Towards Robust Cross-Lingual Transfer for Natural Language Inference over Code-Switched Hinglish

Kushagra Mahajan, Nikhil Gupta, Shubham Phal
School of Computer Science
Carnegie Mellon University
kmahajan@andrew.cmu.edu, nikhilgu@andrew.cmu.edu, sphal@andrew.cmu.edu

Abstract

The goal of Natural Language Inference (NLI) is to explore the logical relationship between pairs of sentences.NLI is a challenge for low-resource and code-mixed languages due to the lack of sufficient annotated corpora in these domains. In this work we ask the question of whether cross lingual transfer can facilitate NLI performance over a low resource code-switched corpus. We hypothesize that the NLI performance over low resource code-switched languages can be improved by translating it into its either its high resource matrix or embedded language, or by adapting the language models to the code-switched domain. For evaluating our hypothesis we perform our experiments on a low resource code-mixed Hinglish dataset, GLUECoS proposed by (Khanuja et al., 2020). We experiment with multiple techniques to improve the cross-lingual transfer and our observations uncover interesting crosslingual transfer phenomemon that take place in language models while achieving state-of-the-art performance on the GLUECoS NLI benchmark that beats both the highest accuracy on the GLUECoS leaderboard (57.74%) as well as that reported most recently by Chakravarthy et al. (2020) (63.69%). We summarize our findings and present directions for future work. Furthermore we opensource our code and model.

1 Introduction

Code-switching is a phenomenon in which polyglots actively switch between two languages in conversations and other forms of informal communication such as text messages, tweets etc. Very often the code-switched corpus is available in a single script, which may be different from the native script of the matrix language. In this work, we explore various approaches to perform Natural Language Inference (NLI) on code-switched Hindi-English (Hinglish).

NLI is an important task under the General Language Understanding and Evaluation(GLUE) Wang et al. (2018) benchmark. The task involves determining whether a given premise entails or contradicts a given hypothesis. The domain is relatively well researched for high resource languages with the availability of several large monolingual datasets such as SNLI Bowman et al. (2015) and Multi-NLI (MNLI) Williams et al. (2018), as well crosslingual datasets such as XNLI Conneau et al. (2018). However very few datasets are available for low resource languages and even fewer datasets are available for code-switched languages.

The GLUECoS dataset introduced by Khanuja et al. (2020) is to the best of our knowledge the only available NLI dataset for code-switched Hindi and English (Hinglish). Furthermore the GLUECoS dataset contains only about 2240 Hinglish sentence pairs for NLI. This low resource aspect of the data makes our work particularly challenging as we can only rely on cross-lingual transfer learning techniques to improve the NLI performance on this dataset.

In an attempt to improve performance, we propose translating the code switched data to the either the matrix or embedded high resource language for which there is an abundance of NLI data. We investigate the effects of these transformations by performing a three-way comparative analysis between Hinglish (original) and the translated Hindi (matrix language) and English(embedded language) parallel corpora. Although there exist a plethora of traditional techniques for NLI such as symbolic logic, neural networks and knowledge bases the introduction of transformer models that are pretrained on massive amounts of monolingual and crosslingual corpora have advanced the state-of-the-art in NLI. In our experiments we use two multilingual transformer models namely XLM-Roberta (XLM-R) Conneau et al. (2018) and Multilingual BERT (mBERT) Devlin et al. (2019a)

To summarize our work we look at few shot approaches for NLI over Hinglish, evaluate differ-

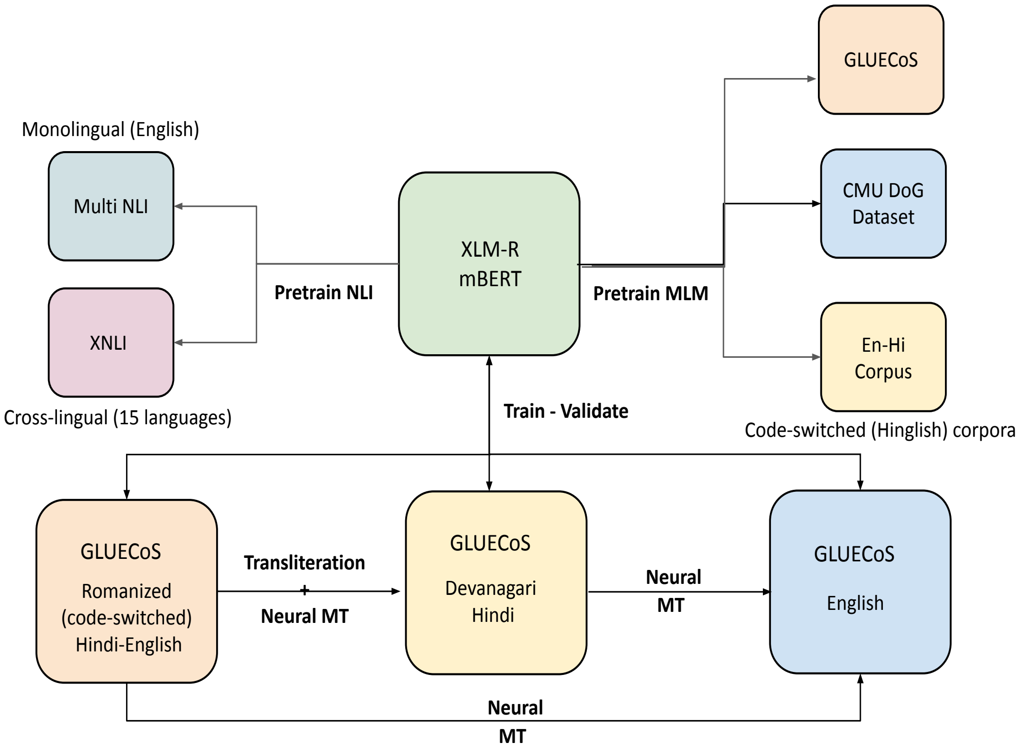


Figure 1: Flowchart illustrating the components of our proposed solution.

ent pretraining strategies, explore state-of-the-art language understanding architectures for the NLI task, perform domain adaptation of language models over Hinglish and propose a direct Hinglish to English translation approach for better NLI performance. Our complete framework is illustrated in figure 1.

The remaining paper is organized as follows: In Section 2, we contrast our work to the existing literature. Section 3 details our methodology and motivates our reasoning behind adopting the proposed approach. Section 4.6 outlines our experimental setup and discusses the results obtained while section 5 provides a qualitative overview of the errors in our pipeline. We summarize our findings in section 6 and state our contributions and propose directions for future work in section 7.

2 Related Work

In recent years transformer models such as BERT Devlin et al. (2019b), Multilingual BERT Devlin et al. (2019b), XLM-RoBERTa (XLM-R) Conneau et al. (2019) have shown superior language understanding across tasks on the GLUE Wang et al. (2018) benchmark. Particularly these models are the current state-of-the-art on large NLI datasets such as SNLI, MNLI and XNLI. Therefore we use these models for our experiments.

The work of Chakravarthy et al. (2020) is most similar to ours. They investigate the effectiveness of language modeling, data augmentation, and architectural approaches to address the code-mixed, conversational, and low-resource aspects of the GLUECoS dataset. We extend this work by incorporating translation and performing a three-way comparison of the NLI performance across Hinglish, Hindi and English. Additionally we evaluate different pretraining strategies and experiment with a novel multimodal attention technique to jointly learn NLI from English and Hinglish representations. Furthermore to reduce error propagation in translation we improve our two stage translation pipeline (Hinglish->Hindi->English) by performing a single stage translation from (Hinglish->English)

(Prasad et al., 2021) propose bilingual intermediate pretraining as a reliable technique to derive large and consistent performance gains on three different NLP tasks using code-switched text namely, Hindi-English Natural Language Inference (NLI), Question Answering (QA) tasks, and Spanish-English Sentiment Analysis (SA). (Agarwal et al., 2021) propose code-mixed dialog generation where

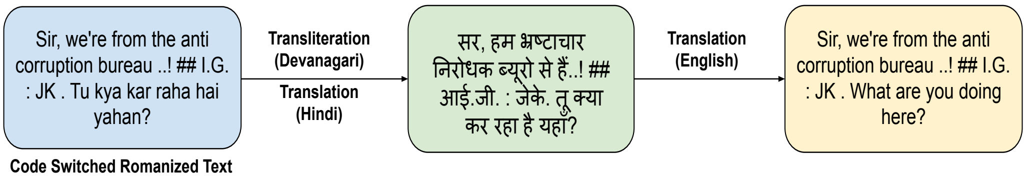

Sir, we're from the anti corruption bureau ..! ## I.G. : JK . Tu kya kar raha hai yahan?
Code Switched Romanized Text
Transliteration (Devanagari)
Translation (Hindi)
सर, हम भ्रष्टाचार निरोधक ब्यूरो से हैं..! ## आई.जी. : जेके. तू क्या कर रहा है यहाँ?
Translation (English)
Sir, we're from the anti corruption bureau ..! ## I.G. : JK . What are you doing here?

Figure 2: Pipeline showing the translation and transliteration of Romanized code-switched text to Devanagari Hindi text, and finally translation to Romanized English text.

the authors generate utterances in code-mixed languages rather than a only a single language that is more often just English. Our approach differs from these approaches in the methodologies used for cross-lingual transfer as we adapt our models on code-switched datasets and also pretrain them on monolingual and crosslingual datasets.

3 Methodology

3.1 Segmented Translation and Transliteration

The GLUECoS dataset consists of Hindi-English code switched data present in non standard romanized script. For multilingual translation models like MarianMT Junczys-Dowmunt et al. (2018), M2M100 Fan et al. (2021) etc. to operate on this data, we need to convert the Romanized Hindi to the Devanagari script. We use the work of (Singh et al., 2018) for segmented transliteration of the Romanized Hindi to Devanagari Hindi. We then translate the Devanagiri Hindi to English. Details about the segmentation process for translation and transliteration are discussed in Section 4.2.

3.2 Direct Translation: Code-switched Hinglish to English

Instead of transliterating Romanized Hindi to Hindi and then translating Hindi to English as explained previously in Section 3.1, we also experimented with direct translation from code-mixed Hinglish to English. For this we followed the method mentioned in (Agarwal et al., 2021) to fine-tune a multilingual model for the translation task. Implementation details are discussed in Section 4.3.

3.3 Pretraining with NLI Objective

The work of Chakravarthy et al. (2020) demonstrated that augmenting the GLUECoS dataset with a limited set of examples (4K) from SNLI and XNLI can help address the low resource aspect of GLUECoS and improve NLI performance. We hypothesize that pretraining multilingual models on large monolingual or multilingual datasets can help crosslingual transfer of these learnings onto new datasets. Thus we extend the prior work by pretraining XLM-R and mBERT on the MNLI and the XNLI dataset.

The Multi-NLI dataset consists of monolingual english premise-hypothesis pairs belonging to different genres while the XNLI dataset consists of premise-hypotheis pairs from Multi-NLI translated into 14 other languages. As the GLUECoS dataset consists of code-switched movie dialogues from diverse genres, these two datasets serve as interesting baselines in evaluating the extent of crosslingual transfer.

3.4 Domain Adaptation via Masked Language Modelling

We hypothesize that linguistically motivated objectives such as Masked Language Modeling (MLM) can help multilingual models such as XLM-R and mBERT better capture the linguistic knowledge in the code-mixed corpus. This enhanced understanding of the code-mixed language can facilitate the performance of downstream tasks such as NLI which require models to develop a general understanding of the language domain.

To validate our hypothesis, we pretrained XLM-R and mBERT with the MLM objective on the Hinglish corpus in GLUECoS. We further augmented this data with data from other sources so as to retain generalizability in the learned Hinglish representations and to ensure that these representations are not biased towards a particular domain or dataset. In all, we augmented the data from three code-switched datasets namely GLUECoS, CMU Hinglish DoG dataset Zhou et al. (2018), and the en-hi-codemixed-corpus (Dhar et al., 2018). Each of these datasets contain code-switched Hindi English data in Roman Script.

Methods	Metrics			
	XLM-R		mBERT	
Pretraining	Macro-F1	Acc	Macro-F1	Acc
XNLI	0.82	0.8215	0.84	0.8432
MNLI	0.90	0.8974	0.88	0.878
Zero-Shot Eval	Macro-F1	Acc	Macro-F1	Acc
XNLI pretraining	0.42	0.4237	0.39	0.3945
MNLI pretraining	0.45	0.4579	0.41	0.4038

Table 1: Pretraining Performance on respective test Set and Baseline performance of pretrained XLM-R and mBERT on GLUECoS (Hinglish)

3.5 Multimodal Attention for NLI

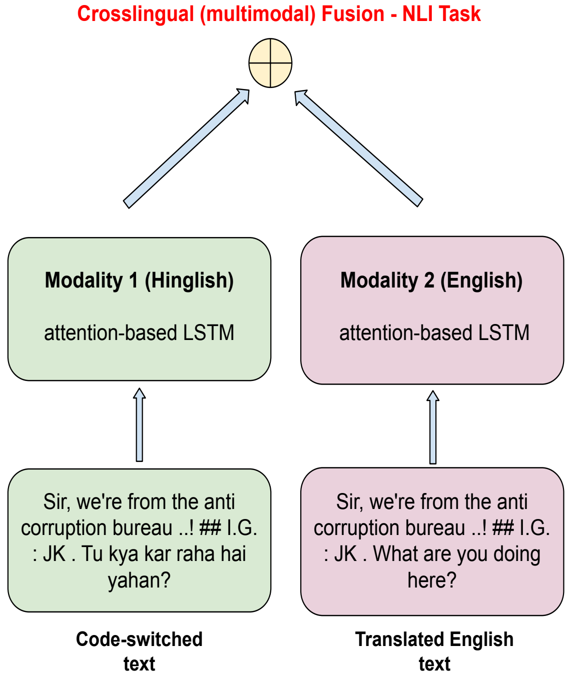
In this approach, we fuse information across different modalities namely, the code-switched Hinglish text and the English text obtained from translation. This approach is motivated by the work of Poria et al. (2017). Attention on the individual modalities allows us to focus on the most important parts of an individual modality, which in our case is individual words in the code-switched Hinglish text and English text respectively. The attention-based fusion mechanism, amplifies the higher quality and informative modalities during fusion in the NLI task.

We hypothesize that the multistage pipeline for converting code-switched Romanized text to English that involves the transliteration and translation of the Romanized Hinglish text to Devanagari Hindi and then translation of this text to Romanized English would result in error propagation through the pipeline as shown in figure 4 and discussed in detail in section 5. While the romanized English corpus performs much better on the NLI task due to the large amount of NLI pre-training data available for English, the errors propagated through the pipeline might adversely impact the NLI performance, and hence it might be beneficial to have an attention over the code-switched text as well. The complete architecture is illustrated in figure 3. We performed a comparative analysis to compare the Multimodal Attention approach for NLI with the fine-tuned approach mentioned in subsection 3.3. Detailed results are discussed in the section ??.

4 Experimental Analysis and Results

4.1 Dataset

We use the GLUECoS dataset (Khanuja et al., 2020) for the final evaluation of the various approaches proposed and to determine the best performing approach. The dataset contains 2,240 unique code-mixed premise-hypothesis pairs and their corresponding labels, with a 80:20 train-test split. It comprises Hindi-English code-switched conversational dialogues from Bollywood movies, and spans several tasks for code-switched Hindi-English such as Named Entity Recognition, Sentiment Analysis, Question Answering and Natural Language Inference. We make use of the NLI labels for our work. In addition, we also make use of the CMU Hinglish DoG dataset (Zhou et al., 2018), and the en-hi-codemixed-corpus (Dhar et al., 2018). These corpora provide parallel data from code-switched Hindi-English to English translations, consisting of a total of over 12000 parallel sentences (around 6000 in each corpus). This allows us to train a translation model directly without going through the intermediate steps of transliteration to Devanagari Hindi followed by translation.



Sir, we're from the anti corruption bureau ..! ## I.G. : JK . Tu kya kar raha hai yahan?

Sir, we're from the anti corruption bureau ..! ## I.G. : JK . What are you doing here?

Figure 3: Architecture of Multimodal Attention for NLI

To pretrain our NLI models, we additionally make use of the MNLI, and XNLI datasets. MNLI is one of the largest English NLI datasets (433k examples) with data from ten distinct genres of written and spoken English. XNLI consists of 7500 human-annotated development and test NLI examples in three-way classification format in 15 languages comprising English, French, Spanish, German, Hindi etc., making a total of 112,500 annotated pairs. This corpus is designed to evaluate cross-lingual sentence understanding, where mod-

Model	Dataset					
	Hinglish		Hindi (Transliterated + Translated)		English (Translated)	
Pretraining-Model	Macro-F1	Acc	Macro-F1	Acc	Macro-F1	Acc
XNLI-XLMR	0.49	0.4897	0.59	0.5973	0.61	0.6116
MNLI-XLMR	0.59	0.6093	0.51	0.5234	0.65	0.6486
XNLI-mBERT	0.46	0.4593	**0.62**	**0.6285**	0.60	0.5967
MNLI-mBERT	0.52	0.5163	0.49	0.4933	0.63	0.6294
MLM-XNLI-XLMR	0.54	0.5446	0.52	0.5189	0.63	0.6233
MLM-MNLI-XLMR	**0.65**	**0.6532**	0.45	0.4497	0.68	**0.6876**
MLM-XNLI-mBERT	0.56	0.5589	0.56	0.5613	0.59	0.5932
MLM-MNLI-mBERT	0.65	0.6472	0.47	0.4765	0.59	0.6087

Table 2: Finetuning pretrained XLM-R and mBERT on GLUECoS

els have to be trained in one language and tested on a different language.

4.2 Segmented Translation and Transliteration

In this method of converting code-mixed Hinglish to English, we handle Romanized Hindi and mixed English words in the inputs separately. After preprocessing the Hinglish text in GLUECoS, we tokenized it at the word level. Next, we identified if the tokens are valid English words or not by using the PyEnchant library. Internally, the library determines the validity using a dictionary based lookup. All the invalid English words were assumed to be Romanized Hindi.

After this, we segmented the inputs into longest possible continuous sub-sequences of English and Romanized Hindi tokens. Romanized Hindi was then transliterated into Devanagari Hindi using the Attention based Seq2Seq model by (Singh et al., 2018). On the other hand, the English segments were translated into Devanagri Hindi using a pre-trained MarianMT model. All the Hindi segments are then concatenated to form the final outputs. The obtained Devanagri Hindi is then again translated to English using the MarianMT model. This method has some limitations which are discussed in section 5.

Since GLUECoS does not have ground truth English translation data, we evaluate the performance of the translation on the CMU Hinglish dataset which is an extended code-mixed form of the Document Grounded Conversations (Zhou et al., 2018) dataset. This source thus contains both code-mixed Romanized Hinglish and corresponding English translations. BLEU score of the obtained translation on the validation set is mentioned in Table 3.

4.3 Direct Translation: Code-switched Hinglish to English

For direct translation to the high-resource language (Code-sqitched Hinglish to English) we used Google's pre-trained *mt5-small* model (Xue et al., 2020) from HuggingFace. Although, (Agarwal et al., 2021) uses *mt5-base*, we had to use *mt5-small* because of training compute and time limitations.

We fine-tuned *mt5-small* first on the back-translation from English to Romanized Hinglish so that the model learns to understand Hinglish as a language. We then, further fine-tuned on the forward translation from Romanized Hinglish to English to get our final model. The finetuning was done using CMU Hinglish DoG dataset (Zhou et al., 2018), and the en-hi-codemixed-corpus (Dhar et al., 2018) datasets. We evaluated the translation performance on the CMU Hinglish validation set. BLEU score of the obtained translation is reported in Table 3. From these results we can conclude that the translation quality using this method as compared to the method mentioned is Section 4.2 is better. So we use the translations from this method for the downstream tasks.

4.4 Pretraining and Finetuning on GLUECoS

As discussed previously, we pretrain the multilingual models with MLM and NLI objectives. To establish a baseline we first pretrain XLM-R and

Method	BLEU Score
Segmented Translation and Transliteration	14.65
Direct Translation	22.30

Table 3: BLEU Scores obtained for Romanized Hinglish to English translation in the CMU Hinglish DoG dataset

mBERT for 10 epochs each on the Multi-NLI and the XNLI datasets. The training is performed in parallel on an 8 gpu AWS p2.8xlarge instance with a batch size of 64. Each epoch takes about 1 hour 15 minutes to complete. The performance of these models on the test sets of these datasets after pretraining is reported in Table 1. We then perform zero-shot evaluation of these models on the GLUECoS dataset. We observe a baseline accuracy of about 44% as shown in Table 1.

Next we use segmented transliteration and translation to convert the code-mixed Hinglish corpus into monolingual Hindi (as described in Section 4.2). We also used Direct Translation (as described in Section 4.3) to translate the Hinglish corpus into monolingual English. We then finetune the pretrained XLM-R and mBERT models on the GLUECoS corpus as well as the parallel monolingual Hindi and monolingual English corpora obtained after translation.

This leads to several interesting observations. Firstly we find that pretraining with the NLI objective on large monolingual or multilingual corpora before finetuning indeed helps improve NLI performance (about 10% increase) on the code-mixed Hinglish corpus. This implies that pretraining on similar datasets can facilitate extensive crosslingual transfer learning on new datasets thus validating our hypothesis.

Secondly we observe that translating the code-mixed corpus into either its matrix or embedded language also improves the NLI performance. We postulate that this is because of the fact that the translated monolingual corpus is in domain with the corpora that was used in training the multilingual language models.

Thirdly we also observe that pretraining with MNLI which is solely a monolingual english corpus results in state-of-the-art NLI performance on the translated English data with XLM-R (64.86%) that beats both the highest accuracy on the GLUECoS leaderboard (57.74%) as well as that reported by Chakravarthy et al. (2020) (63.69%).

Furthermore, consistent with our expectations pretraining with XNLI (crosslingual) gives better performance on the translated Hindi corpus with both mBERT and XLM-R than pretraining with MNLI (monolingual). This is because pretraining with XNLI causes the model to learn NLI on Devanagiri Hindi representations resulting in better performance on the monolingual Hindi data. Finally we also observe that XLM-R performs better than mBERT in almost all our experiments. This could attributed to the fact that XLM-R has a larger vocabulary and thus is considerably larger and stronger model than mBERT. These results are detailed in the top half of Table 2.

4.5 Incorporating the Masked Language Modelling Objective

Motivated by the idea of developing a better understanding of the code-mixed language we first pretrain XLM-R and mBERT with the MLM objective on the augmented dataset as described in Section 3.4. We then replace the LM head with a classification head and further pretrain both the models with the NLI objective on XNLI and the MNLI datasets. We train the model with MLM objective for 10 epochs, with a batch size of 4 and each epoch takes about 40 minutes to complete on an AWS g4dn instance which consists of a single GPU with 16GB GPU memory.

We find that incorporating MLM as a pretraining objective further improves NLI performance. Most notably the inclusion of MLM with XLM-R in pre-pretraining followed by pretraining XLM-R on MNLI and then finetuning XLM-R on GLUECoS allows us to obtain state-of-the-art accuracies using just Hinglish (65.32%) or even better accuracy using the translated English (68.76%). We hypothesize that these results can again be attributed to superior code-switched language understanding that is being developed during the MLM pretraining phase and the extensive crosslingual transfer that happens during the NLI pretraining and finetuning phases.

Another point of interest is the consistent decrease in performance on Hindi while pretraining with MLM. We postulate that this happens because of the domain mismatch issue in the Devanagiri Hindi Script with the Romanized Hinglish/English scripts which is further amplified during pretraining with MLM. The results of our experiments are summarized in the bottom half of Table 2.

4.6 Multimodal Attention for NLI

We apply multimodal attention on the code-switched Hinglish text and the English text. Since GLUECoS dataset does not have English translations for code-switched Hinglish, we use the *mt5-small* model that we finetuned for code-switched Hinglish translations to obtain the English translations for GLUECoS data. The rationale behind applying multimodal attention and choosing to apply attention heads on these 2 inputs has been discussed in subsection 3.5. We use the code base provided by (Poria et al., 2017) and modify it to work for the NLI task with our input types. Specifically, we make use of attention-based LSTMs for each unimodal branch.

We train for 15 epochs and use a batch size of 20. The training is performed on an AWS g4dn.xlarge instance. The training time for each epoch is around 12 minutes. We directly train the model on the training subset of the GLUECoS dataset, and test it on the test subset. We obtain an accuracy of 54.62% and a Macro-F1 score of 0.53. Both the scores are lower than what we obtained for XLM-R using pretraining with either MNLI or XNLI datasets. Our hypothesis is that the low score is due to the use of overly simplistic LSTM architectures to solve the complex NLI task. We believe that making use of state-of-the-art transformer-based networks would considerably boost performance of the multimodal attention network. Such a multimodal attention network based on transformers might be able to achieve competitive performance to the pretrained models as well. However, due to time constraints, we were unable to conduct experiments with transformer-based networks.

5 Error Analysis

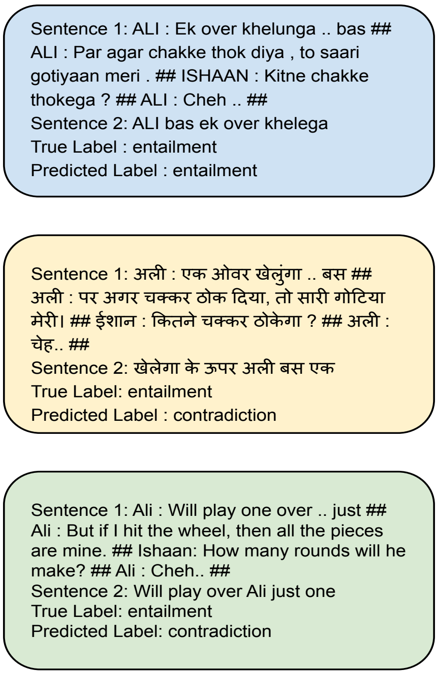
We performed a qualitative analysis of the two stage transliteration and translation pipeline from Hinglish to English. Figure 4 shows an example of a failure scenario where an error during transliteration from Romanized Hindi to Devanagari Hindi results in the error getting propagated into the subsequent translation from Devanagari Hindi to English. In the provided example, the first block has repeated mentions of the Hindi word ‘chakke’ which means a Six in the game of cricket. During the transliteration ‘chakke’ gets transliterated to चक्कर which means round in Hindi. Subsequently, the Devanagari Hindi word चक्कर gets translated to English as ’round’. As a result the segmentation based approach for transliteration eventually results in incorrect Devanagari Hindi and Romanized English translation outputs. This ultimately leads to the predicted label to be different from the ground truth label for the NLI task as shown in Blocks 2 and 3 of the diagram. We remedied this issue by using direct translation as described in section 4.3 to translate Hinglish to English and found that the translation quality was much superior.

Another common type of error is the non-standard form for Hindi words in the Roman script. For example, the Hindi for ‘No’ in Roman script is written in varied forms throughout the datasets like ‘nahi’, ‘nhi’ or ‘ni’. This lack of standardization is commonly seen when a different script is used for a language than its native script. Such words are incorrectly translated by our finetuned *mt5-small* due to variations in the training corpus which leads to improper training. In the translation pipeline, there is ambiguity in the outputs of transliteration for such words. Another common form of error in the translation pipeline is that several English words have the same spelling as the Hindi words in Roman script. Examples of such words include the preposition ‘to’ in English which has the same spelling as ‘to’ in Roman Hindi meaning ‘so’, or the word ‘agar’ which is a gelatinous food substance in English, but it means ‘if’ in Roman Hindi. These kinds of words induced errors in our translation and transliteration pipeline.

6 Conclusion

Based on our extensive experimentation and analysis, we were able to justify the hypothesis that NLI performance improves significantly (around 4%) by translating the code-mixed corpus to a high resource matrix or embedded language. We also discovered that the learned representations of the code-switched language within the language models can be improved considerably through domain adaptation achieved by pretraining with the MLM objective such that the NLI performance on the code-switched Hinglish text (0.6532) is almost comparable to the NLI performance on the translated English text (0.6876). This implies that for a compute constraint environment where performing the translation and transliteration is computationally not feasible, our proposed solution provides an alternative approach to use code-switched data directly without a large impact on NLI performance.

We compared the translation performance of



Sentence 1: ALI : Ek over khelunga .. bas ##
ALI : Par agar chakke thok diya , to saari gotiyaan meri . ## ISHAAN : Kitne chakke thokega ? ## ALI : Cheh .. ##
Sentence 2: ALI bas ek over khelega
True Label : entailment
Predicted Label : entailment

Sentence 1: अली : एक ओवर खेलुंगा .. बस ##
अली : पर अगर चक्कर ठोक दिया, तो सारी गोटिया मेरी। ## ईशान : कितने चक्कर ठोकेगा ? ## अली : चेह.. ##
Sentence 2: खेलेगा के ऊपर अली बस एक
True Label: entailment
Predicted Label : contradiction

Sentence 1: Ali : Will play one over .. just ##
Ali : But if I hit the wheel, then all the pieces are mine. ## Ishaan: How many rounds will he make? ## Ali : Cheh.. ##
Sentence 2: Will play over Ali just one
True Label: entailment
Predicted Label: contradiction

Figure 4: An example of error propagation in the translation and transliteration pipeline

our translation and transliteration pipeline, and direct translation from Hinglish to English using Google's pre-trained *mt5-small* model in terms of their BLEU scores and found that direct translation works much better both qualitatively and quantitatively. In addition, we also attempted to further improve NLI performance using a Multimodal Attention technique for which the results seem promising, but further changes are needed in the architecture and pretraining methodologies for the technique to perform as expected.

To summarize we have demonstrated how extensive crosslingual transfer through pretraining with different objectives helps us obtain considerably higher performance (about 5%) than the current state-of-the-art.

7 Contributions and Future Work

Our next steps would be to further enhance the domain adaptation phase by incorporating larger corpora for pretraining with MLM such as PHINC (Srivastava and Singh, 2020) and IITG-HingCoS (Ganji et al., 2019) which we could not include in our current experimentation due to time and resource constraints. Our hypothesis is that augmenting our current Hinglish dataset for MLM further with other large Hinglish datasets would provide a stronger signal required by language models for a robust understanding of Hinglish. This could in turn drive the downstream performance over the NLI task even further. We also plan to experiment using transformer architectures for multimodal attention over Hinglish and English. Our error analysis indicates bottlenecks in the current segmentation based approaches used for transliterating and translating code-switched data to the monolingual corpus. Thus better approaches for transliteration which can also account for context is an important direction for future work.We would also like to extend our approaches for Hinglish data to other code-switched languages like Spanglish to see if our observations are generalizable across code-switched languages.

Our code for this work is available on GitHub. Our best model can be downloaded from HuggingFace models.

8 Acknowledgements

We would like to thank all the instructors and the course teaching assistants for their guidance. We would particularly like to thank Prof. Alan Black for his continued mentorship throughout the project and providing us with the required datasets, Prof. Shinji Watanabe for suggesting the Multimodal Attention approach as well as Vijay Viswanathan for providing constructive feedback on our approaches, and helping us streamline our experimental setup which was instrumental in helping us define the scope of the project and complete it on time.

9 Individual Team Member Contributions

All team members contributed equally. The following were the areas of focus of each team member
Kushagra Mahajan: Language models using MLM objective for NLI
Nikhil Gupta: Translation and Transliteration Pipeline, Direct Translation
Shubham Phal: NLI Pretraining and Finetuning

References

Vibhav Agarwal, Pooja Rao, and Dinesh Babu Jayagopi. 2021. Towards code-mixed hinglish dialogue generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 271–280.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference.

In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Sharanya Chakravarthy, Anjana Umapathy, and Alan W Black. 2020. Detecting entailment in code-mixed hindi-english conversations. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 165–170.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. 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.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Sreeram Ganji, Kunal Dhawan, and Rohit Sinha. 2019. Iitg-hingcos corpus: A hinglish code-switching database for automatic speech recognition. *Speech Communication*, 110:76–89.

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. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Mazumder, Amir Zadeh, and Louis-Philippe Morency. 2017. Multi-level multiple attentions for contextual multimodal sentiment analysis. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1033–1038. IEEE.

Archiki Prasad, Mohammad Ali Rehan, Shreya Pathak, and Preethi Jyothi. 2021. The effectiveness of intermediate-task training for code-switched natural language understanding. *arXiv preprint arXiv:2107.09931*.

Kushagra Singh, Indira Sen, and Ponnurangam Kumaraguru. 2018. Language identification and named entity recognition in Hinglish code mixed tweets. In *Proceedings of ACL 2018, Student Research Workshop*, pages 52–58, Melbourne, Australia. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2020. Phinc: A parallel hinglish social media code-mixed corpus for machine translation. *arXiv preprint arXiv:2004.09447*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.