Detecting Entailment in Code-Mixed Hindi-English Conversations

Sharanya Chakravarthy* **Anjana Umapathy*** **Alan W Black**
Language Technologies Institute
Carnegie Mellon University
{sharanyc, aumapath, awb}@cs.cmu.edu

Abstract

The presence of large-scale corpora for Natural Language Inference (NLI) has spurred deep learning research in this area, though much of this research has focused solely on monolingual data. Code-mixing is the intertwined usage of multiple languages, and is commonly seen in informal conversations among polyglots. Given the rising importance of dialogue agents, it is imperative that they understand code-mixing, but the scarcity of code-mixed Natural Language Understanding (NLU) datasets has precluded research in this area. The dataset by Khanuja et al. (2020a) for detecting conversational entailment in code-mixed Hindi-English text is the first of its kind. We investigate the effectiveness of language modeling, data augmentation, translation, and architectural approaches to address the code-mixed, conversational, and low-resource aspects of this dataset. We obtain +8.09% test set accuracy over the current state of the art.

1 Introduction

Natural Language Inference (NLI) is a widely researched NLP task which involves determining if a premise entails or contradicts a hypothesis. The performance of machine learning models on this task has important implications for other Natural Language Understanding tasks such as Question Answering, Semantic Search and Text Summarization. While large corpora such as SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018) are available for monolingual and cross-lingual NLI, Khanuja et al. (2020a) introduce the first NLI dataset with Hindi-English (Hinglish) text. We refer to this dataset as CS-NLI.

Code-mixing is a phenomenon prevalent in multilingual communities (Claros and Isharianty, 2009). It poses a number of interesting challenges for NLP applications, such as the mixing of units from multiple grammar systems, morphological differences between monolingual and code-mixed text due to the intermixing of affixes, and non-standard transliteration between the writing systems involved. In CS-NLI, Hindi is present in a non-standard Romanized form. Multilingual speakers most often code-mix in informal settings such as social media, in-person, and telephonic conversations, due to which there is a dearth of clean, large-scale code-mixed corpora such as Wikipedia articles and books that can be used for pre-training, making this a low-resource task.

Khanuja et al. (2020a) leverage Bollywood movie scripts containing Hinglish text to create CS-NLI, with conversations as premises. The creation of hypotheses based on dialogue-like premises transforms the task from one of textual entailment to one of conversational entailment. The inclusion of scripts from multiple movies makes this data inherently noisy due to non-standard Romanization of Hindi, the variation in dialects across movies and differing grammar styles among Hinglish speakers.

In this work, we explore and analyze a variety of techniques to leverage existing pre-trained models such as BERT (Devlin et al., 2019) for processing code-mixed and conversational text. We present a comparison of linguistic, data augmentation and architectural approaches to conversational entailment in code-mixed text. We show multiple techniques that interestingly give similar results, while also beating the current state of the art[1]. The code for the approaches described in this paper will be made available on GitHub [2].

2 Related Work

NLI for monolingual and cross-lingual text is a well-researched task that has been addressed using a variety of techniques including neural networks, symbolic logic and knowledge bases (Bowman

[*] Equal contribution

[1] https://microsoft.github.io/GLUECoS/

[2] https://github.com/sharanyarc96/HinglishNLI

Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text, pages 165–170
Online, Nov 19, 2020. ©2020 Association for Computational Linguistics

et al., 2015). The use of transformer models such as BERT and RoBERTa (Liu et al., 2019), pre-trained on large monolingual corpora, has advanced the state of the art on the SNLI and MultiNLI datasets. While unsupervised pre-training of deep learning models has been shown to improve performance on a variety of NLP tasks, the limited amount of data available precludes large-scale pre-training on code-mixed text. Multilingual BERT (mBERT) (Devlin et al., 2019) is pre-trained on monolingual Wikipedia corpora from 104 languages, including Hindi in its original Devanagari script. XLM-RoBERTa (XLM-R) (Conneau et al., 2020) is trained on the CommonCrawl corpus, which includes Romanized Hindi text, making this model the closest one to being pre-trained on Hinglish.

3 Task Definition

Khanuja et al. (2020a) introduce a dataset spurring two challenging directions of research - NLI for code-mixing, and conversational entailment. The dataset contains 2,240 unique code-mixed premise-hypothesis pairs and their corresponding labels, with an 80:20 train-test split. We tackle the binary classification task of assigning an ENTAILMENT label if the premise entails the hypothesis and a CONTRADICTION label if the premise contradicts the hypothesis. Premises are in the form of multiple utterances from a conversation, with each utterance preceded by the name of the speaker. For example-

Premise (Code-Mixed): RAHUL : Tumhara scooter aur ek joota security guard ko lobby mein mila . ## RIANA : Thank god !!

Premise (Translation): RAHUL : The security guard found your scooter and one shoe in the lobby. ## RIANA : Thank god !!

4 Methodology

Given the success of pre-trained models on other NLI tasks, we tackle this task by fine-tuning BERT, mBERT and XLM-R for sentence-pair classification. Due to the scarcity of examples in CS-NLI, we focus our efforts on the modification and augmentation of the data used to fine-tune these models. In this section, we describe techniques to address the code-mixed, low-resource, and conversational aspects of the task.

4.1 Addressing Code-Mixing

We use approaches such as language modeling, transliteration, and translation to alleviate the absence of code-mixing in the data used to pre-train transformer models.

Masked Language Modeling: We fine-tune mBERT on the masked language modeling objective, following Khanuja et al. (2020b), on a combination of in-domain code-mixed movie scripts and publicly available datasets by Roy et al. (2013) and Bhat et al. (2018) to obtain modified mBERT (mod-mBERT) to be fine-tuned on the sentence-pair classification task.

Transliteration: We perform token-level language identification and transliterate the detected Romanized Hindi words in CS-NLI to Devanagari script using the approach in Singh et al. (2018), to enable mBERT to better understand them.

Translation: Due to the difficulty in training code-mixed to monolingual translation models, we follow the approach in Dhar et al. (2018) to obtain translations. We first transliterate the Romanized Hindi words, and then translate English phrases to Hindi using the Google Cloud translation API. [3].

4.2 Addressing the Low-Resource Aspect

Due to the limited amount of code-mixed NLI data available for fine-tuning, we augment CS-NLI with 4000 monolingual entailment and contradiction examples sampled from the SNLI, XNLI (Conneau et al., 2018), and MPE (Lai et al., 2017) datasets. Transliterations of Devanagari Hindi sentence-pairs from the XNLI dataset provide additional NLI data in Romanized Hindi while SNLI examples do the same in English. The MPE dataset adds examples requiring aggregation of information across sentences (Lai et al., 2017).

4.3 Approaches to Conversational NLI

Each premise in CS-NLI contains turns in the form "Speaker Name: Utterance". Khanuja et al. (2020a) show that a number of hypotheses require an understanding of the transition between speakers, in addition to the meaning of the utterance itself. In order to estimate whether BERT understands the role of speakers, we remove speaker names occurring before each utterance, and fine-tune the models on CS-NLI. We find that the accuracy does not deteriorate, indicating that the BERT models may benefit from reinforcing speaker roles.

Data Augmentation with Speaker Names: Khanuja et al. (2020a) present a set of examples that involve swapping roles. We generate additional CONTRADICTION examples for role

[3] https://cloud.google.com/translate/docs/quickstarts

Example	Premise	Hypothesis	Label
Original	KRITI: VIKAS pad raha hai ## VARUN: Which subject? ## VIKAS: Physics	VIKAS pad raha hai	Entailment
Translation	KRITI: VIKAS is studying ## VARUN: Which subject? ## VIKAS: Physics	VIKAS is studying	Entailment
Contradiction Augmentation	KRITI: VIKAS pad raha hai ## VARUN: Which subject? ## VIKAS: Physics	VARUN pad raha hai	Contradiction
Speaker Name Augmentation	VEENA: MADAN pad raha hai ## ARJUN: Which subject? ## MADAN: Physics	MADAN pad raha hai	Entailment

Table 1: Example of augmentation of CS-NLI by modifying speaker names. Original: Example from CS-NLI, Contradiction Augmentation: Adding a contradiction example by modifying a name in a hypothesis from an entailment hypothesis, Speaker Name Augmentation: Adding an entailment example by modifying all the names in an entailment example

swapping by modifying speaker names found in the hypotheses of ENTAILMENT examples. We augment the existing dataset with examples which differ only in the names of the speakers, with the goal of helping the model to focus on the role of speaker names in detecting entailment. Examples of these augmentation techniques are shown in Table-1.

Utterance Representations using BERT: The premises in CS-NLI contain multiple turns of a conversation. Since BERT is commonly used for single-sentence representations, we encode each turn separately using mod-mBERT. We obtain utterance representations from mod-mBERT and pass them through a bidirectional LSTM (biLSTM). We concatenate the initial and final hidden states of the biLSTM with the mod-mBERT encoding of the hypothesis, and pass them through an MLP with two linear layers to obtain a classification output.

5 Experimental Setup

In the majority of our approaches, we fine-tune BERT, mBERT, mod-mBERT (110M parameters), and XLM-R (550M parameters) for 1 to 6 epochs on an Nvidia GeForce GTX 1070 GPU. We experiment with batch sizes of 8,16, and 32, and learning rates between 1e-5 and 5e-5, and report results using a batch size of 8 and learning rate of 1e-5.

6 Results and Analysis

On fine-tuning the BERT models on CS-NLI, we observe a large variation in the results based on the subset of data used for evaluating the model, as demonstrated in Table-2. To address this variation, we perform eight-fold cross validation with early stopping, and report the mean and standard deviation of the accuracies across eight splits. These results are shown in Table-3. We evaluate the models with the highest cross-validation accuracy on the test set and report these results in Table-4.

Split	Accuracy	Split	Accuracy
1	65.91%	5	60.09%
2	56.50%	6	62.78%
3	57.85%	7	61.71%
4	64.57%	8	58.10%

Table 2: mBERT - Cross-validation accuracy variation

In this section, we provide qualitative and quantitative analysis of our approaches. The qualitative analysis is performed on the cross-validation splits.

6.1 Comparison of Pre-Trained Models

The majority of Hindi words in the NLI dataset are out of vocabulary for BERT. Nevertheless, it obtains a high cross-validation accuracy of 61.11%. We believe it achieves this by tuning the embeddings of WordPiece tokens of both Hindi and English text present in the dataset. To verify that it does not learn only from in-vocabulary English words, we fine-tune BERT after removing the words identified as Hindi, and find that its performance deteriorates sharply.

The benefit of mBERT's multilingual pre-training seems to be lost in CS-NLI due to the script mismatch between Devanagari Hindi used to pre-train mBERT, and Romanized Hindi in CS-NLI.

mod-mBERT performs better than BERT and mBERT due to its enhanced understanding of Hinglish. We believe that fine-tuning on in-domain movie scripts increases mBERT's understanding of conversational code-mixed text, while the inclusion of code-mixed text from other sources enables it to better understand non-standard Romanization.

Model Name	**Mean Acc.**	**Std. Dev.**
FINE-TUNING PRE-TRAINED MODELS		
BERT	61.11%	3.38
mBERT	60.94%	3.16
mod-mBERT	61.28%	2.08
TRANSLITERATION & TRANSLATION (MBERT)		
Transliteration of CS-NLI	62.17%	2.00
Hi translation of CS-NLI	60.04%	3.71
CS-NLI & its Hi translation	63.30%	3.05
AUGMENTATION OF CS-NLI		
mod-mBERT on 3k XNLI	**63.69%**	**1.58**
mod-mBERT on 4k SNLI & 4k XNLI	63.35%	2.53
mod-mBERT on 4k MPE	62.19%	3.11
XLM-R on 4k SNLI & 4k XNLI	63.52%	1.85
CONVERSATIONAL APPROACHES (MOD-MBERT)		
CS-NLI & Speaker Name Augmentation	62.85%	2.00
CS-NLI & Speaker Name, Contradiction Augmentation	61.39%	1.87
biLSTM	54.83%	1.72

Table 3: Results on 8-fold cross validation. Hi: Hindi

Although XLM-R is the only model which contains Romanized Hindi in its pre-training data, the model does not converge when fine-tuned on just CS-NLI. However, on augmentation with monolingual NLI examples, there is a large improvement in performance as shown in Table-3. The output of XLM-R's tokenizer shows that many of the Romanized Hindi words are in the model's vocabulary, in contrast to BERT and mBERT where the words get broken into multiple WordPiece tokens. Despite this fact, the model is unable to fit the training data even with an extensive hyper-parameter search, leading us to hypothesize that larger amounts of data are required for fine-tuning XLM-R. However, the performance of this model on code-mixed datasets bears further investigation.

6.2 Transliteration and Translation

Manual inspection shows that errors in language identification and transliteration result in noisy translated and transliterated versions of the data, deterring the performance. However, we find that augmenting the original training set with its translations allows the model to learn from code-mixed and monolingual forms of the same examples.

Model	**Acc.**
mBERT for 5 epochs i.e. w/o early stopping (baseline)	54.32%
BERT	58.83%
mBERT	60.85%
mod-mBERT	**62.41**%
mBERT on CS-NLI, Hi translation of CS-NLI	56.37%
mod-mBERT on CS-NLI, XNLI	56.82%
mod-mBERT on CS-NLI, SNLI, XNLI	58.16%
XLM-R on CS-NLI, SNLI, XNLI	57.49%

Table 4: Results on the test set. We perform early-stopping while fine-tuning our models. Since we have 8 cross-validation splits, we stop on the epoch that most frequently gives the highest accuracy across these splits.

6.3 Data Augmentation

Although the SNLI, XNLI and MPE datasets contain monolingual examples of textual, non-conversational entailment, augmenting the data with examples from these datasets improves the performance of the models. We believe this is because the addition of these examples aids their general understanding of entailment. The mismatch between the nature of the entailment tasks poses the question of whether there exists an optimal subset and quantity of external data for augmentation. We were unable to find a correlation between the performance and number of external examples added. Finding the categories, if any exist, of examples that are most helpful to the model is challenging. Possible strategies include selection based on length, language complexity, dialect, and domain similarity in the case of Hindi XNLI data. In this work, however, we take a random sample of examples from these corpora.

Since each of these augmentation techniques improve the performance of the model, we augment CS-NLI with different combinations of the datasets, shown in Table-3. We observe an improvement, although it is not proportional to that of the individual augmentations.

6.4 Utterance Representations Using BERT

Separating utterance representations performs worse than the majority of our approaches. The addition of biLSTM layers over the BERT model introduces a large number of uninitialized parameters. We believe that the scarcity of data available to train these parameters leads to its poor performance. Further, the lack of an attention mechanism between utterances and the hypothesis may also pose a problem.

6.5 Qualitative Analysis

Khanuja et al. (2020a) provide an analysis of the various kinds of examples present in CS-NLI. We attempt to discern similarities in the examples that the various models predict incorrectly in order to better address these classes of examples. We analyze various statistical properties of the premises such as their length, the number of turns in the conversation, and the number of distinct speakers, and observe no correlation between these properties and the correctness of the model's predictions. While the complexity of the Hindi and English vocabulary used may make some code-mixed examples more difficult than others, automatically identifying such differences is difficult.

McCoy et al. (2019) show that most neural models including BERT are expected to accurately predict examples involving negation, role swapping, paraphrasing and numerical changes, such as those shown in Khanuja et al. (2020a). However, cross-lingual paraphrasing and negation in CS-NLI make it hard to detect these otherwise simple examples in code-mixed settings.

We evaluate the ability of BERT models to recognize role-swapping by generating examples of this nature. We find that mod-mBERT trained on CS-NLI only predicts 19% of these examples correctly, whereas a model trained using the speaker name data augmentation technique described in Section-4.3, with weighted cross-entropy loss, gets an accuracy of 87% on these examples, substantiating this approach.

6.6 Performance on the Test Set

The accuracy of mBERT with early stopping is 6% higher than the baseline. mod-mBERT shows the best performance with an accuracy that is 8% higher than the baseline, while the augmentation and modification approaches seem to reduce the performance of the model. We attribute the large difference between the test set and cross-validation accuracies to the sensitivity of models to different splits in the dataset, as shown in Table-2.

7 Conclusion

Our results show that there is a long way to go in NLP for code-mixed language tasks. Even using standard techniques such as multilingual language modeling and data augmentation, our results are still behind an equivalent task in a high resource environment.

Although this dataset contains higher level challenges such as sarcasm detection that are not yet solved even in high-resource languages, even phenomena such as negation, role swapping and paraphrasing become challenging due to code-mixing.

Code-mixed language pairs can be thought of as a separate language (Sitaram et al., 2019), and perhaps large-scale pre-training on code-mixed data would be able to push the boundaries of code-mixed interpretation, as has been the case with high-resource languages.

References

Irshad Bhat, Riyaz A. Bhat, Manish Shrivastava, and Dipti Sharma. 2018. Universal dependency parsing for Hindi-English code-switching. 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 987–998, New Orleans, Louisiana. Association for Computational Linguistics.

Samuel 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*, pages 632–642.

Monica Stella Cardenas Claros and Neny Isharianty. 2009. Code-switching and code-mixing in internet chatting: between 'yes,''ya', and 'si': a case study. *Jaltcall*, 5:67–78.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel 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*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

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.

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

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020b. 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.

Alice Lai, Yonatan Bisk, and Julia Hockenmaier. 2017. Natural language inference from multiple premises. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 100–109, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*.

Rishiraj Saha Roy, Monojit Choudhury, Prasenjit Majumder, and Komal Agarwal. 2013. Overview of the fire 2013 track on transliterated search. In *Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval Evaluation*, pages 1–7.

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.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

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.