Exploring Methods for Building Dialects-Mandarin Code-Mixing Corpora: A Case Study in Taiwanese Hokkien

Sin-En Lu[†], Bo-Han Lu[†], Chao-Yi Lu[‡], Richard Tzong-Han Tsai[†§*]

[†]Department of Computer Science and Information Engineering,
National Central University, Taiwan [‡]Purdue University, USA
[§]Center for GIS, Research Center for Humanities and Social Sciences,
Academia Sinica, Taiwan

alznn3306@gmail.com, 110522028@cc.ncu.edu.tw
lu940@purdue.edu, thtsai@g.ncu.edu.tw

Abstract

In natural language processing (NLP), code-mixing (CM) is a challenging task, especially when the mixed languages include dialects. In Southeast Asian countries such as Singapore, Indonesia, and Malaysia, Hokkien-Mandarin is the most widespread code-mixed language pair among Chinese immigrants, and it is also common in Taiwan. However, dialects such as Hokkien often have a scarcity of resources and the lack of an official writing system, limiting the development of dialect CM research. In this paper, we propose a method to construct a Hokkien-Mandarin CM dataset to mitigate the limitation, overcome the morphological issue under the Sino-Tibetan language family, and offer an efficient Hokkien word segmentation method through a linguistics-based toolkit. Furthermore, we use our proposed dataset and employ transfer learning to train the XLM (cross-lingual language model) for translation tasks. To fit the code-mixing scenario, we adapt XLM slightly. We found that by using linguistic knowledge, rules, and language tags, the model produces good results on CM data translation while maintaining monolingual translation quality.

1 Introduction

Code-switching or code-mixing (CM), which stands for using more than one language in one conversation or sentence, often occurs in multilingual societies. Because of the rapid development of social media, CM has become more prevalent in the past decade, making it be a new challenge in natural language processing (NLP).

Although Mandarin is the dominant language in Taiwan, Taiwanese Hokkien has nearly as many speakers as Mandarin (Liao et al., 2020). Taiwanese tend to mix dialects and Mandarin in daily communication, creating code-mixed languages such as Taiwanese Hokkien-Mandarin or Hakka-Mandarin. Compared with Hokkien, Taiwanese Hokkien integrates Japanese phrases and culture due to historical factors, and gradually evolved into a different dialect. Unlike other CM languages based on recognized writing systems, such as Spanish-English, Hindi-English, or Bahasa Rojak, there were no official writing systems for the dialects in Taiwan until the government built one in the 21st century. Therefore, there is nearly no corpus for Hokkien-Mandarin or other Taiwanese code-mixed languages. Under this circumstance, code-mixing-based NLP tasks (CM tasks) are even more difficult to address in comparison with other monolingual NLP tasks.

The lack of resources (Hedderich et al., 2021), originating from having no formal writing system for Taiwanese Hokkien in the past, makes it hard to make breakthroughs in dialect-related CM tasks. Researchers are often stuck because of lacking corpus to develop deep learning models. Furthermore, not having a writing system increases the possibility of a language's vanishing (Bernard, 1996). Therefore, creating a code-mixed corpus is vital for stepping into the CM-related NLP realm and it also helps to protect the dialects. After creating a code-mixed corpus, more NLP tasks such as machine translation can be developed.

Pre-trained language models have achieved outstanding performance in many NLP tasks. Language models (Devlin et al., 2019; Lample and Conneau, 2019; Liu et al., 2019) have become the mainstream and needful portion in most NLP areas. However, pre-trained language models rely on large-scale corpora, which is a challenge for low-resource languages (Hedderich et al., 2021). Transfer learning is a possible solution because it uses the knowledge from high-resource tasks to improve performance on the related task. It can reduce the amount of required training data and widely improve the effectiveness when solving low-

*Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 6287–6305

resource problems, especially in translation task (Pan and Yang, 2010; Zoph et al., 2016; Hedderich et al., 2021; Wang et al., 2021a). Some research further focused on machine translation, and showed that multilingual models can generalize monolingual inputs to code-switching sentences (Johnson et al., 2017; Pires et al., 2019) without being specifically trained to learn the representations of CM languages.

In this paper, we take Hokkien-Mandarin, a code-mixed language, as the research target because of its large population of speakers in Taiwan. Due to the shortage of Hokkien-Mandarin corpus, we have done several tasks to overcome the low-resource challenge. First, we proposed a method for Hokkien word segmentation via a Mandarin tokenizer toolkit. The method would not be affected by the language morphology and can maintain the syntax structure of Hokkien. It can be seen as the first linguistics-based solution without training a language-specific word segmentation model. Then, we used the Hokkien word segmentation tool to synthesize a Hokkien-Mandarin code-mixed corpus for further use. After that, we proposed a Hokkien-Mandarin cross-lingual language model and achieved good performance on Hokkien-Mandarin CM translation and maintained the monolingual translation result at the same time.

Our main contributions are as follows: (1) We proposed a method of implementing Hokkien word segmentation. (2) We presented a parallel corpus of 76,013 Hokkien-Mandarin CM sentences and 75,150 non-parallel CM data. (3) We built a Hokkien-Mandarin CM translation model through the cross-lingual model.

2 Background of Taiwanese Hokkien

Taiwanese Hokkien, also known as Taiwanese, Hokkien, Taigi, Southern Min, or Min-Nan, is a branched-off variety of Southern Min dialects popular in Taiwan. Under the history background (Chen, 2008), the ability to use Taiwanese Hokkien declines by age (Chen, 2008; Liao et al., 2020; Tan, 2019; of Linguistics at Academia Sinica, 2007; Yang, 2021; Pan, 2016; Ho, 2020). Taiwanese Hokkien has always been the most widely spoken dialect in Taiwan, many people can have conversations in both Mandarin and Taiwanese Hokkien. CM between dialects and Mandarin is a common phenomenon in Taiwan. Previous research shows that CM in Taiwan can be divided into fluent and faltering Hokkien CM scenarios, depending on the individual's ability to master the dialect. Therefore, the degree and proportion of using Hokkien and Mandarin vary from person to person, and there is no universal rule or consensus. In the following paragraphs, we will simply use "Hokkien" to represent "Taiwanese Hokkien".

There are two methods to represent the writing system of Taiwanese Hokkien, logograms, and phonography. A logogram is a written character that represents a word or morpheme. In contrast, phonography is an orthography in which the graphemes correspond to the phonemes of the language. The only logogram writing system is *Written Taiwanese Hokkien* (WTH), which is entirely made of Mandarin characters (Hàn-jī). WTH uses the morpheme and meaning of conventional Mandarin characters instead of their phone to create characters for Taiwanese Hokkien. WTH has an official standard for the writing system and is now taught in schools in Taiwan. On the other hand, there are various phonography writing systems such as *POJ (Peh-ōe-jī)*, *Tai-lo* and *Han-Romanization mixed script (Han-lo)*. Table 1 shows examples of different writing systems in Taiwanese Hokkien.

2.1 Difficulties in Written Taiwanese Hokkien

To eliminate the problems caused by pronunciation diversity in Hokkien and considering that the government in Taiwan is promoting WTH as the main writing system of Hokkien, we use WTH as the main writing system in this research. However, when using WTH to address the CM tasks in Hokkien and Mandarin, we will face two main problems: *ambiguous language boundary* and *literary and colloquial readings* problems.

Ambiguous Word Boundary is caused by the most important feature of WTH. The WTH writing system uses Mandarin characters to represent the meaning of Hokkien, and new characters are created as supplements. Sharing character space reduces the barrier to learning Hokkien, but it also raises a new problem: *The definition of the language boundary is vague when Mandarin and Hokkien are mixed.* The homophones of Hokkien and Mandarin cannot be clearly distinguished by the text alone. Also, the meanings of shared characters may change or disappear when code-switching occurs. Therefore, preprocessing is needed while addressing CM data.

WTH	白話字（POJ）是一款用拉丁（羅馬）拼音系統來寫臺灣的語言的書面文字。因爲當初是傳教士引入來的,所以也有人共POJ叫做教會羅馬字,或者是簡稱教羅。不而過現代的使用者袂少毋是教徒,教徒嘛眞濟袂曉POJ。
Tai-lo	Peh-uē-jī (PUJ) sī tsıt khuán iōng Latin (Lô-má) phìng-im hē-thóng lâi siá Tâi-uân ê gí-giân ê su-bīn bûn-jī. In-uī tong-tshoo sī thuân-kàu-sū ín–jıp-lâi ê, sóo-í iah-ū-lâng kā PUJ kiò-tsò Kàu-huē Lô-má-jī, hek-tsiá sī kán-tshing Kàu-lô. Put-jî-kò hiān-tāi ê sú-iōng-tsiá bē-tsió m-sī kàu-tôo, kàu-tôo mā tsin tsē bē-hiáu PUJ.
POJ	Peh-ōe-jī (POJ) sī chıt khoán iōng Latin (Lô-má) phèng-im hē-thóng lâi siá Tâi-ôan ê gí-giân ê su-bīn bûn-jī. In-ūi tong-chhosī thôan-kàu-sū ín–jıp-lâi ê, só-í iah-ū-lâng kā POJ kiò-chò Kàu-hōe Lô-má-jī, hek-chiá sī kán-chheng Kàu-lô. Put-jî-kò hiān-tāi ê sú-iōng-chiá bē-chió m-sī kàu-tô, kàu-tômā chin chē bē-hiáu POJ.
Han-lo	白話字（POJ）是一款用拉丁（羅馬）拼音系統來寫臺灣ê語言ê書面文字。In-uī當初是傳教士引入來的,所以也有人kā POJ 叫做教會羅馬字,或者是簡稱Kàu-lô。Put-jî-kò 現代ê 使用者bē-tsió毋是教徒,教徒mā眞濟袂曉POJ。

Table 1: Examples of different writing systems in Hokkien

Literary and Colloquial Readings refers to various pronunciations of the same character depending on whether it represents a morpheme in the colloquial or literary lexical layers. This phenomenon is widespread in Sinitic languages and has existed for a long time (Yang, 2015). Therefore, literary and colloquial readings is also one of the characteristics of Hokkien. For instance, the word *"八" (eight)* is written as "*pat*" in literary readings, and written as "*peh*" in colloquial readings. However, since literary pronunciation is usually established by convention, special cases require additional attention to avoid misunderstandings. Due to this phenomenon, many different processing strategies are required.

3 Related Work

3.1 Code-Mixing

CM has been a widely-discussed issue for a long time. There is a wide spectrum of opinions on the reasons and motivation of CM occurrence (Mcclure, 1977; Hoffmann, 1991; Lance, 1970; Aguirre, 1985; Bokamba, 1988; Myers-Scotton, 1993; Sridhar and Sridhar, 1980). To figure out the rules of CM occurrence, research on CM falls essentially into two types: theoretical (also called formal) study and functional study, both proposing different hypotheses and grammatical constraints (Timm, 1975; Poplack, 1980; Pfaff, 1979; Sridhar and Sridhar, 1980). In this paper, we only focus on theories that might be related to our research. The Equivalence Constraint (Poplack, 1978) reports that code-switches tend to occur at points in discourse where the juxtaposition of Language 1 (L1) and Language 2 (L2) elements do not violate a syntactic rule of either language. Poplack (1980) also proffers the Free Morpheme Constraint, which states that the codes in CM language may be switched after any constituent in discourse provided that the constituent is not a bound morpheme. Matrix Language Frame (Joshi, 1982; Myers-Scotton, 1997) defined the dominant language in a CM text as *matrix language*, and other languages are called *inserted languages* or *embedded languages*. All grammar or syntax rules should be under the dominant language. The Functional Head Constraint (Di Sciullo et al., 1986; Belazi et al., 1994a) claims that "the language feature of the complement f-selected[1] by a functional head, like all other relevant features, must match the corresponding feature of that functional head". This means that a language switch between a functional head and its complement does not happen in natural speech. Notice that The Functional Head Constraint should be language-independent.

Both Shih and Su (1995) and Chang (2001) agree that nouns have the largest proportion of transformative words, followed by verbs. Also, adhesive words and function words never appear in language switching and are classified as *function units*. The rest are classified as *content units*. Function units cannot be converted alone while the content unit can be freely converted. In terms of semantics, most conversion words belong to *common expressions* and *common core expressions*. Researchers believe that the reason for language switching is not the lack of vocabulary, but the expression of different social pragmatic functions.

3.2 Pre-trained Language Models

BERT (Devlin et al., 2019) is the first pre-trained language model, which has achieved outstanding performance in the NLP field. Since BERT has made great improvement in the NLP field, using pre-trained language model (Liu et al., 2019; Lewis et al., 2020) has gradually become standard in

[1]The term F-select, or select following F-selection rule, is a feature selection method. Simply put, a sentence satisfies not only the syntax structure but also the semantics.

NLP tasks. Several studies extended the language model to cross-lingual tasks, such as XLM (Lample and Conneau, 2019). XLM is a Transformer (Vaswani et al., 2017) based architecture that was pre-trained with one of three language modeling objectives: Causal Language Modeling (CLM), Masked Language Modeling (MLM), and Translation Language Modeling (TLM). CLM helps the system to learn the probability of a word when given the previous words in a sentence, which can be seen as a causal language model. MLM can be regarded as the Cloze task, the model would randomly mask the tokens in the sequence, and learn to predict the masked tokens. TLM is a translation language modeling objective for improving cross-lingual pre-training. XLM has achieved state-of-the-art performance on multiple cross-lingual understanding (XLU) benchmarks, and has also obtained significant improvement in both supervised and unsupervised neural machine translation tasks.

3.3 NLP task in Code-Mixing

Neural network models for NLP rely on labeled data for effective training (Schuster et al., 2019). To deal with CM tasks with neural networks, it is necessary to prepare a large corpus. Some previous CM research collected the corpora from the real world, such as text messages or the internet. Other research also collected them by manually translating monolingual data to CM data (Singh and Solorio, 2018a; Patra et al., 2018; Lee and Wang, 2015; Sharma et al., 2016; Banerjee et al., 2018; Singh et al., 2018; Dhar et al., 2018; Chakravarthi et al., 2020; Xiang et al., 2020; Srivastava and Singh, 2020). Apart from preparing data manually, a popular strategy for obtaining CM data is through data augmentation.

Pratapa et al. (2018) proposed a method to synthesize CM data which is established on the Equivalence Constraint Theory. The researchers designed a computational approach to create a grammatically valid CM corpus by parsing the pair of equivalent sentences and reducing the perplexity of the RNN-based language model through their proposed dataset. Apart from linguistic theory, there is a lot of research focused on generating data through neural-network-based methods, including GAN (Goodfellow et al., 2014)-based method (Chang et al., 2019; Gao et al., 2019), a deep generative model (Samanta et al., 2019), multi-task learning based (Winata et al., 2018; Gupta et al., 2020), pointer-generator network method (Winata et al., 2019), and regarding generating CM corpus as a translation task (Gupta et al., 2021; Gautam et al., 2021).

Type	Data	#Content
Mono.	Taiwanese songs	30 songs
Mono.	elementary school text books	349 articles
Mono.	Hokkien Reading Competition	550 articles
Mono.	Subtitles of TV programs	126,578 sent.
Para.	iCorpus	64,110 sent.
Para.	MoE's Dictionary (MoeDict)	14,985 sent.

Table 2: Statistics of Hokkien Corpus. *Mono.* refers to monolingual data, and *Para.* refers to parallel data. *Sent.* refers to sentences.

Sinha and Thakur (2005) is one of the earliest CM translation studies which separated CM translation into three parts. First, identify the language of each word. Then, the recognized noun and adverb phrases in one language are translated into the other language. Finally, translate the language-unified sentences to the final target sentence. Rijhwani et al. (2016) put forward a similar concept of Sinha and Thakur (2005), concretely defining the task of each step in a CM translation system architecture. Their idea of CM translation had deeply affected the research of CM translation (Singh and Solorio, 2018b; Rijhwani et al., 2016; Dhar et al., 2018; Mahata et al., 2019; Srivastava and Singh, 2020) in the next few years.

4 Hokkien-Mandarin CM Dataset

Hokkien is one of the most popular dialects in Taiwan (Klöter, 2004; Rubinstein, 2016), and switching between Hokkien and Mandarin is very common. However, to the best of our knowledge, there is no code-mixed dataset for Hokkien and Mandarin. Establishing a CM dataset is a challenge we have to overcome.

In our study, we collect two types of data: monolingual data in Mandarin and Hokkien separately, and parallel data in Hokkien-Mandarin. For the Mandarin corpus, we collect the latest Mandarin corpus from Wikipedia. We also gather Taiwan news as a corpus from 2018 to 2019. We collect 2.2 GB of data for training a Mandarin language model. For Hokkien and Mandarin parallel corpus, we used iCorpus[2] and the example sentences from MoE's Dictionary of Frequently-Used Taiwan

[2] https://github.com/Taiwanese-Corpus/icorpus_ka1_han3-ji7

English	Don't strew things all over the ground.	How much do you make a month?	You don't be so serious with him.
Mandarin	東西不要撒得滿地都是	你一個月賺多少錢？	你不要跟他計較
Hokkien	物件毋通掖甲一四界	你一月日趁偌濟錢？	你毋通佮伊計較
Expected	物件,毋通,掖,甲,一四界	你,一月日,趁,偌濟,錢？	你,毋通,佮,伊,計較
Articut	物件,毋通,掖,甲,一四界	你,一月日,趁,偌濟,錢？	你,毋通,佮,伊,計較

Table 3: Hokkien Sentence Word Segmentation Results in Articut.

Minnan[3] (MoeDict).

4.1 Hokkien Dataset

In Hokkien monolingual dataset, our resources contain Taiwanese songs[4], textbooks of elementary school[5], and articles from Hokkien Reading Competition[6], and the subtitles of Hokkien television program from Chinese Public Television. Note that we only select articles from the Hokkien Reading Competition at the high school level and above, and we excluded data that may contain unrecognized characters or Han-lo script.

For parallel datasets, iCorpus is organized and produced by Academia Sinica, it contains 3,266 news reports from Formosa TV, totaling 64,110 sentences. Including punctuation marks, iCorpus has about 500K Hokkien and 1M Mandarin words. In MoE's Dictionary, there are around 15K example sentences with corresponding Mandarin translations, which were manually created for Hokkien education.

Table 2 shows the statistics of the final Hokkien data. All data are from open-source resources or public government data, collected and organized by the author, and are only used for academic research.

4.2 Hokkien Word Segmentation

Word Segmentation is usually an important step when processing Sinitic languages. However, there is no open-source word segmentation tool for Hokkien, so we need to develop one before other tasks. The most typical word order in Hokkien is *Subject*, *Verb*, and *Object*, which is also the same as that in Mandarin. But there are many sentence patterns with more complicated structures and diverse grammar rules in Hokkien (Tang, 1999). To synthesize Hokkien-Mandarin CM sentences under constraints, it is important to parse the structure of Hokkien sentences precisely. Due to the syntactic complexity of Hokkien, the majority of Mandarin NLP toolkits sometimes provide unexpected results when used with Hokkien sentences. The ability to address unknown words is below our expectations, resulting in losing word boundaries. Also, we may lose the part-of-speech (POS) or syntax information while parsing the sentences. It is difficult to synthesize the CM sentences according to incorrect word boundaries or syntactic parsing results. Our experiment reveals some issues while using BERT-based Mandarin tokenizers, please refer to A.1 for more details.

To capture word boundaries and parse Hokkien syntax, another Mandarin tokenizer, Articut (Wang et al., 2021b), is our solution. There are two reasons why we believe applying Articut to implement Hokkien word segmentation is a potential solution. First, both Hokkien and Mandarin belong to Sino-Tibetan Family. The positions of functional heads in the same language family are almost the same (Tang, 1999). Therefore, the syntax of Hokkien and Mandarin are similar (can even be regarded as the same) in linguistics. Second, the working principle of Articut is the X-bar theory (Chomsky, 1970), which makes it possibly the only tokenizer designed based on linguistics.

According to Chomsky (1970), the X-bar stands for that every phrase in every sentence in every language is arranged in the same way. Each phrase has a head and may include other phrases in the complement or specifier position. X-bar embodies two central principles, Headedness Principle and Binarity Principle. In the Headedness Principle, every phrase has a head. In the binarity principle, every node branches into two different nodes. X-bar relies on these functional heads to check the POS of each word forward and backward. Through the binary structure of the X-bar, Articut can calculate the vocabulary boundary and determine their POS at the same time. As a result, in Hokkien word segmentation, we only need to adjust some "internal order" of morphology. That is, Articut can be

[3] https://github.com/g0v/moedict-webkit
[4] https://github.com/Taiwanese-Corpus/Linya-Huang_2014_taiwanesecharacters
[5] https://github.com/Taiwanese-Corpus/kok4hau7-kho3pun2
[6] https://language110.eduweb.tw/Module/Question/Index.php

adapted to Hokkien, which is mainly designed for Mandarin. Moreover, the issue of morphology can be solved by the custom dictionary provided by Articut.

Second, the Functional Head Constraint proposed by Belazi et al. (1994a) is actually based on the assumption of X-bar theory. The constraint follows Chomsky (1993), assuming that f-selection(Belazi et al., 1994b), a special relationship between a functional head and its complement, is one member of a set of feature-checking processes. In a nutshell, when every noun has a proper position and each functional head works smoothly in a sentence, without exception, a complete syntax tree can be generated.

Therefore, when Articut is processing an input phrase, it checks whether the input satisfies the X-bar theory so that a complete syntactic tree can be generated. After the syntactic tree be successfully created, it signifies that the feature-checking operations have been finished once and the functional head in the sentence can be grasped successfully. Hence we can grab the word boundary and the POS tagging in Hokkien sentences.

We apply MoE's Dictionary of Frequently-Used Taiwan Minnan[7] as the custom dictionary of Articut to implement Hokkien word segmentation. As shown in Table 3, Articut can parse sentences correctly and is not affected by unknown words or morphology. The result verified our conjecture. Through X-bar and a custom dictionary, Articut can correctly identify word boundaries.

4.3 Synthesis of Code-Mixed Corpus

After collecting the corpora, we first normalize the data through the rule-based method to deal with the literary and colloquial reading issue in the Hokkien corpora. We convert words with this issue into Mandarin characters. Furthermore, there are a lot of Tai-lo words that are regarded as noise in our data, so it is necessary to convert them into pure Written Taiwanese Hokkien. The next step is to synthesize a code-mixed corpus. Similar to Pratapa et al. (2018), we synthesize the Hokkien-Mandarin CM dataset based on the matrix language frame, the equivalence constraint, and the functional head constraint. In our work, we defined Hokkien as the matrix language and Mandarin as the embedded language.

Corpus	Sent. Nums.	Symbol Coverage	CMI	SPF
iCorpus-CM	61,690	0.1813	0.571	0.301
iCorpus-CMDA	63,604	0.1525	0.497	0.306
MoeDict-CM	12,409	0.1847	0.483	0.267
MoeDict-CMDA	13,460	0.1907	0.374	0.229

Table 4: Statistics of CM Datasets.

We then applied the equivalence constraint and the functional head constraint to the sentences from the 2 parallel corpora, iCorpus and MoeDict. We first build a Hokkien-Mandarin dictionary based on MoE's Dictionary of frequently used Taiwan Hokkien, and then parse the Hokkien sentence using Articut to get word boundaries and POS tags. Finally, we switch the word to the corresponding Mandarin word according to the dictionary. Under the functional head constraint and the equivalence constraint, the language switch point in our synthetic progress is based on several previous studies (Wu et al., 2011; Shih and Su, 1995; Chang, 2001; Cheng, 1989). The switch point rules are as follows: (1) If **Head Noun** appears in the sentence, switch the **Head Noun**. (2) Switch **Idioms**, but keep the common sayings and proverbs. (3) Switch all **Person** and **Location** in sentence. (4) Switch **Noun Phrase** and **Verb Phrase** in the sentence. (5) Switch the **Noun** after **Preposition**. Lastly, to distinguish the Hokkien and Mandarin, we add _@ after each Hokkien character.

We produce a total of four datasets, iCorpus-CM, iCorpus-CMDA[8], MoeDict-CM and MoeDict-CMDA[9], by two different methods. The first type, iCorpus-CM and MoeDict-CM, means that sentences match all the above rules of switching points and all the constraints, and all the words are precisely translated. The second type, iCorpus-CMDA and MoeDict-CMDA, means that sentences match all of the rules of switching points but not all the constraints, sentences might contain ambiguously translated words. The statistics of four datasets are shown in Table 4. The CM dataset complexity: Switch Point Fraction (SPF) (Pratapa et al., 2018) and Code-Mixing Index (CMI) (Ghosh et al., 2017; Gambäck and Das, 2016), are also reported in Table 4. For the summary of all datasets in our work, please refer to Table 5. And the examples of CM sentences in our data are shown in Table 6.

[7] https://twblg.dict.edu.tw/holodict_new/

[8] DA means Data Augmentation.

[9] https://github.com/alznn/Taiwanese-Hokkien_Mandarin_CM_Dataset

Type	Language	Dataset
Monolingual	Hokkien	Taiwanese songs, elementary school textbooks, Hokkien Reading Competition, subtitles of Hokkien television program
Monolingual	Mandarin	Mandarin Wiki, News
Parallel	Hokkien-Mandarin	iCorpus, MoE's Dictionary
Parallel	CM-Mandarin	iCorpus-CM, iCorpus-CMDA, MoeDict-CM, MoeDict-CMDA,

Table 5: Summary of all Corpora

Corpus	Example
English	Announced at 11 o'clock in the evening Eastern time on the 4th.
Mandarin	在美東時間四日深夜十一時宣布
iCorpus	佇美東時間四號深夜十一點宣布
iCorpus-CM	佇_@ 美東時間 四_@ 號_@ 子夜 十_@ 一_@ 點_@ 宣_@ 布_@
iCorpus-CMDA	佇_@ 美_@ 東_@ 時_@ 間_@ 四_@ 號_@ 深夜 十_@ 一_@ 點_@ 宣_@ 布_@
English	This doesn't work. That doesn't work. You have so many opinions.
Mandarin	這個不行,那個不可以,你的意見眞多。
MoeDict	這个袂使,彼个毋通,全你的意見了了。
MoeDict-CM	這_@ 个_@ 不可 , 彼_@ 个_@ 毋_@ 通_@ , 全_@ 你_@ 的_@ 意見 了_@ 了_@ 。
MoeDict-CMDA	這_@ 个_@ 袂_@ 使_@ , 彼_@ 个_@ 毋_@ 通_@ , 全_@ 你_@ 的_@ 意見 了_@ 了_@ 。

Table 6: CM sentence example in each corpus.

5 Data Quality

5.1 Human Scoring

For human evaluation, we hired three annotators. One of them holds the intermediate level of Hokkien language proficiency certification from the Ministry of Education[10] and the advanced level of Hokkien accreditation from National Cheng-Kung University[11]. The other has the junior-level certification of Hokkien language proficiency from the Ministry of Education. The last one has no certification. The annotators are from three different generations of Taiwanese speakers, elders, middle-aged, and youngsters. We will use annotators A, B, and C to represent each annotator, respectively. All annotators are anonymous and they do not know each other. The annotators are clearly informed of the purpose and use of the entire experiment before the annotating. Each annotator spends 30 hours scoring the data. The annotating costs total $480 US dollar.

The annotators were asked to read 1,879 CM sentences sampled from iCorpus-CM and 179 CM sentences sampled from MoeDict-CM in two phases. In the first phase, the annotators need to score the CM sentence on a scale of 1 (very poor) to 5 (excellent). Our grading criteria are based on the Khanuja et al. (2020) and we widen the score interval. The instructions to the annotators are as follows:

[10] https://blgjts.moe.edu.tw/tmt/index.php

[11] https://ctlt.twl.ncku.edu.tw/gtpt/index.html

1. **Very poor**: The sentence is unreasonable, extremely unnatural, and does not exist in daily life.

2. **Poor**: The sentence is reasonable but slightly unnatural. It takes time to comprehend the sentence's meaning. The sentence structure barely exists.

3. **Fair**: The sentence is reasonable and natural. The sentence is fairly easy to understand and may appear in daily conversations.

4. **Good**: The sentence is reasonable and natural. It is well structured and expected to appear in daily conversations.

5. **Excellent**: The sentence is reasonable and fluent. It is well-structured and often used in daily conversations.

Given that CM behaviors vary from person to person, the scores might be affected by the annotator's life experience. We further ask the annotators to evaluate the sentence in three aspects, colloquialism, coherence, and intelligibility. The metrics were inspired from Banerjee et al. (2018) and have slight modifications on the definition to fit the characteristics of our dataset. The annotators need to follow our instructions and rate each metric on a scale of 1 (poor) to 3 (good). The metrics are described as follows:

1. **Colloquialism**: Check whether the CM sentence is colloquial enough that people may use it in daily life.

	iCorpus (Sent:1879)				MoeDict (Sent:179)			
Annotator	Colloquialism	Intelligibility	Coherence	Total	Colloquialism	Intelligibility	Coherence	Total
A	2.351	2.463	2.422	3.608	2.268	2.503	2.307	3.374
B	2.353	2.470	2.560	3.949	2.223	2.358	2.542	3.810
C	1.884	2.494	2.767	3.537	1.502	2.564	2.721	3.134
Avg.	2.20	2.48	2.58	3.70	2.00	2.47	2.52	3.44

Table 7: Result of human scoring in CM dataset. Annotator A, B, and C represent elders, the middle-aged population, and youngsters, respectively.

2. **Intelligibility**: Check whether the words used in the switching point are correct, including POS and meaning. Ensure all sentences are easy to understand.
3. **Coherence**: Check whether the language switch point is reasonable, the CM sentence was smooth enough and not forced.

The results are shown in Table 7. According to Table 7, it is clear that the evaluation of colloquialism is quite different from the annotator C and others. After investigation, we found that it is because many vocabularies were written in WTH, which the annotator C was not sure how to pronounce. Therefore the annotator C gave a lower score for colloquialism.

We suppose this reflects that even though WTH is designed according to Mandarin characters, it is still hard for understanding the relationship between the spoken language system and the writing system.

5.2 Inter-rater score

We also calculated the kappa value (Fleiss and Cohen, 1973) between our annotators. Similar to Dhar et al. (2018), we randomly selected 100 sentences from our parallel corpus and then requested one of the annotators to translate them into CM sentences to evaluate the reliability of our human evaluation. The other two annotators were assigned to rate the translated sentences into three categories, Totally Agree, Fair Agree, and Disagree. Then, we consider the classification of Agree label as True and Disagree label as False. Finally, we calculate the kappa value by using these labels. The final kappa value is about 0.740. All data we used and created are open source and for academic purposes only.

6 Hokkien Language Model and Translation System

Once we synthesize the code-mixing corpus, we can intuitively apply the data to various NLP tasks. In this study, we chose to apply our CM dataset to the translation task. The main advantage of developing a translation model is that we can translate CM sentences to monolingual sentences and use them as input data for other NLP applications without retraining CM-based NLP models.

Previous research (Johnson et al., 2017; Pires et al., 2019) has pointed out that multilingual models can generalize well on CM data. We believe that since bilingual people can use knowledge of one language to aid the learning of the other, they can identify the language to which the vocabulary belongs in a code-mixed sentence. Also, understand the meaning of the sentence without learning the sentence structure and grammar of the code-mixed language.

Considering the size of our corpus, we use XLM as our multilingual model architecture. Following XLM, we build a vocabulary dictionary at the character level, including all Mandarin and Hokkien characters, Roman numerals, and English letters, for a total of 26,780 characters. Same as our CM dataset, we distinguish Mandarin and Hokkien characters by appending the _@ symbol when building the vocabulary set. We also keep the same special token as XLM. At last, we applied two methods to XLM to verify our hypothesis, dynamic language identification (DLI) mechanism and transfer learning.

Dynamic Language Identification Compared with the pre-defined language embedding input in XLM, the DLI mechanism can dynamically distinguish whether the language is Hokkien or Mandarin through the _@ symbol, and assign the corresponding language embedding to each word. Not only does it suit CM data scenarios where sentences consist of multiple languages, but it also allows XLM to detect which language each word belongs to.

Transfer Learning To figure out if the model can leverage monolingual knowledge to process CM sentences well, we attempt to apply transfer learning to three training objectives (CLM, MLM, TLM) used in XLM. We trained the XLM model

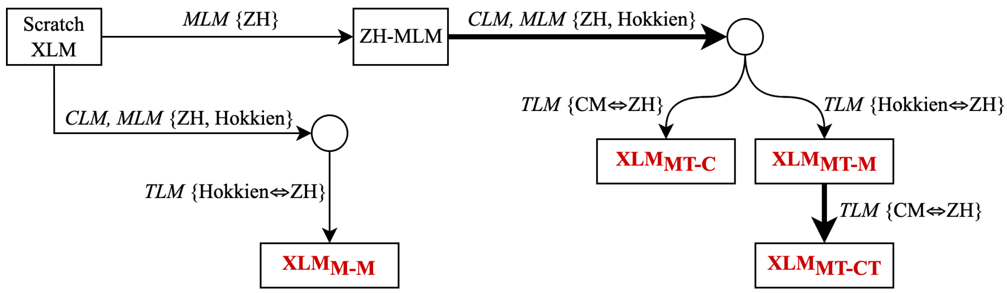

Figure 1: The training process of the model. In $\mathrm{XLM}_{\mathrm{X-Y}}$, we use X for CLM, MLM stage and Y for TLM stage. M: train on monolingual data only. T: utilize transfer learning. C: the training resource contains CM data. (The processes using transfer learning are indicated by bold arrow symbols.)

from scratch using only ZH-Hokkien corpus as our baseline model $\mathrm{XLM}_{\mathrm{M-M}}$. The training process for other models is divided into three stages. 1) We perform an MLM pre-training strategy on the ZH corpus, called ZH-MLM. 2) We continue pre-training the model from the first stage on the both ZH and Hokkien corpus, using CLM and MLM. 3) Once the model in the second stage converges, we continue training the TLM. We train the TLM model using CM-ZH and Hokkien-ZH parallel data, called $\mathrm{XLM}_{\mathrm{MT-C}}$ and $\mathrm{XLM}_{\mathrm{MT-M}}$ respectively. At last, we utilize the parameters from $\mathrm{XLM}_{\mathrm{MT-M}}$ to continue training TLM on the CM-ZH corpus, called $\mathrm{XLM}_{\mathrm{MT-CT}}$. The process of how we obtain these models is also shown in Figure 1.

7 Experiment

For the dataset, taking the diversity in CM habits into consideration, we sample some data from the Hokkien corpus and synthesize them into the CM data randomly. We shuffle all the sentences and split them into training, validation, and test sets in a ratio of 8:1:1 in each parallel corpus. Moreover, a few parts of our Hokkien corpus which can be found corresponding to Mandarin translation were reserved for testing. There are 823 sentences in total, and we use them to synthesize the CM preserved assessment dataset (PAD) by the same method. PAD is not used in any pre-training stage of the XLM model. The CMI and SPF in PAD data are 0.537 and 0.3, respectively. Our evaluation metrics are BLEU (Papineni et al., 2002) and BERT-Score (Zhang et al., 2020). All experiments were done three times and we report the average score. In the model configuration, we set the dimension of embedding to 768. The rest of the configuration follows the vanilla XLM. All models are trained on NVIDIA 3090 GPU. It took 7 days to train ZH-MLM and about 1 to 3 days for the rest of the models.

Our experiment results show that using DLI brings a positive influence on most configurations of the model. Furthermore, transfer learning in the CLM and MLM stages significantly improves performance. It demonstrates that continuous training with our CM data enables monolingual language models to provide better performance when applied to CM translation tasks. For more details and discussions in our experiment, please refer to A.2.

8 Conclusion

In this paper, we introduce the Hokkien background and the CM phenomenon between dialects and Mandarin in Taiwan. We proposed a Hokkien-Mandarin CM dataset based on the linguist theory and the Hokkien grammar rules. We proposed a solution to the Hokkien word segmentation through a linguistics-based toolkit, Articut. Based on the X-bar theory, we can avoid the negative impact of morphology on the Sino-Tibetan languages. We simply modify the language embedding mechanism and use transfer learning in the XLM model, which performs well on both CM and monolingual translation. We again prove that under a fully trained language model and well-defined language identity, the cross-lingual model can generalize the knowledge to a CM sentence without special training. We also verify the feasibility of linguistic-based background knowledge as a low-resource language solution. After developing the CM corpora and translation systems, it can be further extended to any existing monolingual task. Meanwhile, we can use them to generate speech data, and train a CM speech recognition model.

Limitations

Our research is designed for the Sino-Tibetan language family. However, the language features may not be generalized to other language families well. Furthermore, in our case, Hokkien has officially defined Mandarin characters. Without officially defined characters, it might be difficult to eliminate the differences between writing systems and create useful datasets. Finally, during data construction, we can directly mark the language, so we can assume language identification has 100% accuracy in the translation model. Without a good language identification system, the performance of the translation model might be affected.

Acknowledgements

The authors wish to thank prof. Iû Ún-giân and all previous giants who devote their lives to preserving, studying, and promoting Taiwanese dialects. Special thanks to Wang, Wen-jet, CEO of Droidtown Linguistic Tech. Co., Ltd., who selflessly imparts linguistic knowledge to the authors and shares the design context of Articut. Discuss using Articut to implement Taiwanese Hokkien word segmentation with authors. We want to express our sincere gratitude to Dr. Ching-Ching Lu, and all members of the IISR research team, especially Dr. Yu-Chun Wang, Chun-Kai Wu, Ka-Ming Wong, and In-Zu Gi, for the generous assistance, discussion, and advisement they provided for this paper. This research was supported in part by the National Science and Technology Council under grants 109-2221-E-008 -058 -MY3 and 109-2221-E-008 -058 -MY3.

References

Adalberto Jr Aguirre. 1985. An experimental study of code alternation.

Suman Banerjee, Nikita Moghe, Siddhartha Arora, and Mitesh M. Khapra. 2018. A dataset for building code-mixed goal oriented conversation systems. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3766–3780, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Hedi M Belazi, Edward J Rubin, and Almeida Jacqueline Toribio. 1994a. Code switching and x-bar theory: The functional head constraint. *Linguistic inquiry*, pages 221–237.

Hedi M Belazi, Edward J Rubin, and Almeida Jacqueline Toribio. 1994b. Code switching and x-bar theory: The functional head constraint. *Linguistic inquiry*, pages 221–237.

H Russell Bernard. 1996. Language preservation and publishing. *Indigenous literacies in the Americas: Language planning from the bottom up*, pages 139–156.

Eyamba G Bokamba. 1988. Code-mixing, language variation, and linguistic theory:: Evidence from bantu languages. *Lingua*, 76(1):21–62.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020. Corpus creation for sentiment analysis in code-mixed Tamil-English text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 202–210, Marseille, France. European Language Resources association.

Ching-Ting Chang, Shun-Po Chuang, and Hung yi Lee. 2019. Code-switching sentence generation by generative adversarial networks and its application to data augmentation. In *INTERSPEECH*.

Shu-chen Chang. 2001. Code-mixing of english and taiwanese in mandarin discourse.

Mei-Ling Chen. 2008. Code-switching in mandarin and taiwan southern min：a case study of two tv talk shows. *Linguistics at National Tsing Hua University*, pages 1–131.

Yuh-show Cheng. 1989. A preliminar synctactic study on mandarin/taiwanese code switching. *Unpublished MA thesis. National Taiwan Normal University*.

Noam Chomsky. 1970. Remarks on nominalization. ra jacobs & ps rosembaum (eds.), readings in english transformational grammar.

Noam Chomsky. 1993. A minimalist program for linguistic theory. *The view from Building 20: Essays in linguistics in honor of Sylvain Bromberger*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mrinal Dhar, Vaibhav Kumar, and Manish Shrivastava. 2018. Enabling code-mixed translation: Parallel corpus creation and MT augmentation approach. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 131–140, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Anne-Marie Di Sciullo, Pieter Muysken, and Rajendra Singh. 1986. Government and code-mixing1. *Journal of linguistics*, 22(1):1–24.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619.

Björn Gambäck and Amitava Das. 2016. Comparing the level of code-switching in corpora. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1850–1855, Portorož, Slovenia. European Language Resources Association (ELRA).

Yingying Gao, Junlan Feng, Ying Liu, Leijing Hou, Xin Pan, and Yong Ma. 2019. Code-switching sentence generation by bert and generative adversarial networks. In *INTERSPEECH*, pages 3525–3529.

Devansh Gautam, Prashant Kodali, Kshitij Gupta, Anmol Goel, Manish Shrivastava, and Ponnurangam Kumaraguru. 2021. CoMeT: Towards code-mixed translation using parallel monolingual sentences. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 47–55, Online. Association for Computational Linguistics.

Souvick Ghosh, Satanu Ghosh, and Dipankar Das. 2017. Complexity metric for code-mixed social media text. *Computación y Sistemas*, 21.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Abhirut Gupta, Aditya Vavre, and Sunita Sarawagi. 2021. Training data augmentation for code-mixed translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5760–5766, Online. Association for Computational Linguistics.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2020. A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2267–2280, Online. Association for Computational Linguistics.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.

Pei-Hsuan Ho. 2020. *Code-Mixing of Taiwan Mandarin and Southern-Min: A Case Study of the Use of Hybrid Words*.

C. Hoffmann. 1991. *An Introduction to Bilingualism*. Longman linguistics library. Longman.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Aravind Joshi. 1982. Processing of sentences with intrasentential code-switching. In *Coling 1982: Proceedings of the Ninth International Conference on Computational Linguistics*.

Simran Khanuja, Sandipan Dandapat, Sunayana Sitaram, and Monojit Choudhury. 2020. A new dataset for natural language inference from code-mixed conversations. *arXiv preprint arXiv:2004.05051*.

Henning Klöter. 2004. Language policy in the kmt and dpp eras. *China Perspectives*, 2004(56).

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Donald M Lance. 1970. The codes of the spanish-english bilingual. *TESOL Quarterly*, pages 343–351.

Sophia Lee and Zhongqing Wang. 2015. Emotion in code-switching texts: Corpus construction and analysis. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 91–99, Beijing, China. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yuan-Fu Liao, Chia-Yu Chang, Hak-Khiam Tiun, Huang-Lan Su, Hui-Lu Khoo, Jane S. Tsay, Le-Kun Tan, Peter Kang, Tsun-guan Thiann, Un-Gian Iunn, Jyh-Her Yang, and Chih-Neng Liang. 2020. Formosa speech recognition challenge 2020 and taiwanese across taiwan corpus. In *2020 23rd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*, pages 65–70.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sainik Kumar Mahata, Soumil Mandal, Dipankar Das, and Sivaji Bandyopadhyay. 2019. Code-mixed to monolingual translation framework. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, pages 30–35.

Erica Mcclure. 1977. Aspects of code-switching in the discourse of bilingual mexican-american children. technical report no. 44.

Carol Myers-Scotton. 1993. Common and uncommon ground: Social and structural factors in codeswitching. *Language in society*, 22(4):475–503.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Institute of Linguistics at Academia Sinica, editor. 2007. *Yu yan zheng ce de duo yuan wen hua si kao / Zheng Jinquan [and others] bian ji.*, chu ban. edition. Yu yan, she hui yu wen hua xi lie cong shu ; 2. Zhong yang yan jiu yuan yu yuan xue yan jiu suo, Taibei Shi.

Hue-Hua Pan. 2016. Localization of language usage in ethnic chinese media and hybridity in southern min soap operas and movies of taiwan, singapore, and china. *Taiwan International Studies Quarterly*, 12(3):173–206.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task @icon-2017. *CoRR*, abs/1803.06745.

Carol W Pfaff. 1979. Constraints on language mixing: Intrasentential code-switching and borrowing in spanish/english. *Language*, pages 291–318.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Shana Poplack. 1978. *Syntactic structure and social function of code-switching*, volume 2. Centro de Estudios Puertorriqueños,[City University of New York].

Shana Poplack. 1980. Sometimes i'll start a sentence in spanish y termino en espanol: toward a typology of code-switching1.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553, Melbourne, Australia. Association for Computational Linguistics.

Shruti Rijhwani, Royal Sequiera, Monojit Choudhury Choudhury, and Kalika Bali. 2016. Translating codemixed tweets: A language detection based system. In *3rd workshop on Indian language data resource and evaluation-WILDRE-3*, pages 81–82.

Murray A Rubinstein. 2016. *The Other Taiwan, 1945-92*. Routledge.

Bidisha Samanta, Sharmila Reddy, Hussain Jagirdar, Niloy Ganguly, and Soumen Chakrabarti. 2019. A deep generative model for code switched text. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5175–5181. International Joint Conferences on Artificial Intelligence Organization.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3795–3805, Minneapolis, Minnesota. Association for Computational Linguistics.

Arnav Sharma, Sakshi Gupta, Raveesh Motlani, Piyush Bansal, Manish Shrivastava, Radhika Mamidi, and Dipti M. Sharma. 2016. Shallow parsing pipeline - Hindi-English code-mixed social media text. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1340–1345, San Diego, California. Association for Computational Linguistics.

YH Shih and ZZ Su. 1995. A study of mandarin code-mixing in taiwanese speech. In *First International Symposium on Languages in Taiwan, and then collected in The Proceedings of the Symposium*, pages 731–767.

Kushagra Singh, Indira Sen, and Ponnurangam Kumaraguru. 2018. A Twitter corpus for Hindi-English code mixed POS tagging. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 12–17, Melbourne, Australia. Association for Computational Linguistics.

Thoudam Doren Singh and Thamar Solorio. 2018a. Towards translating mixed-code comments from social media. In *Computational Linguistics and Intelligent Text Processing*, pages 457–468, Cham. Springer International Publishing.

Thoudam Doren Singh and Thamar Solorio. 2018b. Towards translating mixed-code comments from social media. In *Computational Linguistics and Intelligent Text Processing*, pages 457–468, Cham. Springer International Publishing.

R. Mahesh K. Sinha and Anil Thakur. 2005. Machine translation of bi-lingual Hindi-English (Hinglish) text. In *Proceedings of Machine Translation Summit X: Papers*, pages 149–156, Phuket, Thailand.

Shikaripur N Sridhar and Kamal K Sridhar. 1980. The syntax and psycholinguistics of bilingual code mixing. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 34(4):407.

Vivek Srivastava and Mayank Singh. 2020. PHINC: A parallel Hinglish social media code-mixed corpus for machine translation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 41–49, Online. Association for Computational Linguistics.

Tik-Siu Tan. 2019. How to write taiwanese: integration, ideology and comparison of various taiwanese writing systems.

Tingchi Tang. 1999. *Minnan yu yu fa yan jiu shi lun*, chu ban edition. Xian dai yu yan xue lun cong. Taiwan xue sheng shu ju.

Lenora A Timm. 1975. Spanish-english code-switching: el porque y how-not-to. *Romance philology*, 28(4):473–482.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Rui Wang, Xu Tan, Renqian Luo, Tao Qin, and Tie-Yan Liu. 2021a. A survey on low-resource neural machine translation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4636–4643. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Wen-jet Wang, Chia-jung Chen, Chia-ming Lee, Chien-yu Lai, and Hsin-hung Lin. 2021b. Articut: Chinese Word Segmentation and POS Tagging System.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Code-switching language modeling using syntax-aware multi-task learning. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 62–67, Melbourne, Australia. Association for Computational Linguistics.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Code-switched language models using neural based synthetic data from parallel sentences. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 271–280, Hong Kong, China. Association for Computational Linguistics.

Yi-Lun Wu, Chaio-Wen Hsieh, Wei-Hsuan Lin, Chun-Yi Liu, and Liang-Chih Yu. 2011. Unknown word extraction from multilingual code-switching sentences in Chinese. In *ROCLING 2011 Poster Papers*, pages 349–360, Taipei, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Rong Xiang, Mingyu Wan, Qi Su, Chu-Ren Huang, and Qin Lu. 2020. Sina Mandarin alphabetical words:a web-driven code-mixing lexical resource. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 833–842, Suzhou, China. Association for Computational Linguistics.

Hsiu Fang Yang. 2015. Wén bái yì dú 文白異讀(literary and colloquial readings).

Yan-Zhang Yang. 2021. A study on the current situation and opportunity of code-switching between mandarin and southern min in taiwanese - using tv programs as an analysis corpus. *Journal of Taiwan studies, Takushoku University*, 5:91–118.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Appendix

A.1 Problems of using Mandarin Toolkit

CKIP[12] is one of the Mandarin NLP toolkits based on BERT. It can be seen as the most robust Mandarin tokenizer toolkit. Our experiment reveals that CKIP may provide unexpected results in Hokkien sentence word segmentation due to the different grammar structure between Hokkien and Mandarin. Some examples of tokenization results are shown in Table 8 and Table 9.

In Table 8, the words in sentence P1 share the same characters and meanings between WTH and

[12] https://github.com/ckiplab/ckiptagger

Mandarin. Most words in sentences P2 and P3 share the same characters and meanings. As for the words with different characters, such as "毋通" corresponding to "不要" in P2, or "目尾" corresponding to "眼角", "皺痕" corresponding to "皺紋" in P3, they have similar meanings ("毋"/"不", "目"/"眼", "皺痕"/"皺紋") in both languages either. Therefore, the word segmentation results of these sentences are in our expectations. We call them positive cases.

Table 9 shows the less ideal examples. The examples N1 and N2 are the word segment results of Hokkien sentences. Each word in these sentences, from the structure of the word to the word itself, has a different meaning in Mandarin. As the phrases "一四界", "一月日", and "偌濟" are used in Hokkien only. Also, it is difficult to associate the meanings of individual characters with the meaning of the phrases. Though the phrases "物件", "掖", and "趁" exist in Mandarin, the meanings in the two languages are very different. Therefore, the results in N1 and N2 are worse than we expected.

N3 displays the word segment outcome when a Hokkien sentence follows the same grammar structure as a Mandarin sentence but contains any newly created words. Term "佮" is a new word designed for Hokkien, and would not appear in Mandarin corpora, which implies it is an unknown word to a BERT-base tokenizer. Compared with the P2 in Table 8, N3 consists of the same sentence structure and overlap word "毋通". The remaining words have the same meaning and vocabulary in Hokkien and Mandarin. However, CKIP couldn't parser the "毋通" and other words correctly.

It can be observed from the results that the BERT-based model tends to process unintelligible vocabulary in a character-base manner. We speculate the reason is that BERT uses a character-based method to split the Mandarin characters. And most BERT-based toolkits rely on "continuous distribution of context relation in high-dimensional vector spaces" to implement word segmentation or other semantic analysis, which brings too much noise while applying these toolkits on Hokkien sentences.

In CKIP, the ability to address unknown words is lower than we expected. BERT-based processing method will make it lose word boundaries and the POS or syntax information while parsing the sentence. So in this case, it would be difficult to synthesize the CM sentences based on the word boundary or syntactic parsing results.

A.2 Model Discussion

A.2.1 Hokkien language model on BERT

In this section, we will explain how to transfer a Mandarin BERT model to train a Hokkien language model. The main reason we need to train a Hokkien language model is that although most of the characters in Written Taiwanese Hokkien and Mandarin are the same, the grammar and meaning are often different. They still need to be regarded as two different languages. Pre-trained language models have become the standard step for NLP tasks today and rely on a large corpus (Hedderich et al., 2021). This implies that training a Hokkien Language Model from scratch is unrealistic. Since Written Taiwanese Hokkien and Mandarin share some characters, transfer learning can be applied to solve the above-mentioned problem. Using a pre-trained Mandarin language model and transferring the parameters to a Hokkien language model could be a potential solution.

There are about 800 characters in Written Taiwanese Hokkien that do not exist in Mandarin. To pre-train a Hokkien language model, we first replace the special tokens and rarely used characters in vanilla BERT's vocabulary dictionary with these newly created Hokkien words. The examples of the replaced vocabulary set are shown in Table 11. Second, we set a higher priority to mask these new words forcing the model to learn these words during the MLM step. Then, we continue training a BERT-based language model with Mandarin model parameters on the monolingual Hokkien corpus.

We trained our model with 500k steps, 16 batch size, 256 max sequence length, 3000 warm-up steps, and $2e{-}5$ learning rate. The Hokkien language model achieves over 78% accuracy on mask word prediction, and the loss function is reduced to $2.23e{-}3$. The examples of the Hokkien language model predicting results are shown in Table 12.

Note that example numbers 1, 2, 6, and 10 are newly created words in Hokkien and the language model can predict them correctly. In the model configuration, we set the dimension embedding to 768. The rest of the configuration follows that of the vanilla XLM. All models are trained on NVIDIA 2080 GPU. It took 12 hours to train the Hokkien LM.

A.2.2 Monolingual Language Model in XLM

The experiment in Section A.2.1 shows that using transfer learning brings a positive effect on

Sent. Number	P1	P2	P3
English	Being a professor is my lifelong wish.	You, don't live a life of debauchery.	I started to have wrinkles in the corners of my eyes.
Mandarin	做教授是我一生的願望。	你不要放蕩過一生。	我的眼角開始有皺紋了。
Hokkien	做教授是我一生的願望。	你毋通放蕩過一生。	我的目尾開始有皺痕矣。
Expected	做,教 授,是,我,一 生,的,願望,。	你,毋通,放蕩,過,一生,。	我,的,目 尾,開 始,有,皺痕,矣,。
CKIP-BERT	做,教 授,是,我,一 生,的,願望,。	你,毋通,放蕩,過,一生,。	我,的,目 尾,開 始,有,皺痕,矣,。
CKIP-ALBERT	做,教 授,是,我,一 生,的,願望,。	你,毋通,放蕩,過,一生,。	我,的,目 尾,開 始,有,皺痕,矣,。

Table 8: Positive cases of Hokkien Sentence Word Segmentation in CKIP.

Sent. Number	N1	N2	N3
English	Don't strew things all over the ground.	How much do you make a month?	You don't be so serious with her.
Mandarin	東西不要撒得滿地都是	你一個月賺多少錢？	你不要跟他計較
Hokkien	物件毋通掖甲一四界	你一月日趁偌濟錢？	你毋通佮伊計較
Expected	物件,毋通,掖,甲,一四界,。	你,一月日,趁,偌濟,錢,？	你,毋通,佮,伊,計較,。
CKIP-BERT	物件,毋,通,掖,甲,一,四,界,。	你,一,月,日,趁,偌濟,錢,？	你,毋,通,佮伊,計較,。
CKIP-ALBERT	物件,毋,通,掖甲一四,界,	你,一月日,趁,偌,濟錢,	你,毋,通,佮伊,計較,。

Table 9: Negative cases of Hokkien Sentence Word Segmentation in CKIP.

Hokkien language model. We then use the same idea to train our Mandarin language model and Hokkien language model from scratch in the XLM. The results of training our monolingual language model in CLM and MLM stages are shown in Table 10.

A.2.3 Using Dynamic Language Identity

In the following XLM experiments, we first show the benefits of applying DLI to XLM. The results are shown in Table 13. Using DLI brings a positive influence in every configuration of the model, especially for model 0 in the configuration of $\mathrm{XLM}_{\mathrm{M-M}}$. Applying DLI brings 5.6 points of BLEU score improvement on the PAD-CM dataset. We believe that when the model is translating without any prior knowledge and the CM data has not been read, it mostly processes the CM sentence based on the language ID. Therefore, after providing the correct language mark, the model can make substantial progress.

A.2.4 Influence of AutoEncoder

In addition, we display the impact of AutoEncoder (AE). AE in XLM randomly selects a character A in the input sentence and replaces it with another character B in the vocabulary set. The model must learn how to restore character B to character A. When A and B are in different languages, the model is actually equivalent to creating a CM sentence as an input sentence automatically. Therefore, this process helps the model learn how to deal with the CM input, even if no parallel CM corpus was used in TLM. Model 2 has the same settings as model 1 but without AE. It is also observed that after using AE, the BLEU score of PAD CM data significantly improves.

A.2.5 Using Transfer Learning

$\mathrm{XLM}_{\mathrm{MT-M}}$ (models 4 to 7) show that using transfer learning in the CLM and MLM stages significantly improves performance. Although we use monolingual datasets, they still have outstanding performance in CM translation. Compared to $\mathrm{XLM}_{\mathrm{MT-C}}$ (model 8 to 11), which are TLM models trained from scratch using the CM corpus, BLUE scores of $\mathrm{XLM}_{\mathrm{MT-M}}$ (model 4 to 7) are only one point lower on each configuration. Moreover, when it comes to the BERT score, which emphasizes semantic meaning, there is almost no difference when the DLI mechanism is applied. This also verifies our hypothesis in section 6. Applying transfer learning to the monolingual language model can be regarded as ***a way for bilinguals to learn multiple languages***. Training the TLM model from scratch by using a parallel dataset without code-mixing, and then testing it on the CM corpus, can be viewed as ***bilinguals understanding the meaning of code-mixed sentences when they have never learned its structure and grammar***. The PAD-CM BLEU Scores of the $\mathrm{XLM}_{\mathrm{MT-M}}$ (model 4 to 7) are no less than those trained directly on the CM corpus (model 8 to model 11) or applied

Model	Setting	PARAMS		Test					
		Dim	Batch Size	zh mlm_ppl	zh mlm_acc	min mlm_ppl	min mlm_acc	min_zh mlm_ppl	min_zh mlm_acc
XLM Scratch	ZH MLM	768	32	3.818	68.949	-	-	3.818	68.949
XLM Scratch	CLM, MLM	768	64	9.796	55.418	8.518	55.293	1.741	88.7494
XLM Transfer	CLM, MLM	768	64	4.794	65.048	5.948	62.522	1.391	92.560

Table 10: Language model in XLM.

index	select word	map word
3	[unused2]	佮
4	[unused3]	个
5	[unused4]	仝
6	[unused5]	囥
7	[unused6]	紲
8	[unused7]	蹛
9	[unused8]	爿
10	[unused9]	徛
11	[unused10]	翕

Table 11: Example of replacing an unused word in the BERT vocabulary.

num.	answer	predict
1	佮	佮
2	仝	仝
3	支	支
4	[PAD]	的
5	保	保
6	佮	佮
7	歇	歇
8	的	暗
9	一	一
10	爿	爿

Table 12: Example of `[MASK]` token predicting results.

transfer learning to further learn the CM corpus (model 12 to 15). $\mathrm{XLM_{MT-CT}}$ (model 12 to 15) illustrated the effect of applying transfer learning to all models in XLM. It brings about a one-point improvement in the BLEU score but faces the risk of overfitting at the same time.

A.3 Case Study

In this section, we will discuss the impact of DLI and transfer learning on the model. We utilize the PAD dataset and present the results in Tables 14, 15 and 16. Since Table 13 shows that overfitting might occur in model 13, $\mathrm{XLM_{MT-CT}}$ with autoencoder and without DLI mechanism, we will neglect the results of this model in the following case studies.

A.3.1 Dynamic Language Identity

We will discuss the mechanism of DLI first. Please refer to Table 14 for more information. The code-mixing sentence in the example switches *"昨天 (yesterday)"* and *"布市 (fabric market)"* to Mandarin. It is obvious that in $\mathrm{XLM_{M-M}}$, model (-)AE(-)DLI and model (-)AE(+)DLI do not converge well and translate *"布市"* to *"別[13]地震 (earthquake)"* and *"零售 (retail)"*, respectively. The former one is illogical and unrelated to the context. As for the latter one, we suspect that this is because *market* and *retail* are slightly related. Model (+)AE(-)DLI translates *"布 (fabric)"* into *"布袋市 (fabric bag market)"*. Since most bags were made of fabric in the past, we refer to "bags" as "fabric bags" in Mandarin. Since *"布 (fabric)"* and *"布袋 (fabric bag)"* are related in Mandarin, the two phrases can be similar in contextual vector spaces. We believe that the model picked the phrase closer to *"布 (fabric)"*.

In $\mathrm{XLM_{MT-M}}$, models that utilize the DLI mechanism are capable of translating or preserving *"布市 (fabric market)"*, but a model that utilizes neither DLI nor AE translates *"布市 (fabric market)"* to *"布帳 (fabric tent)"*. Although it seems to be contextual related just as *"布(fabric)"* does, *"布帳 (fabric tent)"* is extremely uncommon in Mandarin. We suspect that transfer learning enables the models to pick phrases that are contextually related and translate them. Similarly, $\mathrm{XLM_{MT-M}}$ (+)AE(-)DLI model translated *"市 (market)"* to *"市府 (city hall)"*. Because *"市"* also means *city* in Mandarin, we believe it is the reason that the model translates *"布 (fabric) 市 (city)"* to *"布 (fabric) 市府 (city hall)"*.

$\mathrm{XLM_{MT-C}}$ model and $\mathrm{XLM_{MT-CT}}$ model are both capable of translating accurately. We assume this is because the models leverage transfer learning and code-mixed corpora.

A.3.2 Transfer Learning

We want to discuss the impact of transfer learning on models in this section. We sample one sentence from the Hokkien sentences and one from the code-mixing sentences and present the results in Table 15 and Table 16, respectively.

Before looking into Table 15, please note that

[13]The meaning of *"別"* varies depending on the context and cannot be translated by itself.

Num	Name	Config. AE	Config. DLI	Testset Acc	Testset BLEU	PAD Mono BLEU	PAD CM BLEU	PAD CM BERT-Score Precision	PAD CM BERT-Score Recall	PAD CM BERT-Score F1
0	$\text{XLM}_{\text{M-M}}$	+	+	90.898	75.46	51.42	54.87	87.952	89.334	88.585
1	$\text{XLM}_{\text{M-M}}$	+	-	91.149	75.85	50.96	49.23	87.911	89.203	88.503
2	$\text{XLM}_{\text{M-M}}$	-	+	90.841	75.37	52.21	49.54	87.089	87.755	87.387
3	$\text{XLM}_{\text{M-M}}$	-	-	90.736	75.16	49.52	47.56	86.357	87.410	86.847
4	$\text{XLM}_{\text{MT-M}}$	+	+	91.887	83.15	62.59	62.11	90.633	91.645	91.097
5	$\text{XLM}_{\text{MT-M}}$	+	-	91.796	83.03	62.21	61.87	90.122	91.455	90.742
6	$\text{XLM}_{\text{MT-M}}$	-	+	91.440	82.54	61.24	60.25	91.234	91.033	91.105
7	$\text{XLM}_{\text{MT-M}}$	-	-	91.553	82.87	60.05	57.44	89.823	89.752	89.766
8	$\text{XLM}_{\text{MT-C}}$	+	+	91.964	82.92	60.38	62.86	90.670	91.678	91.131
9	$\text{XLM}_{\text{MT-C}}$	+	-	91.951	82.77	59.67	60.74	88.672	90.788	89.656
10	$\text{XLM}_{\text{MT-C}}$	-	+	91.528	82.57	61.24	61.19	91.338	91.122	91.201
11	$\text{XLM}_{\text{MT-C}}$	-	-	91.559	82.58	59.68	60.46	91.003	90.801	90.873
12	$\text{XLM}_{\text{MT-CT}}$	+	+	98.402	95.65	61.18	61.46	90.581	91.416	90.964
13	$\text{XLM}_{\text{MT-CT}}$	+	-	97.285	93.58	30.12	32.79	66.133	76.186	70.561
14	$\text{XLM}_{\text{MT-CT}}$	-	+	98.114	94.58	62.48	62.47	91.502	91.729	91.593
15	$\text{XLM}_{\text{MT-CT}}$	-	-	98.023	94.37	62.36	62.35	91.731	91.807	91.749

Table 13: Result of each configuration in XLM.

Code-mixing Source Sentence: 昨天下_@ 晡_@ 去_@ 布市
Mandarin Target: 昨天下午去布市
English: Went to the fabric market yesterday afternoon.

Model	(-)AE(-)DLI	(-)AE(+)DLI	(+)AE(-)DLI	(+)AE(+)DLI
$\text{XLM}_{\text{M-M}}$	昨天下午去别地震 *Unable to translate*	每天下午去零售 Go to the retail every afternoon.	昨天下午去布袋市 Went to the fabric bag market yesterday afternoon.	昨天下午去布市 Went to the fabric market yesterday afternoon.
$\text{XLM}_{\text{MT-M}}$	昨天下午去布帳 Went to the fabric tent yesterday afternoon.	昨天下午去布市 Went to the fabric market yesterday afternoon.	昨天下午去布市府 Went to the fabric city hall yesterday afternoon.	昨天下午去布市 Went to the fabric market yesterday afternoon.
$\text{XLM}_{\text{MT-C}}$	昨天下午去布市 Went to the fabric market yesterday afternoon.	昨天下午去布市 Went to the fabric market yesterday afternoon.	昨天下午去布市 Went to the fabric market yesterday afternoon.	昨天下午去布市 Went to the fabric market yesterday afternoon.
$\text{XLM}_{\text{MT-CT}}$	昨天下午去布市 Went to the fabric market yesterday afternoon.	昨天下午去布市 Went to the fabric market yesterday afternoon.	昨天下午去布_@市 Went to the 布_@ market yesterday afternoon.	昨天下午去布市 Went to the fabric market yesterday afternoon.

Table 14: Impact of DLI mechanism in code-mixing translation. In English translations, we keep the original texts that cannot be translated.

"擺放 (collocate)", *"放置 (place)"*, and *"放 (put)"* are synonyms and all of them can be translated to *place* under most circumstances. Likewise, "剛才", "才剛", "才", "剛剛", "剛" are all translated to *just*. They are used to represent the concept that *something just happened*. In Hokkien, some expressions for *just* are "頭_@ 拄_@" and "拄_@ 才_@". "頭_@ 拄_@ 才_@", the first three characters in the sample Hokkien sentence, is not commonly used. We speculate that it is used here to emphasize *a certain event happened a very short period of time ago*. The character "頭_@" has the meaning of being *the first* in a sequence, and most of the time is used to represent *head*. In Mandarin, *head* is also written as "頭", but to the best of our knowledge, "頭" almost never means *just* in Mandarin.

After knowing the background knowledge provided in the previous paragraphs, it will be easier to understand Table 15. First, it is obvious that $\text{XLM}_{\text{M-M}}$ without transfer learning cannot capture the meaning of "頭_@ 拄_@" as *just*. Instead, the model translates the phrase into *head*, neglecting the fact that "拄_@" has contextual meaning in Hokkien, and translates the meaning of "才_@" directly, resulting in chaotic and illogical Mandarin translation. After transfer learning, $\text{XLM}_{\text{MT-M}}$ model is able to capture the meaning of "頭_@ 拄_@", and understand that "頭_@ 拄_@ 才_@" is used for emphasis, which makes "剛剛才" (or

Taiwanese Hokkien Source Sentence: 頭_@ 拄_@ 才_@ 咧_@ 囥_@ 的_@ 時_@
Mandarin Target: 剛在擺放的時候
English: When it was just collocated.

Model	Model-Result			
	(-)AE(-)DLI	(-)AE(+)DLI	(+)AE(-)DLI	(+)AE(+)DLI
$\mathrm{XLM}_{\mathrm{M-M}}$	頭才剛在放的時候 When the head was just put.	頭才剛在放的時候 When the head was just put.	頭才剛放下來的時候 When the head was just put **down**.	頭剛才放進去的時候 When the head was just put **in**.
$\mathrm{XLM}_{\mathrm{MT-M}}$	頭才剛放的時候 When the head was just put.	頭才剛進放的時候 When the head was just 進放.	剛剛才放置的時候 When it was just just placed.	剛剛才放著的時候 When it was just just left.
$\mathrm{XLM}_{\mathrm{MT-C}}$	剛剛放置的時候 When it was just placed.	剛才放置的時候 When it was just placed.	剛才才放的時候 When it was just just put.	剛才才放著的時候 When it was just just left.
$\mathrm{XLM}_{\mathrm{MT-CT}}$	剛剛才放的時候 When it was just just put.	剛剛在放的時候 When it was just put.	剛才在放的時候 When it was just put.	剛才停留的時候 When it was just stopped.

Table 15: Impact of Taiwanese Hokkien translation example in different model configuration. In English translations, we keep the original texts that cannot be translated.

Code-mixing Source Sentence: 並_@ 無_@ 羅_@ 漢_@ 跤_@ 仔_@ 佮_@ 目_@ 蓮_@ 救_@ 母_@ 的_@ 故事
Mandarin Target: 並沒有單身漢跟目蓮救母的故事
English: There is no story of a single man and Mulian Rescues His Mother[1]

Model	Model-Result			
	(-)AE(-)DLI	(-)AE(+)DLI	(+)AE(-)DLI	(+)AE(+)DLI
$\mathrm{XLM}_{\mathrm{M-M}}$	並沒有羅漢和眼蓮救母的事情也沒有 There is no thing of a 羅漢 and Yanlian Rescues His Mother also no.	並沒有羅漢足和目蓮救母的事情 There is no thing of a 羅漢 foot and Mulian Rescues His Mother.	並沒有羅漢腳踏車和蓮救_@母的故事 There is no story of a 羅漢 bicycle and Lian 救_@ Mother.	並沒有單身漢和眼蓮救母的故事 There is no story of a single man and Yanlian Rescues His Mother.
$\mathrm{XLM}_{\mathrm{MT-M}}$	並沒有羅漢腳和目蓮救母的途過 There is no 途過 of a bachelor and Mulian Rescues His Mother.	並沒有單身漢和眼蓮救母的故事 There is no story of a single man and Yanlian Rescues His Mother.	並沒有羅漢腳仔跟目蓮救母的故事 There is no story of a bachelor and Mulian Rescues His Mother.	並沒有單身漢和眼蓮救母的故事 There is no story of a single man and Yanlian Rescues His Mother.
$\mathrm{XLM}_{\mathrm{MT-C}}$	並沒有單身漢和眼蓮救母的故事 There is no story of a single man and Yanlian Rescues His Mother.	並沒有單身漢和目蓮救母的故事 There is no story of a single man and Mulian Rescues His Mother.	並沒有羅_@ 漢_@ 跤_@ 仔和眼蓮救母的故事 There is no story of a 羅_@ 漢_@ 跤_@ and Yanlian Rescues His Mother.	並沒有單身漢及眼蓮救母的故事 There is no story of a single man and Yanlian Rescues His Mother.
$\mathrm{XLM}_{\mathrm{MT-CT}}$	並沒有單身漢和眼蓮救母的故事 There is no story of a single man and Yanlian Rescues His Mother.	並沒有單身漢和眼蓮救母的故事 There is no story of a single man and Yanlian Rescues His Mother.	並沒有羅漢跤_@仔跟目_@ 蓮救母的故事 There is no story of a 羅漢跤_@ 仔 and 目_@ Lian Rescues His Mother.	並沒有漢腳仔和目蓮救母的故事 There is no story of a 漢腳仔 and Mulian Rescues His Mother.

[1]A popular Chinese Buddhist tale which first attested in a Dunhuang manuscript dating to the early 9th century

Table 16: Impact of transfer learning code-mixing translation example in different model configuration. In English translations, we keep the original texts that cannot be translated.

"剛才才") in the translation. However, considering the contextual meaning in the sentence. We would not use "剛剛才" (or "剛才才") and "的時候" at the same time. It will cause the sentence contains two "just"s in the expression. $\mathrm{XLM}_{\mathrm{MT-C}}$ model no longer translates "頭_@" directly. $\mathrm{XLM}_{\mathrm{MT-C}}$ model and $\mathrm{XLM}_{\mathrm{MT-CT}}$ model are able to capture that "頭_@ 拄_@ 才_@" is one expression of *just* and does not need to be translated twice, making the sentence less redundant.

Lastly, let's look at an interesting code-mixing case shown in Table 16. We want to provide some prior knowledge before starting. First, "羅_@ 漢_@ 跤_@ 仔_@" is a phrase in Hokkien which has the negative meaning of *homeless male*, or *rogue male*. In the past, for many reasons, there was a lack of other appropriate words to express such a concept. The phrase is translated to and documented as "羅漢腳" in historical records, leveraging the Mandarin characters corresponding to the pronunciation. In recent years, "羅_@ 漢_@ 跤_@ 仔_@" gradually takes on the meaning of *single man*, and *single man* is written as "單身漢" in Mandarin. Second, "目" and "眼" can both be translated to *eyes* regarding their primary meaning, but "目" is more versatile. Taking the case in Table 16 as an example, since "目蓮 *(Mulian)*" is a name entity, we expect "目蓮 *(Mulian)*" to be translated to Mandarin in the process of generating code-mixing sentence, and the original form should be preserved in the translation. However, since we didn't label "目蓮 *(Mulian)*" as Mandarin during the stage of generating code-mixing sentence, it is acceptable that the model translated "目_@" to "眼 *(eyes)*".

As the previous case, $\mathrm{XLM}_{\mathrm{M-M}}$ model without transfer learning cannot perform translation well when it comes to code-mixing. Configuration (-)AE(-)DLI doesn't converge well and configuration (+)AE(-)DLI contains the phrase "腳踏車 *(bicycle)*", which is totally unrelated to the original sentence. We speculate that the model predicts "腳" in the prediction stage, and the language model gives the phrase "腳踏車 *(bicycle)*" as a result since these two phrases might have contextual relationships in Mandarin. The configuration of that model using the DLI mechanism provides a translation that has closer meaning to the original sentence. In configuration (-)AE(+)DLI, "羅漢腳" was translated to "羅漢足". The character "腳" means *foot* and "足" is its synonym. Although the translation is not smooth, the sentence still has a similar meaning.

The remaining models that utilize transfer learning techniques all translate "目_@ 蓮_@" to either "眼蓮" or "目蓮". Most of the models translated "羅_@ 漢_@ 跤_@ 仔_@" accurately to "單身漢" or "羅漢腳". In some of the translations, the models translate "仔", a character that often acts as an auxiliary word in Hokkien. In Mandarin, "仔" is rarely used as an auxiliary word. Even though the positions of "仔" in the translated sentences are a bit weird, it doesn't affect the meaning of the sentences.

We speculate the reason for translating "仔" is that the translation models are affected by the Hokkien language model more. We believe smoother translations can be obtained by making the models learn more about the sentence structures of Hokkien and Mandarin.