# CoMix: Guide Transformers to Code-Mix using POS structure and Phonetics

**Gaurav Arora**
Amazon
gaurvar@amazon.com

**Srujana Merugu**
Amazon
smerugu@amazon.com

**Vivek Sembium**
Amazon
viveksem@amazon.com

## Abstract

Code-mixing is ubiquitous in multilingual societies, which makes it vital to build models for code-mixed data to power human language interfaces. Existing multilingual transformer models trained on pure corpora lack the ability to intermix words of one language into the structure of another. These models are also not robust to orthographic variations. We propose CoMix[1], a pretraining approach to improve representation of code-mixed data in transformer models by incorporating phonetic signals, a modified attention mechanism, and weak supervision guided generation by parts-of-speech constraints. We show that CoMix improves performance across four code-mixed tasks: machine translation, sequence classification, named entity recognition (NER), and abstractive summarization. It also achieves new SOTA performance for English-Hinglish translation and NER on LINCE Leaderboard and provides better generalization on out-of-domain translation. Motivated by variations in human annotations, we also propose a new family of metrics based on phonetics and demonstrate that the phonetic variant of BLEU correlates better with human judgement than BLEU on code-mixed text.

## 1 Introduction

Code-mixing, i.e., embedding linguistic units of one language (*embedded language* $L_E$) into a sentence grammatically structured as per another language (*matrix language* $L_M$), is common in multilingual communities. Growing mobile penetration coupled with the increased adoption of informal conversational interfaces is leading to further rise in such communication. Currently, over 20% of user generated content from South Asia and parts of Europe is code-mixed (Choudhury et al., 2019). Hinglish (code-mixed Hindi-English) has nearly 350 million speakers (GTS, 2019) making it one of the most widely spoken languages. Recent literature suggests that multilingual users associate code-mixing with cultural affinity and prefer chatbots that can code-mix (Bawa et al., 2020). Code-mixed modeling is, thus, a foundational prerequisite for linguistic systems targeted towards such users.

Transformer models such as BART (Lewis et al., 2020) and BERT (Devlin et al., 2018) have been successful across various NLP tasks. These models can readily capture code-mixing semantics if a large corpus was available for training. Unfortunately, that is not true for most code-mixed languages. Existing approaches rely on learning from a parallel corpus of embedded and matrix languages (e.g., English and Hindi for Hinglish). Recent work (Chen et al., 2022), however, shows that multilingual models such as mBERT trained on monolingual sources fail to effectively interleave words from topologically diverse languages.

Adapting transformers to code-mixed data requires addressing the following challenges: **1. Divergent grammatical structure.** For code-mixed languages such as Hinglish, where $L_E$ and $L_M$ have different Parts-of-Speech (POS) patterns, models trained on monolingual corpora do not yield similar representations for equivalent words across languages, which is needed to facilitate interleaving of $L_E$ and $L_M$ words. Linguistic theories propose certain syntactic constraints for code-mixed generation (Poplack, 1980), but these are not usually incorporated into the modeling. **2. Code-mixing diversity.** Code-mixed languages also exhibit a wide diversity in the degree of code-mixing (e.g., ratio of $L_E$ to $L_M$ words). Fig 1 shows multiple Hinglish constructions for a given sentence in English. Accounting for this variation in code-mixing is necessary for high fidelity modeling. **3. Orthographic variations.** The informal nature of code-mixed interactions and lack of standardized transliteration rules leads to users employing adhoc

[1]CoMix is not a trademark and only used to refer to our models for code-mixed data for presentational brevity.



*Findings of the Association for Computational Linguistics: ACL 2023*, pages 7985–8002
July 9-14, 2023 ©2023 Association for Computational Linguistics

phonological rules while writing code-mixed content. Fig 1 shows Hinglish sentences with similar sounding words and their variations ("kis", "kys").

**Contributions.** In this paper, we adapt transformer models for code-mixed data by addressing the above challenges. To ensure applicability to multiple downstream tasks, we focus on pretraining.

1. We propose CoMix, a set of generic pretraining methods to improve code-mixed data representations that can be applied to any transformer model assuming the availability of POS-tagger and phonetic transcription tools. These include: (a) Domain Knowledge-based Guided Attention (DKGA) mechanism that facilitates intermixing of linguistic units of $L_E$ into the structure of $L_M$ through a modified attention function, (b) Weakly Supervised Generation (WSG) that generates code-mixed data for training in a controllable fashion driven by linguistic constraints, and (c) inclusion of phonetic signals to align embeddings of similar sounding with different orthographic representation.

2. We instantiate CoMix pretraining for BART and BERT and demonstrate efficacy on multiple downstream NLP tasks, namely Machine Translation, NER, Sequence Classification, and Abstractive Summarization with relative improvements of up to 22%. CoMixBART and CoMixBERT achieve new state-of-the-art (SOTA) results for English-Hinglish translation and Hinglish NER tasks on LINCE Leaderboard (Aguilar et al., 2020), beating previous best mT5 (Jawahar et al., 2021) and XLM-R (Winata et al., 2021) models, despite having less than 0.5x and 0.1x model size respectively.

3. We evaluate out-of-domain code-mixed translation performance on two test sets, one created in-house and other one adapted from GupShup corpus (Mehnaz et al., 2021), and show that CoMix generalizes better than other models. To the best of our knowledge, this is the first such evaluation for English-Hinglish translation. We hope our benchmark will assist the community to improve out-of-domain generalization of code-mixed translation, a critical need for low-resource regimes.

4. To address the limitations of existing metrics in handling orthographic variations in code-mixed data, we propose a new family of natural language generation (NLG) metrics based on phonetic adaptation of existing metrics. We observe that PhoBLEU, the phonetic variant BLEU, is better aligned to human judgement (+0.10 - 0.15 on Pearson correlation) than BLEU on Hinglish.

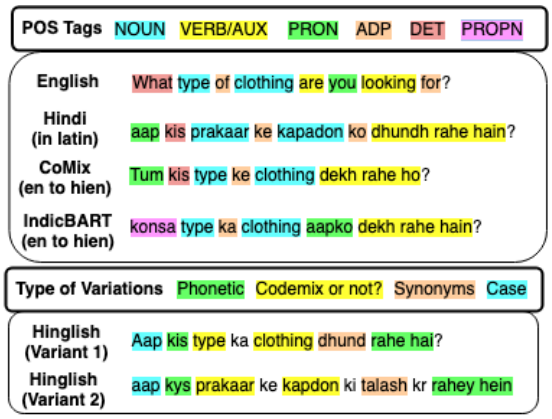



Figure 1: [Top] Divergent POS structure of Hindi ($L_M$) and English ($L_E$) with CoMix output following $L_M$ structure better than that of IndicBART. [Bottom] Different types of valid variations in code-mixed data.

## 2 Related Work

**Multilingual and Code-Mixed NLP.** Recent advances in large multilingual pre-trained models such as mBERT (Devlin et al., 2018) and mBART (Liu et al., 2020) have led to significant gains on many multilingual NLP tasks. However, evaluation of these models on code-mixed content for machine translation (Chen et al., 2022), sequence classification (Patwa et al., 2020), summarization (Mehnaz et al., 2021) and other tasks (Aguilar et al., 2020) points to their inability to intermix words from two languages since these are pretrained on monolingual text without any language alternation. Our CoMix approach encourages the model to learn representations that allows appropriate embedding of words from one language into structure of another via domain knowledge guided attention and through weakly supervised code-mixed generation. Prior work (Sanad Zaki Rizvi et al., 2021) focuses on generating synthetic code-mixed data using constraints from linguistic theories followed by learning. We perform joint generation and learning using pretrained models that has dual benefit of data generation and improving model representations, and has been shown to be effective for anomaly detection in medical images (Li et al., 2019).

**Incorporating Phonetics in Language Modeling.** Combined modeling of phonemes and text has been a topic of recent interest and has contributed in improving robustness to ASR errors (Sundararaman et al., 2021). In code-mixed domain, Soto et al. (Soto and Hirschberg, 2019) engineered spelling and pronunciation features by calculating distance between pairs of cognate words to improve perplex-

ity of English-Spanish models. We also incorporate phonetic signals to learn robust representations.

**Sentence Evaluation Metrics.** Automated sentence evaluation metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) for comparison of unstructured sentences led to rapid innovation in NLP by facilitating ready evaluation of NLP systems against ground truth without additional human annotations. However, these metrics are unreliable for code-mixed content as they disregard widely prevalent orthographic variations. We propose a new family of metrics to address this gap.

## 3 CoMix Approach

Given a corpus of sentence pairs from $L_M$ and $L_E$, our goal is to adapt transformer models such as BART and BERT to overcome the key challenges in modeling code-mixed data. To ensure applicability to multiple downstream tasks, we focus on the pretraining phase. We assume access to POS tagging and phonetic transcription tools [2] which is true for many languages (see Section 7). Below we summarize our approach for each of the challenges.

**P1 - Divergence in POS structure of $L_E$ and $L_M$:** To enable transformer models to extrapolate from $L_E$ and $L_M$ to code-mixed data, we rely on linguistic constraints. We observe that coarse groups of POS labels of concepts are preserved across translation (see Section 7) and that code-mixed sequences often retain the POS structure of $L_M$ sequence. Assuming access to POS labels the above constraints provide token-level correspondence for parallel training sentences, which can be used to augment the transformer attention mechanism and lead to representations that facilitate accurate interleaving of $L_E$ and $L_M$ words. [Section 3.1]

**P2 - Variations in the level of code-mixing:** To accurately model variations in code-mixed data such as the mixing propensity, we propose a weakly supervised approach that combines POS constraints with a control on the code-mixing probability to generate code-mixed sequences from parallel monolingual corpora[3] for training. [Section 3.2]

**P3 - Orthographic variations:** To align similar sounding words with orthographic variations, we incorporate phonetic signal as an additional input channel. We modify the transformer architecture to include two multi-head self-attention layers, one each for text and phoneme channels. [Section 3.3]

[2]We use Stanza for POS-tagging and Refined Soundex implementation Pyphonetics for phonetic transcription.

[3]By parallel monolingual corpora we mean parallel corpora wherein each one of the parallel sentences are in pure/monolingual form of their corresponding language.

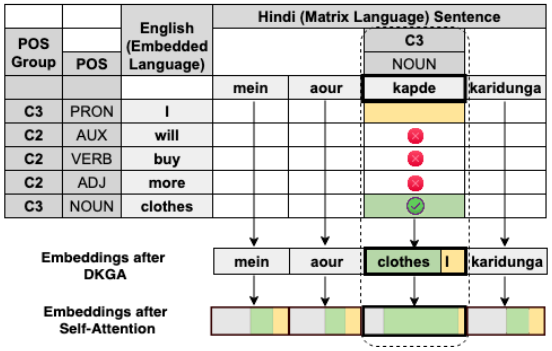



Figure 2: DKGA mechanism to force intermixing of $L_E$ tokens (*clothes, I*) into $L_M$. DKGA blocks attention on red dots and guides attention on yellow, green colored box tokens with green tick denoting right choice. C1,C2,C3 are POS groups defined in Appendix A.1.

### 3.1 Domain Knowledge Guided Attention

Attention (Vaswani et al., 2017) is an essential mechanism of transformer architecture that converts an input sequence into a latent encoding using representational vectors formed from the input, i.e., queries, keys and values to determine the importance of each portion of input while decoding. Let $X$ and $Z$ denote the sequence of input tokens and the associated representations. Further, let $Q$, $K$, $V$ denote the sequences of query, key and value vectors derived from appropriate projections of $Z$. In this case, attention is typically defined in terms of the scaled dot-product of $Q$ and $K$.

To incorporate domain knowledge, we propose augmenting attention with an additional independent term $f^{\text{DKGA}}(X)$ defined on the input:

$$Attention(Q, K, V) = softmax(\frac{QK^T + f^{\text{DKGA}}(X)}{\sqrt{d_k}})V, \quad (1)$$

where $d_k$ is the dimension of query and key vectors. While the notion of DKGA is general, to aid with code-mixing, we focus on linguistic constraints. We construct three groups of POS labels (see A.1) that are preserved during translation (see 7). Let $X$ denote the concatenation of the parallel monolingual sentences, i.e., $X = X_M \| X_E$, where $X_M$ and $X_E$ are sentences in $L_M$ and $L_E$ respectively. Let $\text{POS}^{\text{GP}}(x)$ denote the group of the POS label of a token $x$. The linguistic constraints require that aligned token pairs from $X_M$ and $X_E$ belong to the same POS label group. Hence, for matrix tokens, we restrict attention to compatible embedded

words by choosing $f^{\text{DKGA}}(X) = [f_{ij}^{\text{DKGA}}]$, where

$$f_{ij}^{\text{DKGA}} = \begin{cases} 0 & \text{if } \text{POS}^{\text{GP}}(x_i) = \text{POS}^{\text{GP}}(x_j) \\ & \text{and } x_i \in X_M, x_j \in X_E \\ & \text{or } i = j \text{ for } x_i \in X_E \\ -\infty & \text{otherwise.} \end{cases} \tag{2}$$

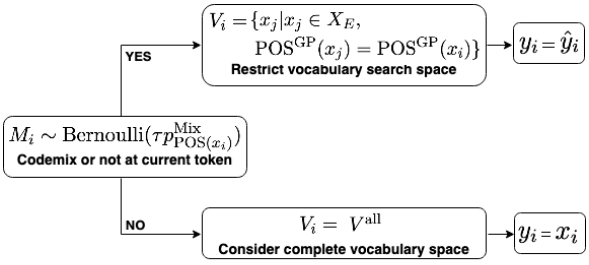
Note that the above asymmetric choice is motivated by the fact that code-mixed sentences retain the POS structure of $L_M$. Fig 2 shows how tokens from $X_E$ are selected using the above strategy which coupled with self-attention ensures learning representations that facilitate better intermixing of $L_E$ tokens into $L_M$ structure. See A.4.9 for example. Instead of a hard constraint on POS-label preservation, the DKGA function can also be modified to incorporate soft transition probabilities of POS labels during an $L_M$ to $L_E$ translation, which could be learned from parallel sentence pairs with token level alignment. We can also extend DKGA to include other sources for attention guidance, e.g., domain ontologies, word-alignment and also for cross-attention.

**Pretraining with DKGA.** We modify all self-attention blocks in the encoder with DKGA and pretrain CoMixBERT with masked language modeling (MLM) objective (Devlin et al., 2018) and CoMixBART with denoising objective (text infilling with span=1). We mask the tokens of $X_M$ for which we want DKGA to guide attention to embedded words (e.g., in Fig 2, "kapde" will be masked).

### 3.2 Weakly Supervised Generation (WSG)

Lack of large code-mixed corpora poses a big challenge for code-mixed modeling. Hence, to facilitate direct training and allow control over desired properties such as the level of code mixing, we propose a weakly supervised mechanism for code-mixed generation using any transformer-based encoder-decoder model. The key idea is to nudge a pretrained multilingual model to code-mix by restricting the search space in the autoregressive step to a small suitable set of $L_E$ tokens exploiting the the fact that tokens with similar meaning and POS labels in $L_E$ are likely to replace the $L_M$ token.

Fig 3 shows the generative mechanism (Equation 3 shows the corresponding equations). At each auto-regressive step, we first determine the choice to code-mix, denoted by $M_i$, sampled based on the mixing probability $p_{\text{Mix}}$ of the POS label of the token $x_i \in X_M$ and an overall code-mixing




where $\hat{y}_i = \text{argmax}_{y \in V_i} P(y | y_1, y_2, ..., y_{i-1}, X_M)$

Figure 3: Flow chart for WSG generative mechanism.

level $\tau$. The vocabulary search space denoted by $V_i$ is chosen as the POS-compatible words (same POS group as that of $x_i$) from $X_E$ for code-mixing, and the entire vocabulary $V^{all}$ otherwise. The next token is generated via greedy search with teacher forcing. In case of code-mixing, the target $y_i$ is set to the predicted value $\hat{y}_i$ and $x_i$ otherwise. We train the model with negative log-likelihood loss using $X_M$ as the input, $Y = [y_i]_{i=1}^N$ as the target, $\hat{Y} = [\hat{y}_i]_{i=1}^N$ as the prediction. Due to the self-dependency in WSG, the efficacy depends on whether the underlying model can correctly order the tokens in $V_i$, which is a reasonable expectation from SOTA pretrained multilingual models.

$$\hat{y}_i = \text{argmax}_{y \in V_i} P(y | y_1, y_2, ..., y_{i-1}, X_M),$$
$$y_i = \begin{cases} \hat{y}_i & \text{if } M_i = 1 \\ x_i & \text{if } M_i = 0 \end{cases},$$
$$V_i = \begin{cases} \{x_j | x_j \in X_E, \\ \text{POS}^{\text{GP}}(x_j) = \text{POS}^{\text{GP}}(x_i)\} & \text{if } M_i = 1 \\ V^{\text{all}} & \text{if } M_i = 0 \end{cases},$$
$$M_i \sim \text{Bernoulli}(\tau p^{\text{Mix}}_{\text{POS}(x_i)}). \tag{3}$$

In our experiments, we set $\tau = 1$ and $p^{Mix}$ to 1 for POS groups $\{NOUN, PROPN, ADJ, ADV, VERB\}$ where code-mixing is frequent and 0 for the rest but can learn it from a small code-mixed corpus in future. The proposed WSG mechanism can also be applied to encoder-only models such as BERT by considering a similar restriction of the vocabulary set $V_i$ at the last layer.

### 3.3 Mixing Phonetic Signal

Given a text sequence $X$, let $X^{\text{Ph}}$ denote the corresponding phonetic sequence. To incorporate both the signals, we replace the multi-head self attention layer in the transformer encoder layer with two multi-head self attention layers, one each for

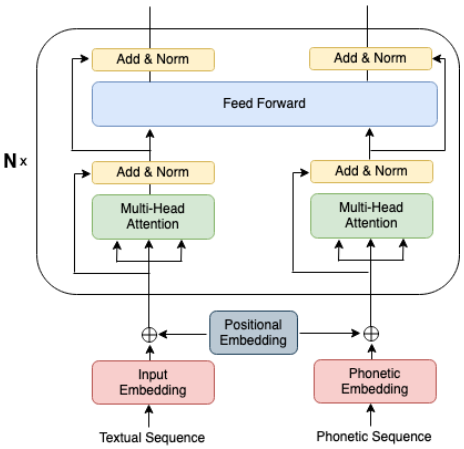



Figure 4: CoMix Transformer Encoder. Positional embeddings and feed-forward expansion weights are shared between the phonetic and textual channels.

*text* and *phoneme* channel. Text sequence shares feed-forward layers with the phonetic sequence as shown in Fig 4 since we want phonetic representations to be in the same space as text representations . To keep the number of parameters in check, we add phonetics part of the encoder to only alternate encoder layers. Our decoder uses the concatenated sequence of contextual embeddings from $X$ and $X^{\text{Ph}}$ as keys and values for cross attention.

**Pretraining with Phonetics.** We pretrain CoMixBERT for phonetics with MLM objective (as in BERT) and CoMixBART with denoising objective (text infilling with span length 1).

## 4 Phonetic Sentence Comparison Metrics

Lack of standardized transliteration rules is a key challenge not only for modeling but also for evaluating code-mixed or multi-lingual NLG tasks, e.g., English to Hinglish translation Human annotators employ orthographic variations and are also inconsistent in the use of punctuation and upper-lower casing as shown in Fig 1. Most NLG evaluation metrics such as BLEU, do not account for these variations, which leads to pessimistic and inaccurate estimates of the performance of NLG systems. To address this gap, we propose a new family of metrics based on the phonetic representation. Let $s(\cdot,\cdot)$ be any metric such as BLEU and ROUGE (Banerjee and Lavie, 2005) that facilitates comparison of a word sequence against a reference one. Given a pair of sentences $(X, Y)$, we define the phonetic metric as $Pho\text{-}s(X, Y) = s(X^{\text{Ph}}, Y^{\text{Ph}})$, where $X^{\text{Ph}}, Y^{\text{Ph}}$ are the phonetic sequences. In this paper, we limit our focus to PhoBLEU and present observations in Sec 6.6.

| | Baseline | Compared against SOTA | Metrics |
|---|---|---|---|
| Named Entity Recognition | Indic BERT | XLM-R Large | Weighted F1-Score |
| Classification | | - | Micro F1-Score |
| Machine Translation | Indic BART | mT5 | BLEU, PhoBLEU |
| Abstractive Summarization | | PEGASUS, BART | BLEU, PhoBLEU, ROUGE |

Table 1: Baselines, metrics and SOTA models.

## 5 Experiments

### 5.1 Downstream Tasks, Baselines and Metrics

Table 1 lists the four downstream tasks, baselines, SOTA models and metrics used in the evaluation[4]. For translation on HooD dataset (Sec 5.2), we also include Echo baseline which just passes the input sequence as output and helps measure the contribution of input-output overlap to the final performance.

Previous studies (Dabre et al., 2021) (Kakwani et al., 2020) indicate that IndicBART (Dabre et al., 2021) and IndicBERT (Kakwani et al., 2020) are competitive relative to mBART and mBERT respectively on Indic languages. Further, since we initialize our models CoMixBART and CoMixBERT with weights from IndicBART and IndicBERT, we consider these as strong baselines for our evaluation of generative and classification tasks respectively.

### 5.2 Datasets for Downstream Tasks

We evaluate on LINCE Eng-Hinglish dataset for translation (Chen et al., 2022), SemEval-2020 Hinglish Sentimix dataset for sequence classification (Patwa et al., 2020), GupShup Hinglish chats to summaries (GupShup H2H) dataset (Mehnaz et al., 2021) for summarization and LINCE Hinglish (Singh et al., 2018) dataset for the NER task. Table 9 in Appendix A.3.2 lists data statistics.

**Hinglish Out-of-Domain Translation Dataset (HooD).** We introduce two out-of-domain translation test-sets for Hinglish. Of these, the first one from shopping domain was prepared by in-house human experts who translated English sentences generated by humans and models like GPT3, following the guidelines in Appendix A.3.1. The second test set was prepared from GupShup corpus (Mehnaz et al., 2021) from parallel English-Hinglish summaries of conversations created by linguists (Gliwa et al., 2019). These datasets can help

[4]Formulations of all tasks are in A.2.

| | # datapoints | Avg. # of all tokens | | Avg. # of tokens in Target | | Code-Mixing |
|---|---|---|---|---|---|---|
| | | Source | Target | English | Hindi | Index (CMI) (Das and Gambäck, 2014) |
| **HooD Shopping** | 1050 | 16.63 | 17.76 | 3.35 | 14.41 | 19.2 |
| **HooD Open Domain** | 6831 | 20.29 | 22.46 | 6.16 | 16.29 | 28.96 |

Table 2: Data statistics for Hinglish Out-of-Domain (HooD) dataset.

assess the zero-shot transfer capabilities of models from movies to shopping and open-domain for models trained on LINCE English-Hinglish dataset. Table 2 shows statistics of the HooD dataset.

### 5.3 Experimental Setup

#### 5.3.1 Pretraining

**Initialization**. Training large transformer models from scratch requires thousands of GPU hours, which can be prohibitive. To ensure broader accessibility and best utilise existing models, we initialize CoMixBART decoder and encoder's non-phonetic weights (NPW) from IndicBART and CoMixBERT's NPW from IndicBERT. These are pretrained using Samanantar English-Hindi parallel corpus (Ramesh et al., 2021).

**CoMixBART**. We pretrain CoMixBART with DKGA on 1M sentences from Samanantar for 36k steps ($\sim$ 2 epochs) on three 24GB GPUs with batch size of 2816 tokens, linear learning rate warmup and decay with 16k warmup steps. We use Adam optimizer with max learning rate of 1e-3, label smoothing of 0.1, dropout of 0.1 and token masking probability of 0.4. For WSG, we pretrain the DKGA model for additional 2k steps with the same setup except label smoothening and masking probability of 0. Learning curve for pretraining with DKGA and WSG is shown in Appendix A.4.2. Since pretraining CoMixBART for phonetics from scratch is computationally prohibitive because of its size, we devise a way to obtain reasonable weights for downstream training. We initialize embeddings of phonetic tokens with the mean of embeddings of the text tokens that map to it. We also initialize phonetic self-attention layer parameters with the same weights as that of corresponding text channel's self-attention layer.

**CoMixBERT**. We pretrain CoMixBERT with DKGA, WSG and Phonetics on 100k sentences from Samanantar on six 32GB GPUs with batch size of 20 per GPU, starting with a learning rate of 5e-5, linear learning rate warmup and AdamW optimizer. We pretrain DKGA and WSG for 1k steps and Phonetics for 3k steps. We are able to pretrain CoMixBERT with Phonetics because it has 7x less parameters than CoMixBART.

#### 5.3.2 Downstream Fine-tuning

**CoMixBART.** The pretrained model is fine-tuned for downstream tasks in two stages. First, we attach a custom task-specific head to the decoder and train its weights, CoMixBART's NPW (encoder and decoder) for 5k steps on three 24GB GPUs with batch size of 2048 tokens, linear learning rate warmup, and decay with 2k warmup steps and max. learning rate of 5e-4 using Adam optimizer. In the second stage, phonetic weights of CoMixTransformer encoder are initialized as per section 5.3.1. Then, in downstream training of complete model, the weights from previous step are optimised with smaller learning rate than CoMixBART encoder's phonetic weights for additional 5k steps. We use beam search (size 4) for decoding. We train baseline IndicBART model for all tasks using YANMTT (Dabre, 2022), as prescribed in IndicBART repository (IndicBART, 2022) with the same setup as CoMix models. In all cases, we pick the model with best validation score after 5k training steps.

**CoMixBERT.** We attach a custom task-specific head to the model and train using standard fine-tuning procedure. For NER, we also have CRF layer attached after all models including baseline. Since its possible to combine encoder only models without sequential training, we report ensemble results obtained by averaging logits for DKGA+WSG and DKGA+WSG+Phonetic variants as they were better than sequential training. We use grid search to find the right set of hyperparameters for all models including baseline and pick the model with best validation score. We custom-build CoMixBART and CoMixBERT implementation using transformers (Wolf et al., 2020), YANMTT (Dabre, 2022), and PyTorch (Paszke et al., 2019).

## 6 Results and Analysis

### 6.1 Machine Translation

Table 5 shows the results for the LINCE Leaderboard English-Hinglish translation task. CoMix

[5]We show punctuation-less metrics for Echo baseline for HooD in brackets to correct for the inconsistent punctuation.

| | Movies to Shopping domain transfer | | | | | Movies to open domain transfer | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Echo | Indic BART | CoMix | | | Echo | Indic BART | CoMix | | |
| | | | DKGA | DKGA+ WSG | DKGA+WSG Phonetics | | | DKGA | DKGA+ WSG | DKGA+WSG Phonetics |
| **BLEU** | 9.88 (6.49) | 10.37 | 11.95 | 11.95 | **12.15** | 13.23 (10.35) | 16.36 | 17.16 | 17.53 | **18.67** |
| **BLEU$_{uncase}$** | 11.72 (6.79) | 11.82 | 13.47 | 13.32 | **13.57** | 14.07 (10.48) | 17.11 | 18.6 | 18.91 | **19.98** |
| **PhoBLEU** | 7.59 (7.59) | 12.75 | 14.97 | 14.88 | **14.97** | 11.99 (11.99) | 19.04 | 20.78 | 21.16 | **22.13** |

Table 3: Metrics for models trained on LINCE English-Hinglish dataset and tested on HooD dataset.[5]

| | Indic BART | CoMix | | | |
|---|---|---|---|---|---|
| | | DKGA | DKGA +WSG | DKGA +Phonetics | DKGA+WSG +Phonetics |
| **BLEU** | 11.86 | 13.43 | 13.25 | **13.88** | 13.85 |
| **BLEU$_{uncase}$** | 13.98 | 15.65 | 15.66 | **16.35** | 16.34 |
| **PhoBLEU** | 17.38 | 18.63 | 18.43 | 19.10 | **19.13** |

Table 4: Val set results on LINCE Eng-Hinglish dataset.

| | Indic BART | m BART | mT5+ CMDR | CoMix | | |
|---|---|---|---|---|---|---|
| | | | | DKGA+ Phonetics | DKGA +WSG | DKGA+WSG +Phonetics |
| **BLEU** | 11.20 | 11 | 12.67 | **12.98** | 12.41 | 12.51 |
| **#params** | 244M | 610M | 580M | 273M | 244M | 273M |

Table 5: LINCE Leaderboard scores on English-Hinglish translation test set.

| | Baseline | CoMix | | | | |
|---|---|---|---|---|---|---|
| | Indic BERT | Phonetics | DKGA | WSG | DKGA +WSG | DKGA +WSG+ Phonetics |
| **NER** | 80.65 | 82.16 | 81.42 | 81.78 | 82.4 | **83.07** |
| **CLS** | 64.51 | 65.71 | 64.87 | 66.37 | 67.47 | **67.61** |

Table 6: Weighted and Micro F1-score on test sets of NER and Classification (CLS) tasks. For NER, Phonetics model beats previous best XLM-R large model (Winata et al., 2021) by 1.46 points despite being 10x smaller. DKGA+WSG+Phonetics beats XLM-R large model by 2.37 points. Val set results in Appendix A.4.8

achieves the new SOTA result with 12.98 BLEU points beating previous best mT5 based model that is more than double the size. Validation set scores in Table 4 show that CoMix beats IndicBART by over 2 BLEU and 1.7 PhoBLEU points besides yielding faster convergence (see Appendix A.4.3). To test the generalization capabilities, we also evaluate the above models on out-of-domain HooD data. Table 3 shows the results with CoMix improving over IndicBART on both the HooD datasets in terms of both BLEU and PhoBLEU metrics, pointing towards better generalization capabilities of CoMix. HooD Open-Domain dataset has higher Code-Mixing Index (CMI) (shown in section 5.2) than HooD Shopping dataset, which is where DKGA+WSG model improves over DKGA model owing to its pretraining procedure which encourages code-mixing (see Appendix A.4.7). To reduce overfitting for phonetic weights on the downstream dataset, here we train DKGA+Phonetic model on actual training data and 40k unlabelled English sentences and DKGA model's predictions. Fig 12 in Appendix A.4.4 compares few sample generated translations from IndicBART and CoMix.

### 6.2 Named Entity Recognition

Table 6 shows weighted F1-score from LINCE leaderboard for Hinglish NER task. All the CoMixBERT components and their combinations beat baseline IndicBERT by 0.77-2.42 points and SOTA XLM-R large model, which has 10x more parameters, by 0.72-2.37 points. Since combinations yield upto 1.65 points boost, it is likely they capture different facets of code-mixing.

### 6.3 Sequence Classification

Table 6 shows micro-F1 score for Hinglish sentiment classification task where individual CoMix components beat IndicBERT model by 0.36-1.86 points and DKGA+WSG+Phonetics model beats it by over 3 points. Similar to the mBERT training in (Patwa et al., 2020), we train our model in a minimalistic fashion without any data-augmentation or weighted adversarial loss or token ids that can improve performance. Hence, we do not compare our results against other solutions in Semeval-2020 task and only compare against mBERT and IndicBERT.

### 6.4 Abstractive Summarization

On Gupshup H2H summarization dataset, CoMix beats IndicBART on all metrics (BLEU, PhoBLEU, R1, R2 and RL) by margin of 0.8 to 2 points as shown in Table 7. CoMix even beats previously published best BLEU results obtained from PEGASUS model (Zhang et al., 2019) but is worse on R1 and R2 metrics. CoMix is worse on recall-based metrics (R1, R2) and better on precision based metrics (BLEU) than PEGASUS and BART likely because of their ability to recall English-based words in the Hinglish summaries as they were pretrained only on English and the Gupshup dataset has been adapted from English conversa-

| | IndicBART | PEGASUS | BART | CoMix | | | |
|---|---|---|---|---|---|---|---|
| | | | | DKGA | DKGA +WSG | DKGA +Phonetics | DKGA+WSG +Phonetics |
| **BLEU** | 5.9 | 6.16 | 5.96 | 6.4 | 6.25 | **6.72** | 6.09 |
| **R1,R2,RL** | 30.73, 9.35, 24.74 | 35.69, 11.01, - | **36.28, 11.45, -** | 32.39, 10.11, 25.87 | 32.18, 10.13, 25.73 | 32.73, 10.12, 26.02 | 31.72, 9.8, 25.37 |
| **BLEU_uncase** | 6.15 | - | - | 6.55 | 6.42 | **6.9** | 6.35 |
| **PhoBLEU** | 6.6 | - | - | 7.33 | 7.01 | **7.64** | 7.18 |

Table 7: Results on Gupshup H2H test set where "-" indicates metrics not reported in prior work.

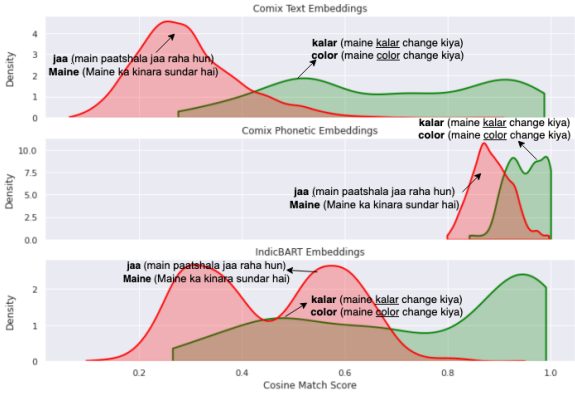



Figure 5: Cosine similarity score distribution of contextual embeddings. Red is for dissimilar words, Green is for similar words. Comix is able to separate similar and dissimilar words better than IndicBART.

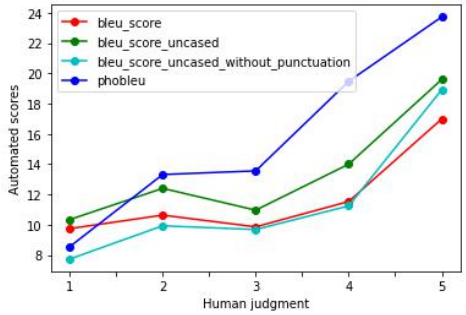



Figure 6: Mean automated scores corresponding to human rating levels.

| | Pearson | | Spearman | |
|---|---|---|---|---|
| | Correlation | P-value | Correlation | P-value |
| **BLEU** | 0.178 | 0.021 | 0.176 | 0.023 |
| **BLEU**$_{uncase}$ | 0.194 | 0.012 | 0.197 | 0.01 |
| **BLEU**$_{uncase,nopunct}$ | 0.23 | 0.002 | 0.257 | 0 |
| **PhoBLEU** | **0.333** | 0 | **0.359** | 0 |

Table 8: Correlation between human ratings and automated metrics for generative code-mixed text.

tional summarization corpus due to which it contains a lot of English named entities and words. We believe CoMixBART performance can further improve if we do pretraining with phonetics in future.

### 6.5 Qualitative Analysis

We examine how well the models in Section 6.1 separate 3655 pairs of words (178 similar, 3477 dissimilar) from 20 sentences in Appendix A.4.6. Figure 5 shows the distribution of cosine similarity of contextual embeddings (phonetic and textual for CoMix, textual for IndicBART) for similar (green) and dissimilar (red) pairs . We note that CoMix text embeddings separate the similar and dissimilar pairs better relative to IndicBART. Note that the scores for phonetic embeddings are on the higher side most likely due to initialisation choice (mean of all text tokens mapped to a phonetic token) and the smaller (0.25x of text) vocab size for phonetics.

### 6.6 Efficacy of PhoBLEU on code-mixed data

On English-Hinglish translation for the LINCE dataset, we observe that annotations from human experts fluent in both Hindi and English achieve a BLEU score of 10.43 BLEU, which is lower than most MT models. Further analysis revealed that BLEU is unable to account for valid variations in spellings, pronouns, and language switching ($L_E$ vs. $L_M$) as shown in Fig 1. To address these gaps, we consider PhoBLEU [as defined in Section 4] and evaluate its correlation with human judgements. We randomly selected 200 English-Hinglish sentence pairs and their system-generated translations to be rated by professional on a scale of 1 to 5, with 1 as poor and 5 as perfect. Completeness (no information lost) and fluency (grammatical correctness) were considered as the rating criteria. Results in Table 8 and Figure 6 show that PhoBLEU is significantly more correlated with human judgement and that its distribution is better aligned with human ratings than other BLEU variants.

## 7 Extensibility of CoMix

The proposed ideas of domain-knowledge guided attention, weak supervision, and phonetic representations are not specific to Hinglish and readily generalize to any language pair where we have a parallel corpus of embedded and matrix language content and tools for POS tagging and phonetic transcription. Below we discuss these requirements

along with other assumptions on the POS structure that permits extensions of our methodology to most common code-mixed language pairs.

**Assumption 1: Availability of parallel corpora.** Most common code-mixed languages happen to include English among the language pair which is typically transcribed in Latin script that permits easy phonetic transcription through tools such as Pyphonetics. Currently, there also exist multiple large parallel corpora (e.g. Flores, CCMatrix, Samanantar) where sentences in English are paired with that of multiple other languages. There are also many ongoing initiatives for creating such parallel corpora even for low-resource languages. Hence, the requirement of a large parallel corpus of matrix and embedded language content is satisfied by most common code-mixed pairs.

**Assumption 2: Availability of pretrained multilingual models.** With the proliferation of massively multilingual foundational models (e.g., mBART (50 languages), mBERT (104 languages), T5 (101 languages)) including advances in synthetic data augmentation, our assumption on the availability of pretrained LLMs or datasets to pretrain those models is also a reasonable one. We choose to work with IndicBART and IndicBERT which support 11 and 12 Indic languages respectively because they provide stronger baselines for Indic Languages and are faster to experiment with because of their smaller size, but the proposed ideas can be readily applied with *any* pre-trained transformer model.

**Assumption 3: Languages of $L_M$ and $L_E$ share the same POS set and access to POS tagging utilities.** (Petrov et al., 2012) proposed Universal POS tagset comprising 12 categories that exist across languages and had developed a mapping from 25 language-specific tagsets to this universal set. They demonstrated empirically that the universal POS categories generalize well across language boundaries and led to a open community initiative by universaldependencies.org on creating Universal POS tags (Nivre et al., 2020) for 90 languages. In our work, we use these universal POS tags to build three coarse groups (nouns-pronouns, adjectives-verbs-adverbs, rest) of POS tags (see Fig 7). Note that even though we utilize POS-tagging, the structural constraints are imposed with respect to these three coarse groups. Fig 7 in A.1 lists the POS tags from the universal POS tags website, which we use in our work. Further, Stanza provides Universal POS-tagging utilities for around 66 languages.

**Assumption 4: Equivalent word pair from $L_M$ and $L_E$ share the same coarse POS group.** We assume that equivalent words in an $L_M$ and $L_E$ pair share the same coarse POS group (from Fig 7 ) and not necessarily the same POS tag. A small-scale empirical analysis of 50 Hindi-English-Hinglish sentences from the HooD dataset (Sec 5.2) indicates this assumption is true in 88.6% of the cases. POS-tags provide complementary (weak) supervision for intermixing (in DKGA) and generation (in WSG) in addition to word semantics already captured in embeddings. Further, even though our current guiding function $f^{DKGA}$ assumes a hard constraint on the word pairs to be in the same coarse POS group, our methodology is general and can be extended to the case where the two languages have different POS tag sets. In particular, given empirical probabilities that a matrix token with POS tag $A$ maps to an embedded token with POS tag $B$ for all possible pairs of POS tags $(A, B)$, we can define the guiding function's value $f_{ij}^{DKGA}$ associated with matrix token $x_i$ and embedded token $x_j$ as the log of the empirical transition probability of the POS tags of the matrix token $x_i$ and embedded token $x_j$. The current choice is the special case where transition probability is uniform for all POS tag pairs within a coarse group and 0 for the rest.

## 8 Conclusion

We presented CoMix, a pretraining approach for code-mixed data that combines (a) domain knowledge guided attention (DKGA), (b) weakly supervised code-mixed generation based on POS-structure constraints, and (c) transformer encoder modification to include phonetic signal. We showed that CoMix yields improvements across multiple code-mixed tasks, achieving new SOTA result for Eng-Hinglish translation and Hinglish NER on LINCE Leaderboard with superior performance on out-of-domain translation. Our approach is applicable to code-mixing with all languages where POS tagging and phonetic transcription is possible. Motivated by gaps in current NLG evaluation metrics for code-mixed data, we proposed a new family of metrics based on phonetic representation and show that PhoBLEU is better correlated with human judgement than BLEU on Hinglish. In future, we plan to extend the applicability of DKGA and WSG to other settings that can benefit from domain knowledge, and explore new metrics for code-mixed NLG with a large scale evaluation.

## Limitations

Our CoMix approach assumes availability of parallel bilingual (embedded and matrix language) corpora and mature tools for POS tagging and phonetic transcription for both the embedded and matrix languages which does not hold true for every language. But these assumptions are reasonable for a large number of languages as shown in Appendix 7. Second, our current choice of guiding function for attention $f^{\text{DKGA}}$ and mixing probability $p^{\text{Mix}}$ are based on limited knowledge of the linguistic structure specific to English and Indic languages, and might need to be adapted for other language families. Additionally, as discussed in Section 4, due to multiple variations in code-mixed generation, current automated metrics that compare system generated text with reference text do not provide a true reflection of a system's ability to generate code-mixed text. Lastly, as with large language models, our CoMix models are also vulnerable to biases inherent in the training corpus.

## Ethics Statement

Our research motivation is to address the inequities in language resources and AI systems for multilingual societies such as India. The primary contribution of our work is a new modeling approach CoMix, which is especially designed to leverage existing pretrained models with moderate computation so that it is accessible to a wider community and does not create an adverse environmental impact. We also created two new Hinglish datasets for out-of-domain evaluation (HooD), which we described in detail in Section 5.2. There are no privacy or intellectual property rights associated with either of these datasets. We will open-source HooD, our models and code in future post organizational approval. Human translations and evaluations reported in the paper have been done by professional annotation teams and are reflective of typical performance. Similar to other large language models, our CoMix model also encodes biases in the original training corpus and the domain constraints used as supervision. While the performance might be acceptable for natural language understanding, it is important to have guardrails while using the models directly for natural language generation.

## References


Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A centralized benchmark for linguistic code-switching evaluation. *CoRR*, abs/2005.04322.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Anshul Bawa, Pranav Khadpe, Pratik Joshi, Kalika Bali, and Monojit Choudhury. 2020. Do multilingual users prefer chat-bots that code-mix? Let's nudge and find out! *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW1).

Shuguang Chen, Gustavo Aguilar, Anirudh Srinivasan, Mona T. Diab, and Thamar Solorio. 2022. Calcs 2021 shared task: Machine translation for code-switched data. *ArXiv*, abs/2202.09625.

Monojit Choudhury, Anirudh Srinivasan, and Sandipan Dandapat. 2019. Processing and understanding mixed language data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): Tutorial Abstracts*, Hong Kong, China. Association for Computational Linguistics.

Raj Dabre. 2022. YANMTT library. `https://github.com/prajdabre/yanmtt`. [Online; accessed 29-May-2022].

Raj Dabre, Himani Shrotriya, Anoop Kunchukuttan, Ratish Puduppully, Mitesh M. Khapra, and Pratyush Kumar. 2021. IndicBART: A pre-trained model for natural language generation of Indic languages. *ArXiv*, abs/2109.02903.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed Indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387, Goa, India. NLP Association of India.

Universal Dependencies. 2014. Universal POS tags. `https://universaldependencies.org/u/pos/all.html`. [Online; accessed 29-May-2022].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong

Kong, China. Association for Computational Linguistics.

GTS. 2019. Hinglish – the biggest language you've never heard of with 350 million speakers. https://blog.gts-translation.com/2019/06/12/hinglish-the-biggest-language-youve-never-heard-of-with-350-million-speakers/. [Online; accessed 29-May-2022].

IndicBART. 2022. IndicBART GitHub Repo. https://github.com/AI4Bharat/indic-bart. [Online; accessed 29-May-2022].

Ganesh Jawahar, El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2021. Exploring text-to-text transformers for English to Hinglish machine translation with synthetic code-mixing. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 36–46, Online. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

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

L Li, M Xu, X Wang, L Jiang, and H Liu. 2019. Attention based glaucoma detection: A large-scale database and CNN model. In *CVPR*, pages 571–580.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Laiba Mehnaz, Debanjan Mahata, Rakesh Gosangi, Uma Sushmitha Gunturi, Riya Jain, Gauri Gupta, Amardeep Kumar, Isabelle G. Lee, Anish Acharya, and Rajiv Ratn Shah. 2021. GupShup: Summarizing open-domain code-switched conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6177–6192, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. SemEval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 774–790, Barcelona (online). International Committee for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Shana Poplack. 1980. Sometimes I'll start a sentence in Spanish Y TERMINO EN ESPAÑOL: toward a typology of code-switching. 18(7-8):581–618.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2021. Samanantar: The largest publicly available parallel corpora collection for 11 Indic languages.

Mohd Sanad Zaki Rizvi, Anirudh Srinivasan, Tanuja Ganu, Monojit Choudhury, and Sunayana Sitaram. 2021. Gcm: A toolkit for generating synthetic code-mixed text. In *2021 Conference of the European*

*Chapter of the Association for Computational Linguistics*, pages 205–211. Association for Computational Linguistics.

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.

Víctor Soto and Julia Hirschberg. 2019. Improving code-switched language modeling performance using cognate features. In *INTERSPEECH*.

Mukuntha Narayanan Sundararaman, Ayush Kumar, and Jithendra Vepa. 2021. Phoneme-BERT: Joint language modelling of phoneme sequence and ASR transcript. In *Interspeech*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2021. Are multilingual models effective in code-switching? *CoRR*, abs/2103.13309.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777.

# A Appendix

## A.1 Classes of POS tags

Figure 7 shows the POS tag groups used by DKGA. We built these groups using information from Universal Dependencies (Dependencies, 2014).

```
Class 1 = {'CCONJ', 'DET', 'INTJ', 'NUM', 'SCONJ'}
Class 2 = {'ADJ', 'VERB', 'AUX', 'ADP', 'ADV'}
Class 3 = {'NOUN', 'PRON', 'PROPN'}
```

Figure 7: Classes of POS Tags (POS Groups) used by guiding function in DKGA.

## A.2 Formulations of Downstream tasks

### A.2.1 Machine Translation

Given a source sequence $X = [x_1, x_2, ...x_S]$ and target sequence $Y = [y_1, y_2, ...y_T]$, autoregressive neural machine translation system learns to model the following distribution

$$P(Y|X) = \prod_{t=1}^{T} P(y_t|y_0, y_1, ..., y_{t-1}, x_1, x_2, ..., x_S) \tag{4}$$

Given a training set $D = \{\langle X^{(i)}, Y^{(i)}\rangle\}_{i=0}^{M}$ with $M$ data points, we aim to maximize $L_\theta = \sum_{i=0}^{M} \log P(Y^{(i)}|X^{(i)}; \theta)$ where $\theta$ is set of model parameters.

### A.2.2 Sequence Classification

In sequence classification task, we are given a sequence $X = [x_1, x_2, ...x_S]$ and corresponding label $y \in \{y_1, y_2, ..., y_k\}$ from fixed set of $k$ classes. Given a training set $D = \{\langle X^{(i)}, y^{(i)}\rangle\}_{i=0}^{M}$ with $M$ data points, we aim to maximize $L_\theta = \sum_{i=0}^{M} \log P(y^{(i)}|X^{(i)}; \theta)$

### A.2.3 Abstractive Summarization

Mathematical formulation for summarization is same as translation so we avoid repeating it here for brevity. In abstractive summarization, unlike translation, target sequence $Y$ is a concise summary of source sequence $X$, usually much shorter in length than $X$.

### A.2.4 Token Classification

In token classification task, we are given a sequence $X = [x_1, x_2, ...x_S]$ and corresponding label $y_s \in \{y_1, y_2, ..., y_k\} \forall s \in \{1, S\}$ for every input token, where $\{y_1, y_2, ..., y_k\}$ is the fixed set of $k$ classes. Given a training set $D = \{\langle X^{(i)}, Y^{(i)}\rangle\}_{i=0}^{M}$ with $M$ data points, we aim to maximize $L_\theta = \sum_{i=0}^{M} \log P(Y^{(i)}|X^{(i)}; \theta)$

## A.3 More details about datasets

### A.3.1 Guidelines for preparing HooD Shopping dataset

Figure 8 shows the guidelines given to human annotators for translating English sentences to Hinglish for HooD Shopping dataset.

### A.3.2 Data statistics of public datasets

Table 9 shows statistics for public datasets which we have used for downstream tasks.

Convert a English sentence to its Hinglish Translation in Roman script keeping following things in mind:

- Assume that you're **chatting with a friend** who is a native speaker of Hindi but also speaks English well.
- Inter-mix English and Hindi words **naturally** as one would in a conversation
- Use **Roman script** irrespective of whether the word being used belongs to English or Hindi
- DO NOT provide pure Hindi translations in Latin script of the English sentence [i.e do not try **hard** to translate English words which you would typically not translate in an **informal** conversation. We would expect you to retain difficult-to-translate and colloquially relevant English words as it is]
- Use Hindi grammar for translated Hinglish and use English words in between wherever its natural to do so.

Few example English sentences and their translations are:

**English** → Thank you for the feedback. I'm sorry to hear that my attitude wasn't up to your standards. I'll try to do better in the future.
**Hinglish** → Aapke **feedback** ke liye **thanks**. Yeh sunkar mujhe dukh hua ki mera **attitude** aapke **standards** ke **level** ka nahi tha. Main **future** mein **better** karne ka **try** karunga.

**English** → Thank you for your question! You can find whatever you need on Amazon.com. We have a huge selection of items, so you're sure to find what you're looking for. Thanks for shopping with us!
**Hinglish** → Aapke **question** ke liye **thanks**! Aapko jo kuch bhi chahiye sab Amazon.com par **search** kar sakte hain. Aapko jo bhi chahiye voh **for sure** mil jayega kyuki humare paas **items** ki **huge selection** hai. Humare saath **shopping** karne ke liye **thanks**!

**English** → I would be happy to help you with your book purchase on Amazon. Please let me know what book you are interested in and I will do my best to find it for you. Thank you for using Amazon Shopping!
**Hinglish** → Mujhe aapko Amazon par **book purchase** karne mein **help** karne par khushi hogi. Bataiye aap konsi **book** mein **interested** hain and usse dhundhne ke liye **I will do my best**. Amazon **Shopping** ko use karne ke liye **thanks**!

Figure 8: Guidelines given to Human annotators for translating English sentences to Hinglish.

| Dataset Name | Number of Datapoints | | |
|---|---|---|---|
| | Train | Dev | Test |
| LINCE English - Hinglish Translation Dataset | 8060 | 942 | 960 |
| SemEval 2020 Task-9 Hinglish Sentimix | 14000 | 3000 | 3000 |
| Gupshup H2H | 5831 | 500 | 500 |
| LINCE Hinglish NER | 1243 | 314 | 522 |

Table 9: Statistics of public datasets we have used for code-mixed Machine Translation, Sequence Classification, Abstractive Summarization and NER.

## A.4 Additional experimental details and results

### A.4.1 Details about tokenization

For Phonetics data, we train our own sub-word tokenizer using sentencepiece[6]. For text data we use pretrained IndicBART's tokenizer for CoMixBART and IndicBERT's tokenizer for CoMixBERT. We consider sub-words POS to be same as the POS of the word from which sub-words have been created.

### A.4.2 More details on pretraining with DKGA and WSG

Figure 9 shows the learning curve for pretraining CoMixBART. As you can see from the curve, loss stabilizes after 25k steps and does not change much. Figure 10 shows the learning curve for pretraining ComixBART with WSG.

Table 10 shows few sample inputs which went into the model during WSG training and corresponding targets constructed by the model. These generated sentences can be used for data-augmentation which we plan to explore in the future.

[6]https://github.com/google/sentencepiece

Figure 9: Change in negative log-likelihood loss with training steps for Pretraining CoMixBART with DKGA.

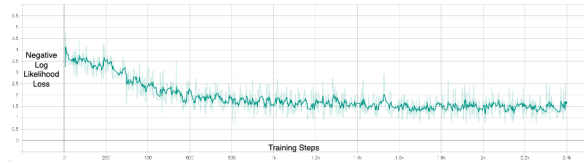



Figure 10: Change in negative log-likelihood loss with training steps for Pretraining CoMixBART with WSG.

### A.4.3 Convergence for IndicBART vs CoMix on LINCE Leaderboard translation task

Figure 11 shows the convergence speed for IndicBART and CoMix models for LINCE English-Hinglish translation task. As you can see from the curve, CoMix is better than baseline IndicBART throughout training.

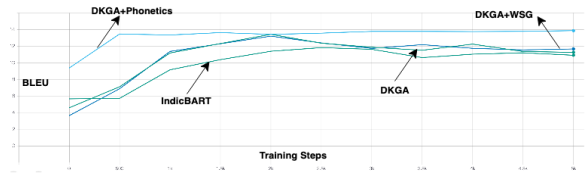



Figure 11: Change in BLEU score with training steps for IndicBART and CoMix.

### A.4.4 Qualitative Analysis on few sample predictions for Code-Mixed Translation

Figure 12 shows few example translations generated by IndicBART and CoMix and their qualitative analysis.

### A.4.5 Set of sentences for cosine similarity distribution

Figure 13 shows the 20 sentences from which every pair of word was labelled similar/dissimilar manually and then used to create Figure 5 that shows cosine similarity score distribution of contextual embeddings obtained from the encoder of CoMix and IndicBART.

| Input | Target built by WSG |
|---|---|
| hamen koi naaraazgi bhi nahin he. \|\|\|\| We dont have any complaint. | hamen koi **complaint** bhi nahin he. |
| jald hi aapako apadet milegaa. \|\|\|\| There will be an update soon. | jald hi aapako **update** milegaa. |
| vaatayan antarrashtriya puraskaar, landan ke vaatayan-euke sanghathan dwaara diya jaataa he \|\|\|\| Vatayan International Awards given by the Vatayan - UK organization in London, honours poets, writers and artists for their exemplary work in their respective fields | vaatayan **work** puraskaar, **UK** ke **artists** sanghathan dwaara **International** jaataa he |
| LIVE / jharkhand main shuruaati rujhaanon main bhaajapa 18 congress+ 37 siton par aage \|\|\|\| Congress leading 37 seats, BJP ahead in 18 | LIVE / jharkhand main **ahead** rujhaanon main bhaajapa 18 **Congress** 37 **seats** par **leading** |
| iske baad vidhayak vahaan se hate. \|\|\|\| The MLA then left the venue. | iske baad **MLA** vahaan se **left** |

Table 10: Few sample inputs which went into the model during WSG training and corresponding targets built by the model. Words in **bold** are the embedding language (English) words.

| English | Hinglish | | | Comment |
|---|---|---|---|---|
| | Reference - Human generated | IndicBART generated | Comix generated | |
| **Good, thank you for asking! I'm doing well today. How are you?** | Good, puchne ke liye thanks! Mai aaj acha hu. Aap kaise ho? | acha, thank you aapke liye! mein achchi kar raha hoon. kaise ho? | Good, thank you for asking! Main aaj achcha kar raha hu. Tum kaise ho? | IndicBART struggles with gender specific pronouns (use of achchi vs achcha, tumhara vs tumhari), whereas Comix does good job. Also notice the use of different orthographic forms across 3 generations [(Mai, mein, Main), (acha, achcha), (hu, hoon)] |
| **Certainly! What is your wife's favorite type of gift?** | Ji haan! Aapke wife ka favourite type of gift kya hain? | Certainly! tumhara wife ka favorite type ka gift kya hai? | Certainly! tumhari wife ka favorite type ka gift kya hai? | |
| **I am, thank you for asking. It was just a little scary. I had to call 911.** | I am, thanks puchne ke liye. Mai bas thoda dar rahi thi. Mujhe 911 ko call karna pada. | mein tho thank you.. voh thoda scary hein.. mein 911 ko call karunga | mai dhanyavaad, aapse poochne ke liye. yah thoda scary tha. mujhe 911 call karna pada. | Comix gets the tense right on "had to call", IndicBART doesn't. Comix doesn't differentiate well between "aapse vs mujhse, mujhe vs mera" etc |
| **I sure am! We're getting a lot of orders in and everyone is working hard to get them out on time. Thanks for your patience as we work to get everyone's orders out as soon as possible.** | I sure am! Hume bahut saare order mil rahe hain aur har koi unhen samay par nikaalane ke lie bohot mehanat kar raha hai. Aapke patience ke lie thanks kyonki hum sabke orders ko jald se jald poora karane ke lie kaam karte hain. | me sure am! We're getting a lot of orders in and everyone is working hard to get them out on time. Thanks tumhe patience as we work to get everyone's orders out as soon as possible possible | I sure am! We're getting a lot of order aa rahe hai aur sabhi kaam karne me hard ho rahe hai. Thanks for your patience as we work to get everyone's orders out as soon as possible. | Even though Comix is slightly better than IndicBART here, it certainly wasn't perfect likely because of the difficulty of the sentence and its length. |
| **I sure can! What type of clothing are you looking for?** | Mai bilkul kar sakta hua! Aap kis type ka clothing dhund rahe hai? | me sure can! konsa type ka clothing aapko dekh rahe hain? | I sure can! Tum kis type ke clothing dekh rahe ho? | Comix does great job with getting pronouns, grammar and everything else perfect. |

Figure 12: Few example translations generated by IndicBART and CoMix.

### A.4.6 CoMix vs IndicBART Cosine Similarity Distribution of Contextual Embeddings

Figure 11 shows the mean and variance of the cosine similarity distribution of 3655 word pairs constructed from the 20 sentences in Figure 13 along different subsets of positive pairs and close negative pairs. We observe that the cosine score for positive pairs based on CoMix text embeddings has a bimodal distribution with high scores for those with same language and spelling, but relative low scores when that is not the case. However, even these low scoring positive pairs are comparable or score higher than close negatives. In the case of IndicBART, we again observe a bimodal distribution for the negative pairs with high scores for pairs that have different semantics but share either the spelling or phonetic representation, which makes it difficult to separate it from the positive pairs. CoMix phonetic embeddings by itself does not seem to be very discriminatory but it is helpful in making up for the shortcomings of CoMix text embeddings for handling phonetic variations.

### A.4.7 DKGA vs WSG. Who's code-mixing more?

Since in WSG we're nudging the model to code-mix, that behaviour is visible in the generated translations by the two models as well. Figure 14 shows few randomly sampled translations generated by DKGA and DKGA+WSG models. It is visible from the translations that DKGA+WSG model is switching between matrix and embedded language more often, because of its pretraining.

### A.4.8 NER and Classification Results

Table 12 shows results on validation and test sets for NER and Sequence Classification tasks.

### A.4.9 Example DKGA attention matrix

Fig 15 shows DKGA attention matrix for an example sentence.

| ID | Source Sentences |
|---|---|
| 1 | maine kalar change kiya |
| 2 | maine color change kiya |
| 3 | maine rang badal diya |
| 4 | I changed the hue |
| 5 | I changed the colour |
| 6 | mein school jaa raha hun |
| 7 | mein patshala jaa raha hun |
| 8 | main paatshala jaa raha hun |
| 9 | mein vidyalay jaa raha hun |
| 10 | I am going to school |
| 11 | What is the main reason? |
| 12 | primary reason kya hai |
| 13 | main reazon kya hai |
| 14 | main karan kya hai |
| 15 | main vajah kya hai |
| 16 | Maine has a beautiful coast |
| 17 | Maine ka coast sundar hai |
| 18 | Maine ka kinara sundar hai |
| 19 | Maine ka coost aakarshak hai |
| 20 | Maaine ka tat sundar hai |

Figure 13: 20 sentences containing words with semantic, phonetic and orthographic similarity and differences.

| | | Comix Text | Comix Phonetics | IndicBART | Example |
|---|---|---|---|---|---|
| **All Pairs** | | (0.31,0.14) | (0.89,0.04) | (0.47,0.16) | |
| **Positive Pairs** | All | (0.68,0.20) | (0.95,0.04) | (0.73,0.23) | |
| | Same language & spelling | (0.86, 0.10) | (0.97,0.03) | (0.94, 0.04) | (mein **school** jaa raha hun, I am going to **school**) |
| | Same language & phonetics, but different spelling | (0.56,0.12) | (0.97, 0.03) | (0.64,0.13) | (Maine has a beautiful **coast**, Maine ka **coost** aakarshak hai) |
| | Same language but different spelling & phonetics | (0.51, 0.13) | (0.91, 0.03) | (0.64,0.11) | (Maine ka coost **aakarshak** hai, Maine ka kinara **sundar** hai) |
| | Different language, spelling & phonetics | (0.48, 0.09) | (0.92, 0.02) | (0.44, 0.09) | {Maine ka coost **aakarshak** hai, Maine has a **beautiful** coast} |
| **Negative Pairs** | All | (0.29,0.10) | (0.88,0.04) | (0.46,0.15) | |
| | Different language but same spelling | (0.65,0.12) | (0.99, 0.00) | (0.82 0.08) | {**main** karan kya hai, **main** paatshala jaa raha hun} |
| | Same language & phonetics, but different spelling | (0.43, 0.07) | (0.95, 0.07 | (0.67,0.10) | {**main** karan kya hai, **Maine** ka coast sundar hai} |
| | Different language and spelling, but same phonetics | (0.53,0.11) | (0.97,0.02) | (0.64,0.12) | {**maine** rang badal diya, **main** vajah kya hai} |

Table 11: Mean and Variance of the cosine similarity distribution of 3655 word pairs constructed from the 20 sentences in Figure 13 along different subsets of positive pairs and close negative pairs.

| | | IndicBERT | CoMix | | | | |
|---|---|---|---|---|---|---|---|
| | | | Phonetics | DKGA | WSG | DKGA+WSG | DKGA+WSG+Phonetics |
| **NER** | **Val** | 79.9 | 80.48 | 81.31 | 79.85 | **82.25** | 81.28 |
| | **Test** | 80.65 | 82.16 | 81.42 | 81.78 | 82.4 | **83.07** |
| **Classification** | **Val** | 59.41 | 59.57 | 58.9 | 60.14 | 60.8 | **60.9** |
| | **Test** | 64.51 | 65.71 | 64.87 | 66.37 | 67.47 | **67.61** |

Table 12: Validation and Test set results of NER and Classification set

| English | Hinglish by DKGA model | Hinglish by DKGA + WSG model |
|---|---|---|
| The department has nearly 450 employees on its rolls. | is **department** ne lagbhag 450 karmchaaree banaae hai | is **department** ne apne **rolls** par lagbhag 450 karmchaaree banaae hai. |
| Behold, this is a recompense for you, and your striving is thanked.' | haan, yeh tho aapka **recompense** hein, aur aapka **striving** dhanyavaad hein | **Behold**, ye aapke liye **recompense** hai, aur aapka **striving** bahut dhanyavaad hai. |
| The naval teams reached the site with specialist divers. | **naval team** jab specialist **divers** pahle gayi thi | **naval** tiimon ne **site** par **specialist divers** ke saath pahunchi. |
| Premier League return would lift morale, says government minister | **premier League** waapas **morale** ko hataayegi, mantri ne kaha | **Premier League return** ne **morale** ko **lift** kiya, **says government minister** |
| This was stated by a police spokesman. | Ye baat policewan ne kahi hai. | Yeh ek **police** pravaktaa ne kaha tha. |
| The college management informed the police about the incident. | **college management** ko is ghatana ke baare mein soochibaddh karaaya gaya. | **college management** ne ghatana ke baare mein **police** ko suchit kiya. |
| Cried lies before them the people of Noah and the men of Er-Rass, and Thamood, | **Cried** hai Noah aur Er-Rass ke logon ke pehle, aur Thamood, | **Cried** kiya un logon ko Noah **and the men** of Er-Rass, aur Thamood, |
| The Elantra will continue to compare against the Skoda Octavia and Honda Civic. | **The Elantra will continue to compare against the Skoda Octavia and Honda Civic.** | **Elantra Skoda Octavia** aur **Honda Civic** ka **compare** karega. |
| Four persons died on the spot, while the rest were seriously injured. | chaar log maare gaye jabaki chaar log gambhir rup se ghayal ho gaye. | chaar log maare gaye, jabaki **rest** gambhir rup se ghayal ho gaye. |

Figure 14: Comparing randomly sampled 10 translations generated by DKGA vs DKGA+WSG. We see that DKGA+WSG model switches between matrix and embedded language more often than DKGA because of how we nudge the WSG model to code-mix during pretraining.

**Step 1:** Intermix embeddings of one language into structure of another using DKGA

| | | | C3 | C2 | C3 | C2 | | C3 | C2 | C2 | C2 | C3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PRON | ADJ | NOUN | VERB | | PRON | AUX | VERB | ADJ | NOUN |
| | | | **mein** | **aour** | **kapde** | **karidunga** | [SEP] | **I** | **will** | **buy** | **more** | **clothes** |
| **C3** | PRON | **mein** | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| **C2** | ADJ | **aour** | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| **C3** | NOUN | **kapde** | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| **C2** | VERB | **karidunga** | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| | | [SEP] | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ |
| **C3** | PRON | **I** | ☑ | ☐ | ☑ | ☐ | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ |
| **C2** | AUX | **will** | ☐ | ☑ | ☐ | ☑ | ☐ | ☐ | ☑ | ☐ | ☐ | ☐ |
| **C2** | VERB | **buy** | ☐ | ☑ | ☐ | ☑ | ☐ | ☐ | ☐ | ☑ | ☐ | ☐ |
| **C2** | ADJ | **more** | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ |
| **C3** | NOUN | **clothes** | ☑ | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☑ |

**Step 2:** Normal self attention over the following sequence. For every token, Choice 1 or Choice 2 token(s) are chosen depending upon choice of codemixing. For the cases where Choice 2 token is chosen, their word embeddings are added weighted by their attention weights.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Choices for a token** | **Choice 1** | mein | aour | kapde | karidunga | [SEP] | I | will | buy | more | clothes |
| | **Choice 2** | (I, clothes) | (will, buy, more) | (I, clothes) | (will, buy, more) | [SEP] | - | - | - | - | - |

Figure 15: DKGA attention matrix and contextual embeddings construction for example sentence. Green ticks is where DKGA will guide attention. If we choose to intermix embedding language token at a position then "Choice 2" tokens will be considered else "Choice 1" tokens will be considered for constructing contextual embeddings. C1,C2,C3 are POS groups defined in Appendix A.1.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☐ A1. Did you describe the limitations of your work?
*Left blank.*

☐ A2. Did you discuss any potential risks of your work?
*Left blank.*

☐ A3. Do the abstract and introduction summarize the paper's main claims?
*Left blank.*

☐ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☐ Did you use or create scientific artifacts?

*Left blank.*

☐ B1. Did you cite the creators of artifacts you used?
*Left blank.*

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*Left blank.*

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*Left blank.*

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*Left blank.*

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*Left blank.*

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*Left blank.*

### C ☐ Did you run computational experiments?

*Left blank.*

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*Left blank.*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*Left blank.*

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*Left blank.*

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*Left blank.*

**D ☐ Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*Left blank.*

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*Left blank.*

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*Left blank.*

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*Left blank.*

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*Left blank.*