Translating Code-Switched Texts From Bilingual Speakers

Stanford CS224N Custom Project

Kalyani Ramadurgam
Department of Computer Science
Stanford University
`kalyrama@stanford.edu`

Surabhi Mundada
Department of Computer Science
Stanford University
`surabhim@stanford.edu`

Abstract

Code-switched language is commonly found in interactions between bilingual individuals, yet has not been optimized in NMT. The goal of this project is to create a model best suited for code-switching translation tasks. Specifically, given an input of **bilingual**, or **code-switched** text, we want to create a model that outputs a translation in one desired language. We experiment with two approaches. First, a LID-Translation Model Pipeline approach, a two model pipeline that 1) uses a language identification model to figure out what words need to be translated in a bilingual text and 2) translates these identified words via a language translation model. This approach includes a language translation model we have fine-tuned and was motivated by the fact that our code-switched dataset did not have ground truth translations. Second, during the course of our project, a new code-switched dataset was released, and thus we also trained a Direct-Translation Bilingual Model. This bilingual model was also tested with and without the LID-Translation Pipeline. Our results show two main findings. First, the LID-Translation Model Pipeline performs better than a direct translation pipeline. Second, the Direct-Translation Bilingual Model performed better than the regular translation models fine-tuned on non-bilingual datasets.

1 Key Information

External collaborators (if you have any): N/A
External mentor (if you have any): N/A
Sharing project: Related project in CS 329s, but the project for that class uses a basic model and focuses on building out the end-to-end system and UI around the model. For this class's project, we are focusing on the specific translation task, pipeline, and model architecture.

2 Introduction

Code switching is the act of switching between two or more languages in a conversation. In many immigrant communities, different generations speak different languages, and often prefer to communicate in text with words from both languages to bridge any communication gaps. These communities often use code-switching to communicate, especially through social media platforms and conversational text [1].

Currently, code-switching is not typically handled in language translation models since code-switched translation data is harder to acquire. However, code-switched language translation is becoming an increasingly explored field, as seen by Ahan M.R.'s *End-to-End Code Switching Language Models for Automatic Speech Recognition* [2]. A new benchmark for code-switching tasks, such as the Linguistic Code-switching Evaluation (LinCE), was also recently created in 2020 [3] to explore this relatively unexplored space. Additionally, there is not a lot of research done on potentially suitable models besides the standard BERT and NMT translation models.

The goal of this project is to create a model best suited for code-switching translation tasks. Specifically, given an input of a piece of text with two languages mixed (*Note:* we will call this **code-switched** or **bilingual** text in this paper), we want to output a translation of the text in one desired language. Essentially, the sentence "Hola mi amor. I need to go to la tienda" should be fed into a model and translated to "Hi my love. I need to go to the store." We also focus on translating the code-switched text in one direction only (i.e. Spanish+English to English).

3 Related Work

A motivation for this project is the lack of code-switched translation research in recent literature. In general, recent efforts related to code-switching are more about data collection and analysis [4] [5], and few works have explored downstream tasks as Language Modeling (LM) [6] and Automatic Speech Recognition (ASR) [7].

The main paper we referenced to build our implementation was Ahan M.R.'s *End-to-End Code Switching Language Models for Automatic Speech Recognition*. This paper specifically synthesizes a Hindi-English dataset with GANs, and does translation on this dataset with BERT, roBERTa, LSTMs, and Seq2Seq models. We use this paper as a baseline to compare our accuracies against and to understand what approaches would be helpful for this task. However, all our code and overall model pipeline is built from scratch as well as utilizes the HuggingFace API's pretrained models. The dataset we use is also completely different than this paper: it is a Spanish-English codeswitched dataset and it does not contain ground truth translations.

Some recent approaches have also suggested using code-switching data as a form of pre-training to increase overall efficacy of monolingual translation systems, since code-switched data can force a model to learn on alignments of text in the source and target languages [8]. Using this logic, we were interested in exploring whether such alignments could give way to accurate translation of already code-switched data. Similar papers in this domain have found that one of the challenges of the NLP concerning code-switched texts is the lack of data. As such, some groups have undertaken the process of creating parallel corpuses in trios: one code-switched, and two in each target language [9]. The first of such datasets was only created in 2019 on an Arabic-English code-switched dataset, so it is clear that this is a modern problem. Additionally, this paper translated the whole code-switched sentence directly with the different systems, and concluded that it would be interesting to identify implicitly the language system of the foreign segments and carry out the translation with an appropriate translation system [9]. We decided to include this concept in our experimentation to test out the effect of employing Language Identification models to tag the language of words in the phrase and only translate the tagged text segments.

4 Approach

We took two main approaches for this project, described below:

- LID-Translation Model Pipeline (Language Identification to Translation Model Pipeline): This is a two model pipeline that first identifies words and phrases that need to be translated and then uses a direct translation model for translating these identified phrases. Our initial model pipeline focused on training a direct translation model that could handle code-switched input and output a translation in one language. When trying to find data for such a model, however, it became clear that very few languages provided bilingual to single language data sets. Instead, we found code-switched data where each word was tagged by language. Thus, we created this LID-Translation pipeline to achieve our goal.
- Direct-Translation Bilingual Model: This is a direct translation model pipeline trained on a bilingual dataset. During the course of this project, a new bilingual dataset with ground truth translations became available on the LinCE benchmark [3], and so we added this approach of building a direct translation model meant specifically for bilingual translations tasks.

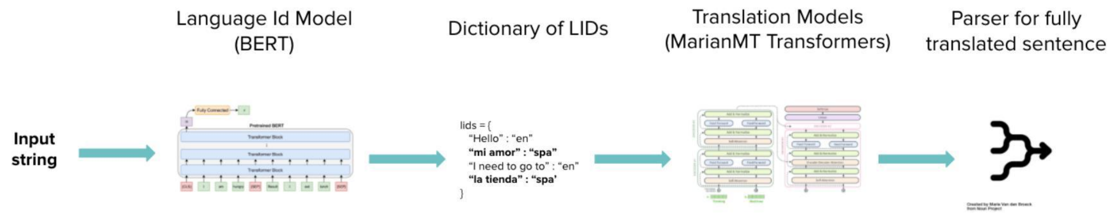


Figure 1: LID-Translation Model Pipeline Diagram

4.1 Baseline Models

We tested the performance of standard translation models on our code-switched dataset for our baseline. Although we chose pre-trained translation models that could handle Spanish to English translations, none of them had been trained with code-switched Spanish-English to English translations. We also didn't use the LID-Translation Model Pipeline for the baseline so that we could clearly understand how standard translation models handle code-switched input. Using a direct translation model also allowed for us to directly compare to baselines described in our referenced paper *End-to-End Code Switching Language Models for Automatic Speech Recognition* [2].

We used the MarianMT, MBART, and T5 models, which are all pre-trained translation models available on the HuggingFace API. The MarianMT is a transformer encoder-decoder model pretrained on the Open Parallel Corpus [10]. We used this because we believed a multilingual model might be able to handle bilingual text well as it is trained on the syntactical structure of many languages. Then, MBART is a seq2seq denoising auto-encoder model pretrained on monolingual corpora from many languages using the BART objective. BART is a seq2seq model that uses a bidirectional encoder (similar to BERT) and left-to-right decoder (similar to GPT) [11]. BART also matches the performance of RoBERTa, and so one advantage is that we can compare our BART's performance to the referenced paper's RoBERTa performance. Finally, T5 (i.e. "Text to Text Transfer Transformer") is an encoder-decoder model pre-trained on both unsupervised and supervised tasks [12]. We used T5 because it is also pretrained on unsupervised tasks, which means it might have more capability up on unseen data such as code-switched data.

4.2 LID-Translation Model Pipeline

As explained in the previous section, the initial goal was to train a language translation model that could handle bilingual text and translate to a single language. As large datasets did not exist for this task, we implemented the LID-Translation Model Pipeline: a pipeline with two machine learning models to 1) identify words and phrases that needed to be translated and 2) a direct translation model for translating the identified phrases. This pipeline is shown in (Figure 1).

First, the input string is sent through a BERT model to capture which words need to be translated. In this case, the input is Spanish + English and words that need to be translated are the words identified as Spanish. From there, the output of the BERT model is parsed and the words that need to be translated into the target language are sent through a MarianMT language translation model. The translated words are pieced back together with the words that don't need translations and outputted as the final translation.

The language identification BERT model used was a token classification model from the Hugging Face API, and more details of this architecture can be seen in (Figure 2) [13]. BERT was chosen because it is trained on a large corpus of multilingual data and has been found to perform well on token identification tasks [14]. This specific model was pre-trained on the Linguistic Code-Switching Evaluation Benchmark (LinCE) dataset, a corpus of 50k tweets with mixed Spanish and English text [3]. SentencePiece tokenizer was used to tokenize the code-switched text [15].

Then, the language translation model was fine-tuned on top of HuggingFace's pre-trained MarianMT model. MarianMT was chosen for translations because of its high performance in such tasks and pretraining to handle multiple languages, as well as its better performance in our baseline experiments

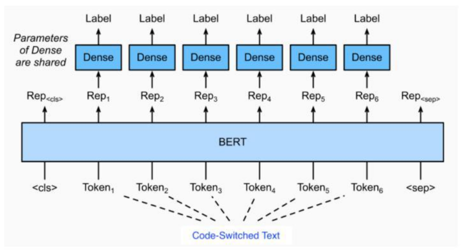


Figure 2: Token classification diagram. Image credit: [16]

[17]. Then, the model was fine-tuned on a web corpus of 21k Spanish sentences with parallel English translations [18]. As a note, experimentation and training of this translation model was also done for the related project in CS329s. This project focuses more on the effect of the LID-Translation pipeline in NMT, rather than the specific model.

4.3 Direct-Translation Bilingual Model

As described previously, the LinCE benchmark released a bilingual dataset with ground truth translations during the course of our work. So, our next approach was to train an new model with this bilingual dataset, in hopes to create a model specially curated for codeswitching tasks.

In order to develop this model, we trained on top of the pre-trained MarianMT model with our bilingual dataset. Using a pretrained model, especially one that has been trained on the same supported languages as our dataset, meant that the model's layers already retained some important structures needed for the translation task. This is especially important since the bilingual dataset we were training with was small. Then, we pre-processed our data by restructuring it into a Pytorch Dataset class so it can be easily passed into MarianMT. We also tokenized and padded the data using the MarianMT's corresponding SentencePiece-based tokenizer [15]. For training, the pre-processed input is passed into the MarianMT model, which follows a similar architecture to a standard Transformer model as seen in (Figure 3) [19] [20]. The input gets passed into the encoder, creating a hidden state, which gets passed into the decoder along with pre-processed and tokenized target translations. Both the encoder and decoder have 6 layers. We chose MarianMT based on our baseline results, and to keep the evaluations between this Direct-Translation bilingual model and our other Spanish to English model consistent. We also used the default causal language modeling loss in MarianMT [21] along with the AdamW optimizer, which is like Adam except fixes a weight decay problem seen in Adam optimizers by decoupling weight decay and loss-based gradient updates [22].

We also tested this model with and without the LID-Translation Model Pipeline approach, utilizing a Hugging Face language identification BERT model to extract the language tags. Another caveat is that although this bilingual dataset was from the same benchmark as the previous data, the supported language was instead Hindi + English to English; all the other models we used had been based on Spanish + English translation data. So, we also utilized Hindi to English language identification and translation models as another baseline that we could better compare our Direct-Translation Bilingual model's results to [23] [24].

5 Experiments and Analysis

5.1 Data

We used the following datasets for our different tasks:

- Baseline tasks: We tested model performance using the LinCE dataset [3]. This is a corpus of 51k code-switched English and Spanish tweets matched to language identification tags.
- LID-Translation Model Pipeline: The language identification model was already pre-trained on the LinCE dataset described above. Then, to fine-tune the language translation model, we used a corpus of 21k Spanish sentences with parallel English translations created from multilingual websites [18]. Finally, we tested model performance using the same Spansih English LinCE dataset described above.

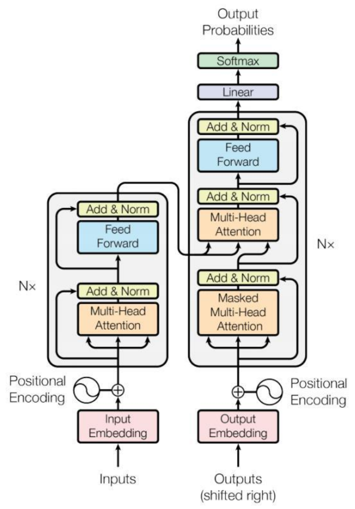


Figure 1: The Transformer - model architecture.

Figure 3: Transformer Architecture. Image credit: [20]

- Direct-Translation Bilingual Model: We used a LinCE dataset [3]. However, this data instead was a corpus of 9k code-switched Hindi and English phrases matched to their ground truth English translation.

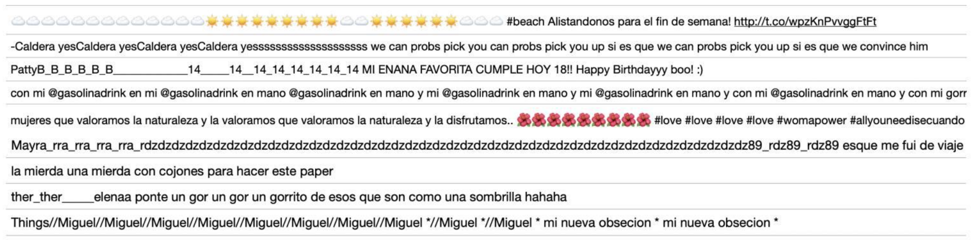
We chose tweet data, website data, and more casual phrases since code-switching is usually seen within more casual language contexts such as social media and messaging.

5.2 Evaluation Metrics

Our evaluation metrics for our baseline were Word Error Rate (WER) and sentiment classification. Since we did not have a ground truth bilingual translation in our LinCE Spanish-English data, we passed the bilingual text into a Spanish to English translation model and analyzed the WER for that translation, since only a few words in each datapoint of the LinCE data were in Spanish. This seemed incomplete, however, because there are multiple ways to translate a phrase, and they contain different connotations. To account for this, we also conducted bilingual sentiment analysis comparing the baseline translations and the original bilingual data, and we used this to test for connotation similarity.

For the LID-Translation Model Pipeline, we evaluated both the performance of the translation model that we had fine-tuned via accuracy on the test set and the performance of translations via sentiment classification. We chose not to compute WER for this model based on the caveats we observed from the baseline evaluations; WER didn't seem practical for evaluating a translation task with no ground truth translations. We also did this evaluation of translations without the language identification component of the pipeline (i.e. a direct Spanish to English translation model), to understand whether this pipeline was effective or not.

Then, for the Direct-Translation Bilingual Model, since this LinCE Hindi-English data contained ground truths, we were able to evaluate translations via BLEU scores and WER. We did these evaluations with and without the language identification component of the LID pipeline, to further understand how the pipeline works with a bilingual model. We also evaluated BLEU scores and WER on the non-finetuned, regular Hindi to English model since this was a better comparison as it was in Hindi and not Spanish; this was also done with an without the LID pipeline. Finally, although we wanted to do sentiment classification on this model, we could not due to limits in good Hindi-English language identification models we could find. In addition to individual-level BLEU scores and WER, we also wanted to observe the BLEU score and WER as we increased the amount of the original and translated Hindi corpuses. This was useful because we were able to visually confirm that both metrics had converged nicely to a value.



☁️☁️☁️☁️☁️☁️☁️☁️☁️☁️☁️☁️☁️🌞🌞🌞🌞🌞🌞🌞🌞🌞☁️☁️🌞🌞🌞🌞🌞🌞☁️☁️☁️ #beach Alistandonos para el fin de semana! http://t.co/wpzKnPvvggFtFt

-Caldera yesCaldera yesCaldera yesCaldera yessssssssssssssssssss we can probs pick you can probs pick you up si es que we can probs pick you up si es que we convince him

PattyB_B_B_B_B_B____________14_____14__14_14_14_14_14_14 MI ENANA FAVORITA CUMPLE HOY 18!! Happy Birthdayyy boo! :)

con mi @gasolinadrink en mi @gasolinadrink en mano @gasolinadrink en mano y mi @gasolinadrink en mano y mi @gasolinadrink en mano y con mi @gasolinadrink en mano y con mi gorr

mujeres que valoramos la naturaleza y la valoramos que valoramos la naturaleza y la disfrutamos.. 🌺🌺🌺🌺🌺🌺🌺🌺🌺 #love #love #love #love #womapower #allyouneedisecuando

Mayra_rra_rra_rra_rra_rdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdzdz89_rdz89_rdz89 esque me fui de viaje

la mierda una mierda con cojones para hacer este paper

ther_ther_____elenaa ponte un gor un gor un gorrito de esos que son como una sombrilla hahaha

Things//Miguel//Miguel//Miguel//Miguel//Miguel//Miguel//Miguel//Miguel *//Miguel *//Miguel * mi nueva obsecion * mi nueva obsecion *

Figure 4: Baseline MBART example translations

Baseline model	Sentiment label accuracy	Average WER
MarianMT	0.8	1.1704
MBART	0.9	0.9129
T5	0.8	1.0413

Figure 5: Comparing baseline models for the Spanish experiment

5.3 Experimental Details

We ran all of our models on the GPU. For the baseline models, inference took about 15 minutes / 1000 samples for both the MarianMT and T5, and around 15 minutes / 20 samples for the BART with the default HuggingFace model configurations. For the LID-Translation Model Pipeline translation model and the Direct-Translation Bilingual Model, we trained for 2 epochs and 3 epochs respectively. We used 6 encoder and decoder layers, as well as the default MarianMT HuggingFace configurations, which include 16 encoder and decoder attention heads, weight decay of 0.1 for the AdamW optimizer, dropout of 0.1, and learning rate of $5e-5$.

5.4 Baseline Results and Analysis

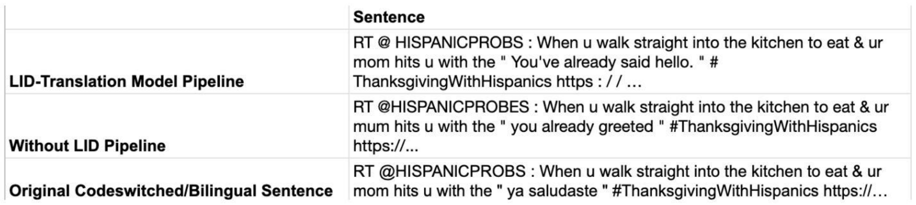
After running the baseline models of MarianMT, MBART, and T5, we found through a manual parsing of data shows that the MarianMT performed the best, since it maintained a lot of the syntactical structure of sentences with fewer Spanish words. On the other hand, MBART and T5 had repeating sequences and seemed to ignore initial word structure (Figure 4). It is important to note, however, that due to time constraints we were only able to run MBART on a very small set of samples, while Marian and T5 were able to run on many more. This could explain the better MBART numbers as seen in Figure 5 even though a manual parsing shows a low-quality translation from MBART.

One interesting finding is also that MBART produced the lowest WER, but this is likely because the repeated sequences were often not separated by spaces, so the outputted translations were often a similar length to the reference text. On the other hand, MarianMT often produced more coherent sentences, but without much training added on words that made the text much longer than the reference text, which increased WER.

Based on our manual evaluations of these findings, we decided that MarianMT maintained the most syntactic structure of words, and this would be the model we would used to evaluate the efficacy of the LID-Translation pipeline and Direct-Translation Bilingual models on both Spanish and Hindi dataset.

5.5 Main Results and Analysis

We can see from Figures 7-10 that the LID-Translation Model Pipeline performs better than direct-translation model pipeline without LIDs for the Spanish + English dataset. When comparing the BLEU scores for the Hindi/English to English translation task, we see that the non-LID non-finetuned model converged to around 0.2, while the LID non-finetuned model converged to a BLEU score of 0.4 (Figure 7). The respective WERs of 1.4 and 1.2 respectively underscore the superior performance of the LID model over the non-LID model (Figure 8). Additionally, the LID pipeline model generated



	Sentence
LID-Translation Model Pipeline	RT @ HISPANICPROBS : When u walk straight into the kitchen to eat & ur mom hits u with the " You've already said hello. " # ThanksgivingWithHispanics https : / / …
Without LID Pipeline	RT @HISPANICPROBES : When u walk straight into the kitchen to eat & ur mum hits u with the " you already greeted " #ThanksgivingWithHispanics https://...
Original Codeswitched/Bilingual Sentence	RT @HISPANICPROBS : When u walk straight into the kitchen to eat & ur mom hits u with the " ya saludaste " #ThanksgivingWithHispanics https://…

Figure 6: Spanish translations

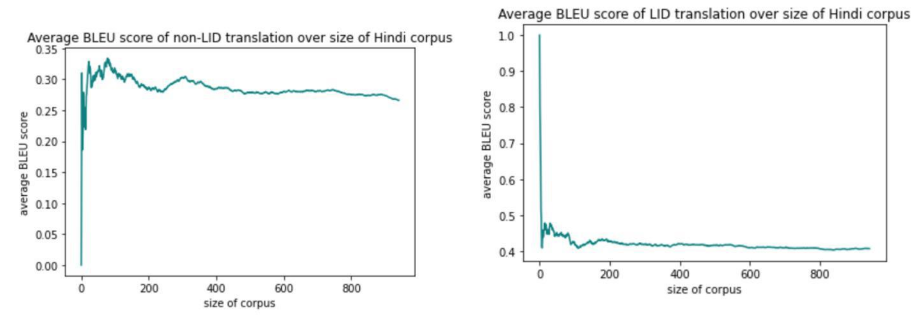


Figure 7: BLEU Scores for Hindi Experiments on Test Set

a higher sentiment label accuracy with the LID-Translation at 91.2%, while without LID was a bit lower at 87.5%., showing that the LID pipeline can maintain connotation of words that may otherwise lose their structural meaning in a general translation model (Figure 9). Furthermore, we can qualitatively analyze the quality of translations, such as the ones seen in (Figure 6). In this translation, the LID-Translation Model Pipeline retains a more natural English translation: "You've already said hello," Meanwhile, the pipeline without LIDs translated into "You've already greeted," a phrase people usually won't say. Altogether, we can see that the LID-Translation Model Pipeline performs better for bilingual translation tasks. This might be because a fully code-switched sentence can't be represented well with a single Transformer since two languages and different conjugations and placements of words create a more complex sentence structure; language IDs on the other hand may help retain important structural information. Additionally, the language-translation model fine-tuned for this task had 73% accuracy on the test set. Additional training is likely to improve this accuracy, thus also improving the sentiment label accuracy of the LID-Translation Model Pipeline. We also note that sentiment classification accuracy is pretty high (80-90%) probably because the bilingual sentence contains words from English as well, thus we should compare them relative to each other.

Then, we found that the Direct-Translation Bilingual Model performed better than the regular Hindi to English translation models fine-tuned on non-bilingual datasets with or without LID. More specifically, the direct translation models with and without LID generate BLEU scores of 0.51 and 0.46, while the respective non-finetuned Hindi translation models generate BLEU scores of 0.41 and 0.23 (Figure 10). The same trends apply to WER, where we see that the direct translation bilingual models perform better than the non-finetuned model in all cases and all metrics. This makes sense given that the model was fine-tuned on a bilingual dataset and thus was meant for a bilingual translation task, as well as perhaps takes into account common structural elements that are unique to code-switched data. We also think this BLEU score for all the models in Figure 10 might be relatively lower due to nuances in Hindi. Specifically, one example we saw is "hello" translates to "listen" in a direct Hindi to English translation 'slang' context. However, in a code-switched context, we want to retain the english "hello." Using LIDs probably catches this. However, this also shows some more complex translation nuances between words from English to Hindi and vice versa.

6 Conclusion

The first main finding our work shows is that the LID-Translation Model Pipeline performs better than a direct translation pipeline without LIDs (i.e. one without the 1st step in LID-Translation Pipeline). Thus, LIDs are valuable to bilingual translation tasks. This could potentially mean that

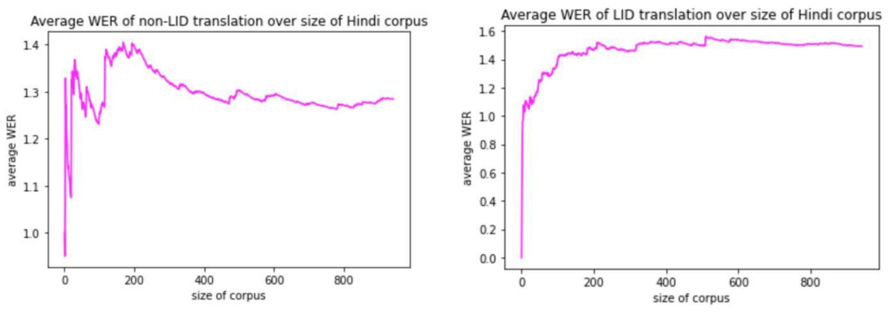


Figure 8: WER for Hindi Experiments on Test Set

	Sentiment label accuracy
LID-Translation Model Pipeline	0.912
Without LID Pipeline	0.8751

Figure 9: Comparing sentiment label accuracy of LID and non-LID pipelines

other language and word tagging (i.e. NER) is valuable to capture the syntactical structure of two languages, which is an area we could explore in the future.

Although our primary limitation was finding code-switched dataset with ground truth translations, we were able to experiment with a small code-switched dataset in Hindi/English that was released during the course of our work, thus creating the Direct-Translation Bilingual model. Our work in this area shows that the Direct-Translation Bilingual Model (fine-tuned on bilingual data) performed better than the regular translation models (fine-tuned on non-bilingual data). This demonstrates that the development of bilingual-specific translation models is a valuable NLP task, and urges development of these types of translation models, especially to help increase communication in immigrant communities.

References

[1] N. Jose, B. R. Chakravarthi, S. Suryawanshi, E. Sherly, and J. P. McCrae. A survey of current datasets for code-switching research. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 136–141, 2020.

[2] Ahan M. R. and Shreyas Sunil Kulkarni. End-to-end code switching language models for automatic speech recognition. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, 2020.

	BLEU Score (across test set)	Average WER
Direct Translation Bilingual Model	0.5186	1.1152
Direct Translation Bilingual Model WITHOUT LID	0.4612	1.2645
Non-Finetuned Hindi → English Translation Model	0.4071	1.2846
Non-Finetuned Hindi → English Translation Model WITHOUT LID	0.2265	1.4936

Figure 10: BLEU Scores and WER for Hindi experiments on test set

[3] Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France, May 2020. European Language Resources Association.

[4] Rijhwani S. Ros ́e C. Levin L. Yoder, M. Code-switching as a social act: The case of arabic wikipedia talk pages. In *Proceedings of the Second Workshop on NLP and Computational Social Science*, pages 73–82, 2017.

[5] Das A. Choukri K. Declerck T. Goggi S. Grobelnik M. Maegaard B. Mariani J. Mazo H. Moreno A. Odijk J. Piperidis S. Gamback, B. Comparing the level of code-switching in corpora. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 73–82, May 2016.

[6] Parekh T. Jyothi P. Garg, S. Dual language models for code mixed speech recognition. In *CoRR abs/1711.01048*, 2017.

[7] Sainath T.N. Weiss R.J. Li B. Moreno P. Weinstein E. Rao K. Toshniwal, S. Multilingual speech recognition with a single end-to-end model.n. 2017.

[8] Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. Code-switching pre-training for neural machine translation. 2020.

[9] Denis Jouvet Dominique Fohr Odile Mella et al.. Mohamed Menacer, David Langlois. Machine translation on a parallel code-switched corpus. In *Conference on Canadian Artificial Intelligence*, 2019.

[10] Jörg Tiedemann. Parallel data, tools and interfaces in opus. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*, 2012.

[11] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. 2019.

[12] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. 2020.

[13] sagorsarker. Huggingface codeswitch-spaeng-lid-lince repo.

[14] Bert base multilingual cased.

[15] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, 2018.

[16] Dive into Deep Learning. Fine-tuning bert for sequence-level and token-level applications.

[17] Huggingface marianmt documentation.

[18] EU Open Data Portal. Spanish-english website parallel corpus.

[19] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in c++. 2018.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[21] Huggingface examples.

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[23] sagorsarker. Huggingface codeswitch-hineng-lid-lince repo.

[24] Helinski-NLP. opus-mt-en-hi repo.