of non-silence phones, the silence phone and unknown word.

3.4. Acoustic model

The building of the acoustic model starts from training a context independent monophone system.

Table 2: *Silence Phone Table*

Phone symbols
<sil> sil
<unk> spn
<pp> nos

Given the training data directory (including both training and development data) and dictionary (data/lang), the trained model will be developed and stored in experiment directory (exp/mono).

Next, we do the alignment, since we do not have an alignment model, we use forced alignment. Each time when we train a new model, we will do alignment once again.

First is simplest trigram1 (tri1). Trigram2 (tri2) applied Linear Discriminant Analysis (LDA) and Maximum Likelihood Linear Transform (MLLT). LDA is an dimensionality reduction, which preserves as much of the class discriminatory information as possible.

The final step we train trigram3 (tri3). Besides LDA and MLLT, trigram3 also apply Speaker Adaptive Training (SAT), which is a speaker adaptation techniques.

3.5. Language Model

Language model is a serial of n-gram models that can assign probabilities to word sequences and access the fluency of the output. Every language model has its own perplexity (ppl). The lower the perplexity is, the better the model performs. In order to deal with the problem of data sparsity, smoothing (add-one or Kneser-Ney) or linear interpolation (change the interpolation weight) is also adopted. These solutions can help the model to output a better result.

Besides, SRILM, a toolkit for building and applying statistical language models, is also used in Kaldi for the training of the model. Since the data has already been divided into three sets (train set, development set and evaluation set). In this step,train and development sets are adopted. Firstly, the word list is extracted from the dictionary (data/lang), and the transcript of train and development set is prepared by cutting out the utterance ID, which is not necessary for SIRLM. Language Model is later generated with the prepared data.

The baseline language model of the N-gram (unigram, bigram, and trigram) with both add-one smoothing and Kneser-Ney smoothing is shown in Figure7. As we can see, trigram with Kneser-Ney smoothing has the lowest perplexity and thus the best performance.

	Add-one smoothing		Kneser-Ney smoothing	
Unigram	ppl= 346.842	ppl1= 679.351	ppl= 348.595	ppl1= 683.181
Bigram	ppl= 599.813	ppl1= 1251.18	ppl= 161.232	ppl1= 289.184
Trigram	ppl= 1127.87	ppl1= 2529.81	ppl= 157.367	ppl1= 281.466

Figure 7: *Perplexities of Baseline LMs.*

3.6. Decoding

In the decoding step, fist we make the G.fst graph by importing lexicon dictionary and the best language model, which is the trigram model with Kneser-Ney smoothing. Before decoding step, we need to generate the HCLG graph. H is acoustic

model graph, C is phonetic context, L is lexicon graph, and G is the language model graph. All the four graphs now has been prepared.

After generating the HCLG graph, we start to run decode. In decoding steps, we use the evaluation data set to validate our ASR model. The system will use the acoustic data from the evaluation set to do the speech recognition, generate a hypothesis transcript. This script will be compared to the original evaluation transcript and score word error rate (WER). The WER of this ASR model is 64.53%.

4. Optimization

4.1. Interpolated Language Model

For the optimization of the baseline, we adopt new approach to improve the language model of the baseline by adding monolingual training data. Considering the the characteristics of the code-switching corpus containing two languages, we wonder how the monolingual data will influence the result of ASR model.

The data from AISHELL2 and LibriSpeech with different lines ranging from 15, 20, 25 lines to 9000, 20000, 50000 lines. For each interpolated data, five lambda weight are tested respectively: 0.1, 0.3, 0.5, 0.7, 0.9. Results of the interpolation is discussed below.

4.1.1. Interpolation Results

Figure8&10 shows the result of the interpolated language model of AISHELL2 and LibriSpeech with the lambda weight from 0.1 to 0.9. Figure9&11 are the line charts of the AISHELL2 and Librispeech interpolation showing the changing tendency in comparison with the baseline language model (green line,perplexity 157.367). The x-axis includes all the interpolated data of different number of lines; the y-axis is the perplexities number from 100 to 700. The color lines represent the lambda weight from 0.1 to 0.9. In AISHELL2 corpus, interpolating of 900 lines always has the best result under different weight,although still not better than the baseline system. When we interpolate lambda weight 0.1 (blue line), the result is the worst. However, when we change lambda to 0.3 (orange line), it has a significant improvement. From weight 0.5 to 0.9, the lines come closer to baseline system, but never beyond that.

In LibriSpeech corpus, the trend is different. Under lambda 0.1, 0.3, 0.5, the 4500-line has better results, while under lambda 0.7, 0.9, the 50000-line is much better than the others. Results show that from weight 0.1 to 0.5, the trend is similar to AISHELL2, but from weight 0.7-0.9, the larger amount of interpolated text has better performance. Especially in interpolating 50000-lines with lambda weight 0.9, although the perplexity 160.049 is still lower than baseline 157.367. However, the best WER 64.44% is slightly bigger than baseline 64.53%.

Furthermore, when comparing the performance of AISHELL2 and LibriSpeech, all the AISHELL2 results are better than LibriSpeech. When under the same lambda weight with the same number of interpolated lines, AISHELL2 improves the baseline a bit more than LibriSpeech.

Another thing to be noticed is the best language model with the smoothing methods. In our baseline system, the best performance is trigram Kneser-Ney smoothing. In AISHELL2 interpolation, all the best performance under same lambda weight is the bigram Kneser-Ney smoothing. In the LibriSpeech, the same bigram Kneser-Ney is also the best from lambda 0.1 to 0.7. However, under lambda 0.9, where the interpolated data is much less than others, is trigram Kneser-Ney smoothing the best, as in the baseline.

To conclude, the interpolation of the monolingual data did not successfully optimize the baseline language model. All the interpolated language models show worse performance than the baseline language model, although some are very close to the baseline result. Only one best WER from LibriSpeech 50000 has better performance, even though the perplexities is still lower. Also, interpolated data of different lines all have the best performance under the weight of lambda as 0.9. Besides, no pattern is found for the number of lines for the best performance. The difference between the monolingual English and Mandarin data suggested, though, the difference between the two languages.

	ppl-λ0.1	ppl-λ0.3	ppl-λ0.5	ppl-λ0.7	ppl-λ0.9	Baseline-ppl
Ai15	550.862	285.416	211.85	177.729	160.687	157.367
Ai20	562.935	290.46	214.487	179.075	161.123	157.367
Ai25	551.739	287.614	213.295	178.565	160.991	157.367
Ai30	541.758	285.235	212.383	178.233	160.94	157.367
Ai50	521.172	277.887	208.817	176.572	160.493	157.367
Ai900	472.613	265.98	203.913	174.463	159.865	157.367
Ai4500	473.157	267.592	205.022	175.045	159.947	157.367
Ai9000	473.681	267.714	205.063	175.023	159.884	157.367
Ai20000	477.049	269.562	206.09	175.493	159.943	157.367
Ai50000	489.213	273.68	208.005	176.288	160.019	157.367
Smoothing	2gram.kn022	2gram.kn022	2gram.kn022	2gram.kn022	2gram.kn022	3gram.kn
best_wer	73.82%	68.62%	66.52%	65.32%	64.53%	64.53%

Figure 8: *Perplexities of interpolated AISHELL2.*

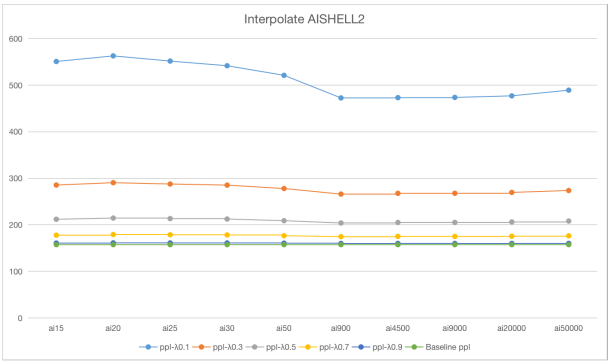

Figure 9: *Perplexities of interpolated AISHELL2.*

4.2. Token Error Rate

Since SEAME is a code-switching corpus with both Mandarin Chinese and English data, the segmentation of words follows the same criteria on the word level, where we separated both Mandarin and English words with a space and computed the Word Error Rate (WER) with Kaldi's decoding tool.

However, for Mandarin Chinese Language, there is no space segmentation between either words and characters in formal writing as in English. In many cases, the segmentation of words is vague and different ways of grouping of words may lead to different calculation of the word error rate. One Mandarin word consists usually two or more Mandarin characters, and these characters in many cases also have independent meanings.

	ppl-λ0.1	ppl-λ0.3	ppl-λ0.5	ppl-λ0.7	ppl-λ0.9	Baseline-ppl
Lib15	659.874	324.596	230.974	186.874	163.341	157.367
Lib20	673.205	326.994	231.701	187.064	163.302	157.367
Lib25	668.617	325.967	231.326	186.922	163.259	157.367
Lib30	670.884	326.77	231.636	187.022	163.276	157.367
Lib50	635.874	316.476	226.948	184.847	162.645	157.367
Lib100	601.409	306.223	222.482	182.904	162.129	157.367
Lib900	563.143	294.734	217.389	180.599	161.395	157.367
Lib4500	557.542	293.028	216.452	179.939	160.902	157.367
Lib9000	562.365	293.905	216.629	179.871	160.724	157.367
Lib20000	569.321	295.495	217.047	179.703	160.418	157.367
Lib50000	581.227	298.205	217.739	179.671	160.049	157.367
Smoothing	2gram.kn022	2gram.kn022	2gram.kn022	2gram.kn012	3gram.kn011	3gram.kn
best_wer	71.54%	67.48%	65.85%	64.90%	64.44%	64.53%

Figure 10: *Perplexities of interpolated LibriSpeech.*

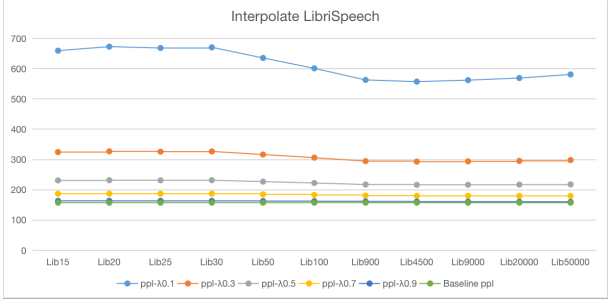

Figure 11: *Perplexities of interpolated LibriSpeech.*

In our the baseline best WER, we noticed the same situation that the segmentation of the Mandarin Chinese words is in some cases random and therefore may influent WER. For example, the hypothesis 'nimennde' and the reference is 'nimen ne' , where the word 'nimen' is correct but not considered a correct word with the insertion 'ne'. Another example is with 'hai meiyou' in reference, but 'haimeiyou' in the hypothesis. Although both 'hai' and 'meiyou' are correct, due to the segmentation, it is both considered to be wrong. Therefore in order to check whether the spacing and grouping of characters have influenced the WER, we also computed the Token Error Rate (TER, or character Error rate).

The hypothesis and evaluation files with the decoding best-WER are taken and spaces were added between the Chinese characters, while the space between the English words remain the same. The TER is improved by around 10 percentage with 54.06%, in comparision with the WER of 64.53%. In order to control the influence of <pp>and <unk>, we also computed the TER without <pp>and <unk>, but the result also improved to 53.69% and 53.30% respectively.

The result confirmed our analysis of the potential influence of Chinese segmentation and also shows the<pp >and <unk >also worsens the ASR system.

5. Analysis & Discussion

5.1. Baseline System

The WER of the code-switching ASR baseline is (64.53%), indicating the performance of this ASR system not very satisfying. When looking into hypothesis text and the reference text, we find that there are many observations worth discussing.

Firstly, since the tones of Mandarin are not added in the dictionary, many Mandarin words with same pronunciation but different tones are not recognized correctly. For example: a Mandarin Chinese name 'Jin(tone1) Fu(tone2)'(is recognized as 'Qin(tone1) Fu(tone4)', which means 'destroyed'; or 'Qin(tone1) Fu(tone2)', which means 'frivolus'.

Secondly, segment words are difficult to be recognized or is recognized as the other language. For example, the special segment 'lah' in South-Asia Mandarin is hard to be recognized or will be mix up with English segment 'ah', but another segment 'meh' has a better performance. Mandarin segments have similar issue such as most of 'ou' cannot be recognized or recognized as English 'oh'; Mandarin segment 'na' cannot be recognized. Besides, a similar Mandarin segment 'hi' cannot be recognized as English word 'hi' as well. However, English segments have better performance than Mandarin segments as well such as 'eh' or 'yeah'. Especially 'yeah', we assume that the reason is that 'yeah' only occur in English dictionary and it does not have close Mandarin pronunciation in this corpus.

Thirdly, as mentioned above, the randomness of the segmentation of the mandarin words may also lead to the high WER, such as in the case of 'hai meiyou' and 'haimeiyou'. This is especially in cases where there are many 'repeat characters' in Mandarin Chinese. In speech recognition, it may cause words error because the system only recognized two or three of them. For example, there is a word 'duo1 duo1 shao3 shao3' in reference text, but recognized two word as 'duo1' and 'duo1 shao3'. In fact, 'duo1', 'duo1 shao3', and 'duo1 duo1 shao3 shao3' are all existed common Chinese words and recorded in the system dictionary. The WER might be lower estimated.

Fourthly, the code-switching makes the recolonization more difficult due to the vague boundary between the two languages. Chinese and English words may be recognized as English word in the middle of the Chinese sentence, such as in the case where the Mandarin 'a' is recognized as the English word 'are'. English words are also in many cases recognized as Mandarin, such as 'gold coast' is recognised as 'bu2 guo4 shi4' or 'guo2 ku4' in Mandarin or 'console' as 'kan3 shu4'. This kind of mix up in a sentence or continuous words. For example 'undergraduate you mean' is mixed up into 'hen3 duo1 ren2 jiu4 yi2 ming2'(many people immigrate) and turn out as a totally different meaning.

Also, <pp>and <unk >are also found to have a high proportion in the hyphoyhesis text. The experiment without<pp >and <unk >both improve slightly to 64.37% and 63.6%.

Finally, there is also different spelling of the same English words, such as ok and "okay". This may also lead to a higher WER for the decoding hypothesis.

5.2. Optimization System

As discussed above in the interpolation result, the interpolation of data from AISHELL2 and LibriSpeech, generally speaking, did not improve the performance of the code-switching ASR model. However, there are still some observations to be discussed.

Since there is the tendency that under lambda 0.9 for the best performance and also the general worse performance after interpolation, we assume that adding monolingual data is not very helpful for the code-switching corpus performance and the small amount of interpolated data may lead to a better performance. However, the result is opposite for both monolingual corpus, where the best interpolated language model has a rather large number of lines interpolated, especially in LibriSpeech

corpus.

Also, Word recognition rate with the interpolated monolingual data seems to decrease. In AISHELL2 interpolated text, some words can be recognized in baseline system with interpolated Mandarin data, such as 'zai4 nan2 da4' (in South University), cannot be recognized in the optimization system. Similarly, after adding LibriSpeech, a few English words can be recognized better than baseline, such as 'semiconductor' was recognized as 'conductor' in baseline system but correctly in optimized language model. However, many English words are still recognized incorrectly, which could be successfully recognized in the baseline system. Furthermore, under the LibriSpeech interpolated model, some English letters are also recognized as words. For example, in the reference text 'b.r.y.a.n' is recognized as 'to be arrow why a and', compared with the 'p r y a n' in baseline model. Another example is 'n.t.u', which can be recognized in baseline system but not the interpolated one.

AISHELL2 also shows some insertion words in the sentence. Because unknown words or non-vocabulary sounds will tend to be translated into wrong Mandarin words.

In short conclusion, after looking at the hypothesis texts, the optimization system cannot improve problems occur in baseline system. By adding monolingual corpus data, it does not help the system to recognize more words, even shows bad influence and reduce the former performance. However, there is still an interesting part from observation of the LibriSpeech data. Although our baseline system is a code-switching corpus, adding almost half amount of the training data makes the performance still close to the baseline. Maybe further testing could show a different tendency from another corpus.

6. Conclusions & Future Work

In this paper, we successfully developed the baseline with the SEAME code-switching data for the ASR of Mandarin-English Recognition, which proves the possibility to train code-switching ASR system with the code-switching data instead of the training with data of two language. Althogh the WER is relatively high and not ideal, the TER considering the segmentation of Chinese characters and the insertion of $<$pp $>$$<$unk $>$has improved the performance by more than 105%.

Besides, in order to optimize and improve the performance, as well as checking the interaction of monolingual language and code-switching data for a better understanding of the code-switching pattern, two monolingual data of Mandarin Chinese and English were chosen and different weight as well as different amount of data are tested in our experiment.

As discussed in the result, the optimization is not ideal with the interpolation of monolingual. Despite the possible influence of the inconsistent topic, this indicated the different linguistic feature underlying the code-switching and monolingual language. It shows that code-switching is not the simple multiplication of two languages, and that the adding of the monolingual data is not helping significantly, if any. Besides, challenges and characteristics of code-switching ASR is also explored in this experiment.

Based on the result and observation of our experiment, more improvement and experiment can be done in the future. For example, One approach is to focus on the acoustic model by adding Mandarin tones in pronunciation dictionary, because, as shown, many Chinese words were wrongly recognized due to the missing of the tones. As for language model, different monolingual data-sets with similar topics of causal conversation as SEAME may be used to check the influence of monolingual data on the baseline system.

Also, a mix text of Mandarin and English with different percentage (as suggested in the best interpolated language model with different number of Mandarin and English lines) can be interpolated to test. Another type of Mandarin-English code-switching corpus can also be interpolated in comparison with the performance of both the baseline and the interpolated system with monolingual data, because the data-set used for the baseline is relatively small. with 100 hours. Much more exploration can be done in the future for the code-switching ASR model.

7. References

[1] SEAME, "Description of the SEAME: Mandarin-English Code-Switching in South-East Asia Corpus," .

[2] AISHELL, "AISHELL-2 Open Source Mandarin Speech Corpus," .

[3] LibriSpeech, "Open Speech and Language Resource," .

[4] F. Sumartono, and Y.Y. Tan, *Juggling Two Languages: Malay-English Bilinguals Code-switching Behavior in Singapore*. Linguistics Journal, July 2018 Volume 12 Issue 1, 2018, pp. 108–138.