Code-Mixing on Sesame Street: Dawn of the Adversarial Polyglots

Samson Tan[§♮] **Shafiq Joty**[§‡]
[§]Salesforce Research Asia [♮]National University of Singapore [‡]Nanyang Technological University
{samson.tan,sjoty}@salesforce.com

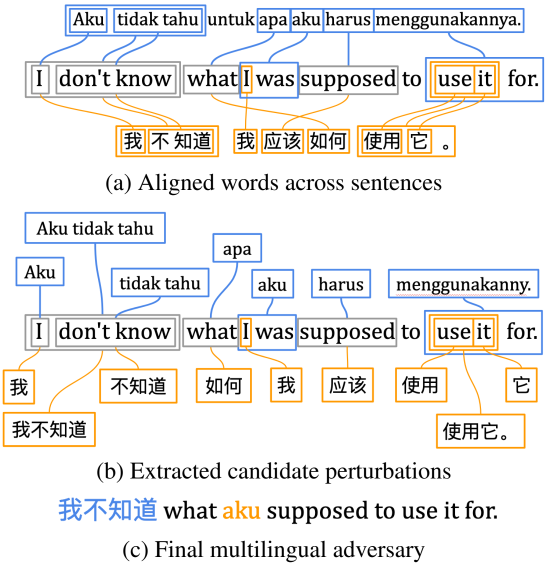
Abstract

Multilingual models have demonstrated impressive cross-lingual transfer performance. However, test sets like XNLI are monolingual at the example level. In multilingual communities, it is common for polyglots to code-mix when conversing with each other. Inspired by this phenomenon, we present two strong black-box adversarial attacks (one word-level, one phrase-level) for multilingual models that push their ability to handle code-mixed sentences to the limit. The former uses bilingual dictionaries to propose perturbations and translations of the clean example for sense disambiguation. The latter directly aligns the clean example with its translations before extracting phrases as perturbations. Our phrase-level attack has a success rate of 89.75% against XLM-R$_{\text{large}}$, bringing its average accuracy of 79.85 down to 8.18 on XNLI. Finally, we propose an efficient adversarial training scheme that trains in the same number of steps as the original model and show that it improves model accuracy.[1]

1 Introduction

The past year has seen incredible breakthroughs in cross-lingual generalization with the advent of massive multilingual models that aim to learn universal language representations (Pires et al., 2019; Wu and Dredze, 2019; Conneau et al., 2020b). These models have demonstrated impressive cross-lingual transfer abilities: simply fine-tuning them on task data from a high resource language such as English after pretraining on monolingual corpora was sufficient to manifest such abilities. This was observed even for languages with different scripts and no vocabulary overlap (K et al., 2020).

However, transferring from one language to another is insufficient for NLP systems to understand multilingual speakers in an increasingly multilingual world (Aronin and Singleton, 2008). In many multilingual societies (e.g., Singapore, Papua New Guinea, etc.), it is common for multilingual interlocutors to produce sentences by mixing words, phrases, and even grammatical structures from the languages in their repertoires (Matras and Sakel, 2007). This is known as *code-mixing* (Poplack et al., 1988), a phenomenon common in casual conversational environments such as social media and text messages.[2] Hence, it is crucial for NLP systems serving multilingual communities to be robust to code-mixing if they are to understand and establish rapport with their users (Tay, 1989; Bawa et al., 2020) or defend against adversarial polyglots.



(a) Aligned words across sentences

(b) Extracted candidate perturbations

(c) Final multilingual adversary

Figure 1: BUMBLEBEE's three key stages of adversary generation: (a) Align words in the matrix (English) and embedded sentences (top: **Indonesian**, bottom: **Chinese**); (b) Extract candidate perturbations from embedded sentences; (c) Construct final adversary by maximizing the target model's loss.

Although gold standard data (Bali et al., 2014; Patwa et al., 2020) is important for definitively evaluating code-mixed text processing ability, such datasets are expensive to collect and annotate. The

[1] Code: github.com/salesforce/adversarial-polyglots

[2] Examples of real code-mixing in Appendix A.

dizzying range of potential language combinations further compounds the immensity of such an effort.

We posit that performance on appropriately crafted adversaries could act as a lower bound of a model's ability to generalize to the distribution simulated by said adversaries, an idea akin to *worst-case analysis* (Divekar, 1984). For example, Tan et al. (2020b) showed that an NLP system that was robust to morphological adversaries was less perplexed by dialectal text exhibiting morphological variation. Likewise, if a system is robust to code-mixed adversaries constructed from some set of languages, it is reasonable to expect it to also perform better on real code-mixed text in those languages. While they may not fully model the intricacies of real code-mixing (Sridhar and Sridhar, 1980), we believe that they can be useful in the absence of appropriate evaluation data. Hence, we:

- Propose two strong black-box adversarial attacks targeting the cross-lingual generalization ability of massive multilingual representations (Fig. 1), demonstrating their effectiveness on state-of-the-art models for natural language inference and question answering. To our knowledge, these are the first two multilingual adversarial attacks.
- Propose an efficient adversarial training scheme that takes the same number of steps as standard supervised training and show that it creates more language-invariant representations, improving accuracy in the absence of lexical overlap.

2 Related Work

Multilingual classifiers. Low resource languages often lack support due to the high cost of annotating data for supervised learning. An approach to tackle this challenge is to build cross-lingual representations that only need to be trained on task data from a high resource language to perform well on another under-resourced language (Klementiev et al., 2012). Artetxe and Schwenk (2019) presented the first general purpose multilingual representation using a BiLSTM encoder. Following the success of Transformer models (Vaswani et al., 2017), recent multilingual models like mBERT (Devlin et al., 2019), Unicoder (Huang et al., 2019), and XLM-R (Conneau et al., 2020a) take the pretraining→fine-tuning paradigm into the multilingual realm by pretraining Transformer encoders on unlabeled monolingual corpora with various language modeling objectives before fine-tuning them on task data from a high-resource language such as English. This is known as cross-lingual transfer.

Code-mixed text processing. Previous research on code-mixed text processing focused on constructing formal grammars (Joshi, 1982) and token-level language identification (Bali et al., 2014; Solorio et al., 2014; Barman et al., 2014), before progressing to named entity recognition and part-of-speech tagging (Ball and Garrette, 2018; AlGhamdi and Diab, 2019; Aguilar and Solorio, 2020). Recent work explores code-mixing in higher-level tasks such as question answering and task-oriented dialogue (Chandu et al., 2019; Ahn et al., 2020). Muller et al. (2020) demonstrate mBERT's ability to transfer to an unseen dialect by exploiting its speakers' tendency to code-mix.

A key challenge of developing models that are robust to code-mixing is the availability of code-mixed datasets. Hence, Winata et al. (2019) use a pointer-generator network to generate synthetically code-mixed sentences while Pratapa et al. (2018) explore the use of parse trees for the same purpose.

Yang et al. (2020) propose to improve machine translation with "code-switching pretraining", replacing words with their translations in a similar manner to masked language modeling (Devlin et al., 2019). These word pairs are constructed from monolingual corpora using cosine similarity. Sitaram et al. (2019) provide a comprehensive survey of code-mixed language processing.

Word-level adversaries. Modified inputs aimed at disrupting a model's predictions are known as adversarial examples (Szegedy et al., 2014). In NLP, perturbations can be applied at the character, subword, word, phrase, or sentence levels.

Early word-level adversarial attacks (Ebrahimi et al., 2018; Blohm et al., 2018) made use of the target model's gradients to flip individual words to trick the model into making the wrong prediction. However, while the perturbations were adversarial for the target model, perturbed word's original semantics was often not preserved. This could result in the expected prediction changing and making the model appear more brittle than it actually is.

Later research addressed this by searching for adversarial rules (Ribeiro et al., 2018) or by constraining the candidate perturbations to the k nearest neighbors in the embedding space (Alzantot et al., 2018; Michel et al., 2019; Ren et al., 2019; Zhang et al., 2019; Li et al., 2019; Jin et al., 2020). Zang

Original	**P:** The girl that can help me is all the way across town. **H:** There is no one who can help me.
Adversary	**P:** olan girl that can help me is all the way across town. **H:** لا يوجد one who can help me.
Prediction	**Before:** Contradiction **After:** Entailment
Original	**P:** We didn't know where they were going. **H:** We didn't know where the people were traveling to.
Adversary	**P:** We didn't know where they were going. **H:** We didn't know where les gens allaient.
Prediction	**Before:** Entailment **After:** Neutral
Original	**P:** Well it got to where there's two or three aircraft arrive in a week and I didn't know where they're flying to. **H:** There are never any aircraft arriving.
Adversary	**P:** общем, дошло до mahali there's two or three aircraft arrive in a week and I didn't know where they're flying to. **H:** 从来没有 aircraft arriving.
Prediction	**Before:** Contradiction **After:** Entailment

Table 1: BUMBLEBEE adversaries found for XLM-R on XNLI (P: Premise; H: Hypothesis).

et al. (2020) take another approach by making use of a annotated sememes to disambiguate polysemous words, while Tan et al. (2020a) perturb only the words' morphology and encourage semantic preservation via a part-of-speech constraint. Other approaches make use of language models to generate candidate perturbations (Garg and Ramakrishnan, 2020; Han et al., 2020). Wallace et al. (2019) find phrases that act as universally adversarial perturbations when prepended to clean inputs. Zhang et al. (2020) provide a comprehensive survey.

Summary. Existing work on pretrained multilingual models has highlighted their impressive zero-shot cross-lingual transfer ability, though some analyses (K et al., 2020) indicate this could be a result of exploiting lexical overlaps rather than an indication of true cross-lingual understanding. Although language-agnosticity is commonly measured via cross-lingual retrieval tasks such as LAReQA (Roy et al., 2020) and similarity search (Artetxe and Schwenk, 2019), we offer a different perspective in this paper by operationalizing it as a model's ability to handle code-mixing. Existing evaluations for code-mixed text processing focus on gold annotated data, but such datasets are (relatively) expensive to compile and face similar scarcity challenges as those for low-resource languages. Existing word-/phrase-level adversarial attacks probing the limits of model robustness have largely focused on monolingual (English) inputs. In contrast, our adversarial attacks are designed to test the robustness of multilingual models to adversarial code-mixers. Finally, we propose an efficient adversarial training scheme to improve the robustness of said models to code-mixed adversaries.

3 Generating Multilingual Adversaries

Code-mixing is a phenomenon where a multilingual speaker mixes words, and even grammatical rules, from different languages in a single sentence. This is distinguished from code-switching, which occurs at the inter-sentential level (Kachru, 1978).

Extreme code-mixing. Inspired by the proliferation of real-life code-mixing and polyglots, we propose POLYGLOSS and BUMBLEBEE, two multilingual adversarial attacks that adopt the persona of an adversarial code-mixer. We focus on the lexical component of code-mixing, where some words in a sentence are substituted with their equivalents from another language in the interlocutor's repertoire. Borrowed words fall into two categories, nonce borrowing and loanwords, though distinguishing between them is beyond the scope of this work.

Since most code-mixers are bilinguals, natural code-mixed sentences tend to be constructed from two languages, with one language determining the syntax of the overall sentence (Poplack et al., 1988). However, in a world with an increasing number of multilingual societies, it is conceivable for code-mixing to occur between more than two languages (Tan, 1988). We take this idea to the extreme to test multilingual representations for their robustness to such cross-lingual lexical variation.

Problem formulation. Given a target multilingual model $\mathcal{M}$, a clean example x with the label y, and a set of embedded languages $\mathbb{L}$ from which to borrow words, we aim to generate the adversarial example x' that maximizes $\mathcal{M}$'s loss. Formally,

$$x' = \underset{x_c \in X}{\arg\max}\ \mathcal{L}(y, \mathcal{M}(x_c)), \qquad (1)$$

where $x_c \in X$ is a candidate adversary generated by perturbing x, $\mathcal{M}$ is a task-specific neural model, and $\mathcal{L}(\cdot)$ is the model's loss function.

3.1 POLYGLOSS: Word-Level Adversaries

To obtain a code-mixed adversary, we first generate candidate adversaries by substituting words in the clean example with their equivalents from another language. These substitutions/perturbations can be generated by via machine translation or mined from bilingual dictionaries. Following Myers-Scotton (1997), we will refer to the original example's language as the matrix language and the perturbation's language as the embedded language.

Next, we perform beam search on the candidates to find the adversary that maximizes the target model's loss in a black-box manner (Alg. 2 in Appendix B.1). In our implementation, we also keep track of successful adversaries and return the ones with the highest and lowest losses. The former is a stronger adversary, while the latter often has fewer perturbations. More details are in Appendix B.1.

Orthographic preservation. When the embedded language uses a different script from the matrix language, code-mixers tend to transliterate borrowed words into the same script (Abuhakema, 2013; Bali et al., 2014). This still poses a significant challenge to multilingual models (Khanuja et al., 2020). We generally preserve the embedded language's script where possible to avoid unfairly penalizing the target model since there is often no standard way of transliterating words.

Scalable sense disambiguation. Due to the polysemous nature of many words, translating the right sense is crucial to preserving the word's (and sentence's) semantics. Common word sense disambiguation methods (Agirre and Edmonds, 2007) use a sense tagger trained on an annotated sense inventory such as WordNet (Miller, 1995). However, this approach requires individual taggers and sense inventories for each matrix and embedded language, making it a serious challenge to extend PolyGloss to low-resource languages.

Instead, we propose to filter candidate perturbations using the embedded language translation of the clean example. This is easily done by checking if the candidate perturbation exists in the translation. Since our examples tend to be single sentences, the probability of different senses of the same word occurring in a single sentence is generally low (Conneau et al., 2018; Popel et al., 2020). This approach only requires a machine translation (MT) system and no extra linguistic information, making it highly scalable as long as a supervised (or unsupervised) machine translation system is available. By using gold translations instead of machine translations, it is even possible to mostly guarantee semantic preservation at the word-level.

Algorithm 1 Bumblebee

```
Require: Clean example-label pair $(x, y)$, Target Model $\mathcal{M}$,
  Embedded languages $\mathbb{L}$
Ensure: Adversarial example $x'$
  $T \leftarrow$ Translate($x$, target-languages $= \mathbb{L}$)
  $\mathcal{L}_x \leftarrow$ GetLoss($\mathcal{M}, x, y$)
  $B \leftarrow \{(\mathcal{L}_x, x, 0)\}$                          ▷ Initialize beam
  $\mathbb{P} \leftarrow$ AlignAndExtractPhrases($x, T$)
  while NotEmpty($B$) do
      $\mathcal{L}_{x_c}, x_c, i \leftarrow$ Poll($B$)
      $C \leftarrow$ GetCandidates($x_c, \mathbb{P}[i]$)
      $\mathcal{L} \leftarrow$ GetLoss($\mathcal{M}, C, y$)                ▷ Losses for $C$
      $i \leftarrow i + 1$
      UpdateBeam($B, \mathcal{L}, C, i$)
  end while
  $x' \leftarrow$ Poll($B$)
  return $x'$
```

3.2 Bumblebee: Phrase-Level Adversaries

Although using bilingual dictionaries with our filtering method ensures that the semantics of a borrowed word matches the original, the dictionary's comprehensiveness determines the presence of sufficient candidate adversaries. In addition, PolyGloss swaps words at the word level, which may hurt the naturalness of the resulting sentence since it is more common for code-mixers to borrow phrases than individual words (Abuhakema, 2013).

A solution to these issues is to replace phrases in the matrix sentence with their equivalents from the reference translations instead of using a dictionary lookup (Alg. 1). A key advantage of this approach is its flexibility and scalability to more languages since it only requires parallel bitexts from the matrix and embedded languages. With the advent of neural sequence-to-sequence models, such bitexts can be easily generated using publicly available MT models. However, a key challenge for this approach is extracting the matrix-embedded phrase pairs from the clean example and its translation. We follow common phrase-based machine translation methods and accomplish this by aligning the matrix and embedded sentences (Koehn, 2010). Implementation details can be found in Appendix B.2.

Syntactic preservation. To improve the adversaries' naturalness, we impose an *equivalence constraint* (Poplack, 1980), preventing a perturbation from being applied if it is from the same language as the previous word *and* will disrupt the syntax of the current phrase if applied (Winata et al., 2019). Such disruptions usually occur when borrowing words from languages with a different word order.

Model	XNLI-13			XNLI-31	
	Clean	$\text{PG}_{\text{uf.}}$	$\text{PG}_{\text{filt.}}$	Clean	$\text{PG}_{\text{filt.}}$
XLM-R_{large}	81.10	**6.06**	28.28	80.60	**8.76**
XLM-R_{base}	75.42	**2.17**	12.27	74.75	**3.57**
$\text{mBERT}_{\text{base}}$	67.54	**2.15**	9.24	66.56	**3.11**
$\text{Unicoder}_{\text{base}}$	74.98	**1.99**	11.33	74.28	**3.73**

Table 2: POLYGLOSS (PG) results (accuracy) on XNLI-13 and -31 test sets with beam width = 1. $\text{PG}_{\{\text{filt., uf.}\}}$ indicates whether the candidate perturbations were filtered using reference translations. Clean accuracy scores are the averages across all languages in the test set. Lower is better.

Model	XNLI-13		Standard XNLI		
	Clean	BB	Clean	Rand.	BB
XLM-R_{large}	81.10	**11.31**	79.85	75.04	**8.18**
XLM-R_{base}	75.42	**5.08**	74.06	65.19	**3.53**
$\text{mBERT}_{\text{base}}$	67.54	**6.10**	65.66	59.17	**4.45**
$\text{Unicoder}_{\text{base}}$	74.98	**4.81**	73.69	65.55	**3.61**

Table 3: BUMBLEBEE (BB) results on XNLI with beam width = 1. *Clean* accuracies are the averages across all languages in each test set. We include a *random* (Rand.) baseline by randomly (rather than adversarially) perturbing sentences and report the average across 5 seeds. Lower is better.

Model	Clean	BUMBLEBEE
XLM-R_{large}	75.64 / 61.39	**35.32 / 22.52**
XLM-R_{base}	68.90 / 53.50	**17.95 / 10.33**
$\text{mBERT}_{\text{base}}$	64.66 / 49.47	**20.66 / 11.68**

Table 4: BUMBLEBEE results on XQuAD (F_1/EM).

4 Experiments

We first evaluate POLYGLOSS and BUMBLEBEE on XNLI (Conneau et al., 2018), then evaluate the stronger attack on XQuAD (Artetxe et al., 2020). XNLI is a multilingual dataset for natural language inference (NLI) with parallel translations for each example in fifteen languages. Each example comprises a premise, hypothesis, and a label with three possible classes: {contradiction, neutral, entailment}. We construct two more datasets from XNLI: XNLI-13 and XNLI-32. XNLI-13 comprises all XNLI languages except Swahili and Urdu due to the lack of suitable dictionaries for POLYGLOSS. We then translate the English test set into eighteen other languages with MT systems to form XNLI-31, increasing the number of embedded languages POLYGLOSS can draw from. XQuAD is a multilingual dataset for extractive question answering (QA) with parallel translations in eleven languages. In the cross-lingual transfer setting, the models are trained on English data, MNLI (Williams et al., 2018) and SQuAD 1.1 (Rajpurkar et al., 2016), and tested on mulitlingual data, XNLI and XQuAD, respectively. We perturb the premise and hypothesis for NLI and only the question for QA. More experimental details can be found in Appendix D.

Matrix language. Although our attacks work with any language as the matrix language, we use English as the matrix language in our experiments due to the availability of English→T translation models and the prevalence of English as the matrix language in many code-mixing societies.

Models. We conduct our experiments on three state-of-the-art massive multilingual encoder models: XLM-RoBERTa, mBERT, and Unicoder, each pretrained on more than 100 languages.

4.1 Results

From Tables 2 and 3, we observe that all the models are significantly challenged by adversarial code-mixing, though XLM-R_{large} is the most robust to both attacks, likely due to having more parameters. However, even after filtering POLYGLOSS's candidate perturbations by the gold translations in XNLI-13, we observe an average drop in accuracy of 80.01%, relative to the models' accuracy on the clean XNLI-13. BUMBLEBEE induces even greater performance drops (average relative decrease of 90.96% on XNLI-13), likely due to its word aligner yielding more candidates than POLYGLOSS's dictionary lookup. Increasing the number of embedded languages POLYGLOSS can draw upon results in greater drops in model performance (average relative decrease in accuracy of 93.66% on XNLI-31).

BERT- vs. XLM-based. We notice that mBERT is more sensitive to intra-phrasal syntactic disruption than the XLM-based models. mBERT is the most robust to BUMBLEBEE out of all the base models when the equivalence constraint is in place, yet is the least robust to POLYGLOSS. However, the latter trend is replicated for BUMBLEBEE if we remove this constraint (Table 16 in Appendix G). A *possible* explanation is that XLM-R and Unicoder were trained on monolingual CommonCrawl (CC) data, while mBERT was trained on multilingual Wikipedia, which could be considered as aligned at the article level since there are articles on the same topic in different languages. Hence, it is possible that this helped to align the languages more accurately in the feature space but made it more sensitive to syntactic disruptions. However, many other hyperparameters differ between the two that could have also influenced their robustness. Hence, we leave a rigorous study of these factors to future work. The higher performance of the XLM-based models on clean data can likely be attributed to the CC corpus being an order of magnitude larger than multilingual Wikipedia (Lauscher et al., 2020).

Candidate filtering. In the *unfiltered* setting, it is impossible for PolyGloss to discriminate between valid and invalid senses for a given context. Hence, a potential criticism is that the large difference in PolyGloss's success rate between the filtered and unfiltered settings could be attributed to the inappropriate senses of polysemous words being chosen and disrupting the semantics of the sentence. On the other hand, filtering perturbations with reference translations of the sentence shrinks the space of perturbations to ~1 per language. Due to the dictionaries' non-exhaustive nature, not every word in the matrix sentence has an entry in the dictionary to begin with, making this filtering step a significant reduction of the space of candidates.

To determine the likely cause of the accuracy difference between the filtered and unfiltered settings in XNLI-13, we increase the number of languages available to PolyGloss to thirty-one. If the difference between the filtered and unfiltered settings were *not* due to a lack of sufficient candidates, we should observe only a minor difference between the filtered settings for both XNLI-13 and -31. However, we observe a 69% drop for XLM-R$_{\text{large}}$, indicating that the former accuracy difference is likely due to the reduced number of valid candidates.

Phrase-level adversaries. In addition to generating more fluent sentences (Table 1), extracting the candidate perturbations directly from the translations does away with the need for sense disambiguation and increases the number of perturbations per example since it is not limited to a static dictionary. The increased effectiveness of Bumblebee compared to PolyGloss (1.13x) is further evidence that a key factor to the success of such adversarial attacks is the availability of sufficient candidates; increasing the dimensionality of the search space increases the probability that an adversarial example for the model exists (Goodfellow et al., 2015). We also include a non-adversarial baseline (Rand.) by sampling candidates from a uniform distribution instead of searching for the worst-case perturbations. Our results in Table 3 indicate that the worst-case performance of multilingual models on code-mixed data may be much lower than the scores reported on human-produced test sets since they were not created in a targeted, adversarial fashion. Experiments on beam width and a proof of concept for fully unsupervised adversaries are in Appendix E.

Model	Devanagari	Transliterated (Latin)
XLM-R$_{\text{large}}$	61.35	**41.97**
XLM-R$_{\text{base}}$	48.62	**30.01**
mBERT$_{\text{base}}$	37.70	**23.41**
Unicoder$_{\text{base}}$	49.34	**30.00**

Table 5: Bumblebee results on XNLI$_{\text{en,hi}}$ using both Devanagari and Latin scripts. Lower is better.

Transliteration. Since real-life code-mixers often use a single script for the entire sentence, we now test the effect of transliteration on Bumblebee's success rate for the English + Hindi language pair. We accomplish this by transliterating all candidates from Devanagari into Latin using the dictionaries released by Roark et al. (2020). From Table 5, we see that transliteration significantly affects the robustness of all models, even the XLM-based ones which were pretrained on similar data.

XQuAD. We observe that both XLM-R and mBERT are significantly challenged by Bumblebee even though only the question was modified (Table 4). We did not experiment on Unicoder to reduce carbon costs since its performance was almost identical to XLM-R$_{\text{base}}$ in our XNLI experiments.

PolyGloss or Bumblebee? As expected, inspection of individual adversarial examples revealed that Bumblebee generated more natural sentences than PolyGloss since the languages used within phrases were more consistent (Table 1). However, incorrect alignments due to the word aligner's probabilistic nature could introduce occasional noise into the adversarial examples. For example, we found "the" (en) to be often aligned with "的" (zh) even though the former is an article and the latter a possessive. We observe that the aligner performs better when the sentences have similar word orders (e.g., English-French vs. English-Chinese) and we can expect the adversaries generated in these settings to be more natural. Hence, we recommend PolyGloss when greater preservation of word-level semantics is desired, and Bumblebee when phrase-level perturbations are desired or bilingual dictionaries are unavailable.

Discussion. K et al. (2020) noted significant performance drops in XNLI accuracy for mBERT when the premise and hypothesis were in different languages (Fake English vs. {Hindi, Russian, Spanish}), theorizing this to be an effect of disrupting the model's reliance on lexical overlap. Our experiments in §4 and §5 lend support to this hypothesis. In Table 1, we see multiple examples where the prediction was flipped from "contradiction" to "entailment" simply by perturbing a few words. If the models did not rely on lexical overlap but performed comparisons at the semantic level, such perturbations should not have severely impacted their performance. Our results on QA also corroborate Lee et al. (2019)'s finding that models trained on SQuAD-style datasets exploit lexical overlap between the question and context.

5 Code-Mixed Adversarial Training

Finally, we propose code-mixed adversarial training (CAT), an extension of the standard adversarial training paradigm (Goodfellow et al., 2015), to improve the robustness of multilingual models to adversarial polyglots. In standard adversarial training, adversarial attacks are run on the training set to generate adversaries for training. However, this makes adversarial training computationally expensive. Hence, we take inspiration from Tan et al. (2020a)'s method of randomly sampling perturbations from an adversarial distribution and generate code-mixed perturbations using word alignment.

To generate the code-mixed adversarial training set X', we first compute the adversarial distribution $\mathcal{P}_{adv}$ by enumerating the perturbations per embedded language in all successful adversaries (§4). Formally, $\mathcal{P}_{adv} = \{f_i\}_{i=1...|\mathbb{L}|}$, where $f_i = \frac{l_i}{\sum_{j=1}^{|\mathbb{L}|} l_j}$ and $\mathbb{L}$ is the set of embedded languages.

Next, for each clean example x, we sample n languages from $\mathcal{P}_{adv}$ before translating the example into the n languages and aligning the translations with x. For sentence-pair classification tasks like NLI, we use a per-sentence n to further increase variation. Intuitively, limiting n improves the example's naturalness and the algorithm's efficiency (the alignment is the most costly step). We then extract phrases from the aligned sentences, yielding our candidate perturbations $\mathbb{P}$. Next, we sample a perturbation with probability ρ from $\mathbb{P}$ for each phrase in x. Reducing ρ yields more natural sentences since they will be less perturbed. Finally, we apply these perturbations to x, obtaining a CAT example x'. Doing this k times for all x in X and adding the result to X yields X' (Alg. 3 in Appendix G).

In contrast to running the adversarial attack on the training set, sampling perturbations from a distribution does not guarantee that the resulting example will be adversarial to the model. This issue can be mitigated by increasing the number of CAT examples observed during training. However, this would increase the computational cost if we were to train the model for the same number of epochs. Hence, we set k to one less than the number of epochs XLM-R$_{\text{base}}$ was fine-tuned for in §4 and train the model for one epoch on the adversarial training set. This exposes the model to more variation in the same number of training steps.

Setting. We conduct our experiments on NLI with XLM-R$_{\text{base}}$ with no loss of generality. In §4, the model was trained for ten epochs. Hence, we set $k = 9, n = 2, \rho = 0.5$ for CAT and train all models for a similar number of steps (60k) with the same hyperparameters as §4. We first test the models on the BUMBLEBEE adversaries generated in §4 before directly attacking the model. Next, we construct more realistic settings by running BUMBLEBEE with only 1-2 embedded languages from standard XNLI, Swahili (sw), Hindi (hi), and Urdu (ur). These languages were the lowest resourced in the pretraining data (Conneau et al., 2020a).

We also construct another *non-adversarial* test set from XNLI by randomly choosing hypotheses and premises from different languages (K et al., 2020). Since the original examples are individually monolingual, this test set will reveal if a model is simply exploiting lexical overlap rather than comparing the underlying concepts.

Finally, we run BUMBLEBEE with embedded languages not seen during task-specific training *and* from a different family (Austronesian) from the XNLI languages, Filipino (tl) and Indonesian (id). This zero-shot defense setting will reveal if CAT encourages the learning of more language-invariant representations, or is simply allowing the model to adapt to the adversarial distribution.

Baselines. Since training on languages in the test set takes us out of the cross-lingual transfer setting, we train a *translate-train-n* baseline for a fair comparison. In this setting, we train on every x and its translations in the n languages sampled in CAT, regardless of whether they contributed words to the final CAT examples. We also include Ganin et al. (2016)'s domain adversarial neural network (DANN), which has been used for cross-lingual adaptation (Joty et al., 2017; Chen et al., 2018).

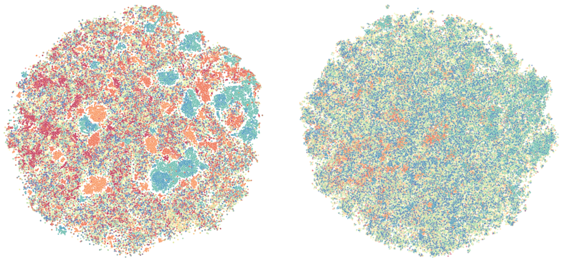
Condition/Method	**Clean**	**Clean$_{\text{DL}}$**	**Adv$_{§4}$**	**Adv$_{\text{sw}}$**	**Adv$_{\text{hi+ur}}$**	**Adv$_{\text{XNLI}}$**	**Adv$_{\text{tl}}$**	**Adv$_{\text{id+tl}}$**
Cross-lingual transfer (from §4)	74.06	66.09	3.53	38.54	29.12	3.53	36.96	24.83
Translate-train-n	**77.25**	72.01	29.44	50.53	40.63	7.04	44.37	33.23
DANN (Ganin et al., 2016)	51.86	35.10	33.45	16.02	17.52	6.54	12.05	7.06
Code-mixed adv. training (CAT)	77.10	**75.46**	**50.21**	**58.58**	**48.20**	**12.63**	**49.14**	**38.16**

Table 6: Results on standard XNLI with XLM-R$_{\text{base}}$. Clean refers to the combined test set of all languages, Clean$_{\text{DL}}$ to the variant where the hypothesis and premise of each example are from different languages, Adv$_{§4}$ to the BUMBLEBEE adversaries from §4, and Adv$_{\{\text{lgs}\}}$ to new adversaries from English + the subscripted languages. Higher is better.

5.1 Results

From Table 6, we observe that both training on fully translated data and on CAT examples improved accuracy on the non-adversarial test sets and robustness to code-mixed adversaries, compared to the cross-lingual transfer model that was only trained on English data. Similar to K et al. (2020), we found that disrupting the models' reliance on lexical overlap (Clean$_{\text{DL}}$) hurt performance. The drop was particularly significant for the cross-lingual transfer (8 points) and *translate-train-n* models (5.24 points). On the other hand, our CAT model only suffered a 1.5-point drop, indicating that the former two models likely rely heavily on lexical overlap to make predictions, while our CAT model may be using "deeper", more language-agnostic features. Crucially, our CAT model achieves similar to better clean accuracy than the baselines, contrasting with prior work showing that adversarial training hurts clean accuracy (Tsipras et al., 2019). Finally, our CAT model is >1.7x more robust to adversaries constructed from all fifteen XNLI languages than the *translate-train-n* model. Although DANN-type training improved robustness to the previous BUMBLEBEE adversaries, clean performance was significantly degraded and BUMBLEBEE was able to find even more damaging adversaries upon attacking the model directly.

When attacked with 1-2 embedded languages that were seen during training, CAT also yields significant improvements in robustness over the baselines: a $>$ 7 point increase compared to *translate-train-n* and a >19 point gain over the zero-shot transfer setting. In the zero-shot defense setting, CAT shows a >12-point gain over the zero-shot transfer model and a >4.7-point gain over the *translate-train-n* model. We believe these results to be due to CAT encouraging the learning of language-invariant representations by exposing the model to cross-lingual lexical variation and preventing the model from exploiting lexical overlaps.



(a) Cross-Lingual Transfer (b) CAT

Figure 2: t-SNE visualizations of XLM-R$_{\text{base}}$ representations fine-tuned using different methods.

6 Seeing is Believing

To further understand the effect of various fine-tuning methods on XLM-R$_{\text{base}}$, we visualize the <s> vector from the layer before the classification head using t-SNE (Linderman et al., 2019). Here, all sentences from XNLI are passed through the representations individually. If a representation were 100% language-invariant, we should expect t-SNE to be unable to separate individual languages into their own clusters. Hence, the extent to which t-SNE is able to do so would indicate the amount of language-specific information in this last layer.

From Fig. 2a, we observe that for the cross-lingual transfer model (§4), t-SNE managed to organize the sentences from several languages (Chinese, Hindi, Thai, Urdu) into distinct clusters. This indicates that a significant amount of language-specific information remains in the vector representations of sentences from these languages. Visualizing the sequence-averaged embeddings makes this even clearer (Fig. 5 in Appendix G). Hence, while XLM-R may be multilingual, it appears to be structured as a space of individual language subspaces as opposed to a mixed, or language-invariant space. On the other hand, t-SNE was much less successful when given the representation trained with CAT (Fig. 2b). Mixing multiple languages in the same sentence and showing the model multiple variants of the same sentence likely encourages the model to refine its representation such that all variants of the same sentence are represented similarly, resulting in a more language-invariant representation. T-SNE plots of the other models are in Appendix G.

7 Limitations and Future Work

We acknowledge that our methods do not fully model real code-mixing since we do not learn the mixing patterns from real data and there are subtleties in real code-mixing we ignore for simplicity, e.g., accounting for the prestige of participating languages (Bhatia, 2011). In addition, it is impossible to *guarantee* the semantic preservation of a sentence generated by BUMBLEBEE due to the word aligner's statistical nature, though we can expect more accurate alignments to improve semantic preservation. Finally, while CAT improves robustness, there remains a significant gap between the robust and clean accuracies. In line with recent work challenging the Anglocentricity of cross-lingual models (Anastasopoulos and Neubig, 2020; Liang et al., 2020), a promising direction of future work lies in investigating how the choice of matrix language affects model robustness.

8 Conclusion

Ensuring that multilingual models are robust to both natural and adversarial code-mixing is important in today's increasingly multilingual world if they are to allow their target users to fully express themselves in human-machine conversations and to defend against adversarial users attempting to evade toxicity/misinformation detection systems.

To approximate a lower bound for model performance on lexical code-mixing, we propose two strong black-box multilingual adversarial attacks and demonstrate their effectiveness on state-of-the-art cross-lingual NLI and QA models. The former generates perturbations from bilingual dictionaries and disambiguates between senses using sentence translations, while the latter generates perturbations by aligning sentences from different languages.

Next, we show that training on code-mixed data synthesized via word alignment improves clean and robust accuracy when models are prevented from exploiting lexical overlap *without* hurting clean accuracy. Crucially, we achieve this in the same number of steps as standard supervised training.

Finally, we use t-SNE visualizations to show that multilingual models are not necessarily language-invariant and that our code-mixed adversarial training scheme encourages language-invariance.

9 Broader Impact / Ethical Considerations

Adversarial attacks and defenses are double-edged swords. On one hand, adversarial examples expose the gaps in existing models and help to focus the research community's attention on flaws that need to be addressed before these models can be used reliably in noisy, real-world environments. On the other, the same adversarial attacks can be used by malicious actors to bypass toxicity/misinformation detection systems. Similarly, methods for improving adversarial robustness can be used to defend against malicious actors and improve robustness to natural noise or linguistic variation, yet they can also be used to strengthen automated censorship systems and limit freedom of speech. For example, our adversarial attacks could be used both as a lower bound for model performance on naturally occurring code-mixed text *and* to bypass misinformation detection systems while preserving the message's intelligibility for multilingual speakers. Our adversarial training method could be used to both improve machine understanding of code-mixers by making multilingual representations more language-invariant *and* suppress the freedom of speech of polyglots who could have been using code-mixing to evade censorship.

At the same time, technology strongly shapes our behavior (Reeves et al., 2019). Consequently, given the centrality of code-switching/mixing to many polyglots' lived experiences (Duff, 2015) and the positive correlations between multilingualism, code-switching, and creativity (Leikin, 2013; Kharkhurin and Wei, 2015; Fürst and Grin, 2018), we should ensure that the natural language technologies we build do not inhibit multilingual speakers from fully expressing themselves, e.g., by discouraging code-mixing due to non-understanding. In addition, studies have found that aphasic polyglots code-mix more frequently than neurotypical polyglots to cope with word-retrieval difficulties (Goral et al., 2019), making it important for natural language technologies to be robust to code-mixing if they are to be inclusive. Therefore, we include both adversary generation and defense methods to avoid tipping the balance too far in either direction.

Acknowledgments

We would like to thank Cynthia Siew (NUS Psychology), Greg Bennett and Kathy Baxter (Salesforce), Lav Varshney (UIUC Electrical and Computer Engineering), Min-Yen Kan (NUS Computer Science), and our anonymous reviewers for their invaluable feedback. We are also grateful to Guangsen Wang, Mathieu Ravaut, Soujanya Lanka, and Tuyen Hoang for contributing manually code-mixed sentences and Bari M Saiful for pointers on replicating the XLM-R results. Samson is supported by Salesforce and Singapore's Economic Development Board under its Industrial Postgraduate Programme.

References

Ghazi M Abuhakema. 2013. Code switching and code mixing in arabic written advertisements: Patterns, aspects, and the question of prestige and standardisation. *The Internet Journal Language, Culture and Society*.

Eneko Agirre and Philip Edmonds. 2007. *Word Sense Disambiguation: Algorithms and Applications*, 1st edition. Springer Publishing Company, Incorporated.

Gustavo Aguilar and Thamar Solorio. 2020. From English to code-switching: Transfer learning with strong morphological clues. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8033–8044, Online. Association for Computational Linguistics.

Emily Ahn, Cecilia Jimenez, Yulia Tsvetkov, and Alan Black. 2020. What code-switching strategies are effective in dialogue systems? *Proceedings of the Society for Computation in Linguistics*, 3(1):308–318.

Fahad AlGhamdi and Mona Diab. 2019. Leveraging pretrained word embeddings for part-of-speech tagging of code switching data. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 99–109.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Antonios Anastasopoulos and Graham Neubig. 2020. Should all cross-lingual embeddings speak English? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8658–8679, Online. Association for Computational Linguistics.

Larissa Aronin and David Singleton. 2008. Multilingualism as a new linguistic dispensation. *International Journal of Multilingualism*, 5(1):1–16.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "i am borrowing ya mixing?" an analysis of english-hindi code mixing in facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126.

Kelsey Ball and Dan Garrette. 2018. Part-of-speech tagging for code-switched, transliterated texts without explicit language identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3084–3089, Brussels, Belgium. Association for Computational Linguistics.

M Saiful Bari, Tasnim Mohiuddin, and Shafiq Joty. 2020. Multimix: A robust data augmentation framework for cross-lingual nlp. *arXiv preprint arXiv:2004.13240*.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the first workshop on computational approaches to code switching*, pages 13–23.

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! *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–23.

Tej K Bhatia. 2011. The multilingual mind, optimization theory, and hinglish. *Chutneying English: The Phenomenon of Hinglish*, pages 37–52.

Matthias Blohm, Glorianna Jagfeld, Ekta Sood, Xiang Yu, and Ngoc Thang Vu. 2018. Comparing attention-based convolutional and recurrent neural networks: Success and limitations in machine reading comprehension. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 108–118.

Khyathi Chandu, Ekaterina Loginova, Vishal Gupta, Josef van Genabith, Günter Neumann, Manoj Chinnakotla, Eric Nyberg, and Alan W Black. 2019. Code-mixed question answering challenge: Crowd-sourcing data and techniques. In *Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 29–38. Association for Computational Linguistics (ACL).

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. 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.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. 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.

Dileep A Divekar. 1984. Dc statistical circuit analysis for bipolar ic's using parameter correlations-an experimental example. *IEEE transactions on computer-aided design of integrated circuits and systems*, 3(1):101–103.

Patricia A Duff. 2015. Transnationalism, multilingualism, and identity. *Annual Review of Applied Linguistics*, 35:57.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Guillaume Fürst and François Grin. 2018. Multilingualism and creativity: a multivariate approach. *Journal of Multilingual and Multicultural Development*, 39(4):341–355.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations*, San Diego, California.

Mira Goral, Monica Norvik, and Bård Uri Jensen. 2019. Variation in language mixing in multilingual aphasia. *Clinical linguistics & phonetics*, 33(10-11):915–929.

Wenjuan Han, Liwen Zhang, Yong Jiang, and Kewei Tu. 2020. Adversarial attack and defense of structured prediction models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online. Association for Computational Linguistics.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. 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)*, pages 2485–2494.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

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

Shafiq Joty, Preslav Nakov, Lluís Màrquez, and Israa Jaradat. 2017. Cross-language learning with adversarial neural networks. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 226–237, Vancouver, Canada. Association for Computational Linguistics.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.

Braj B Kachru. 1978. Toward structuring code-mixing: An indian perspective. *International Journal of the Sociology of Language*, 1978(16):27–46.

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

Anatoliy V Kharkhurin and Li Wei. 2015. The role of code-switching in bilingual creativity. *International Journal of Bilingual Education and Bilingualism*, 18(2):153–169.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474, Mumbai, India. The COLING 2012 Organizing Committee.

Philipp Koehn. 2010. *Statistical Machine Translation*, 1st edition. Cambridge University Press, New York, NY, USA.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online. Association for Computational Linguistics.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Mark Leikin. 2013. The effect of bilingualism on creativity: Developmental and educational perspectives. *International Journal of Bilingualism*, 17(4):431–447.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium*.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *arXiv preprint arXiv:2004.01401*.

George C Linderman, Manas Rachh, Jeremy G Hoskins, Stefan Steinerberger, and Yuval Kluger. 2019. Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. *Nature methods*, 16(3):243–245.

Yaron Matras and Jeanette Sakel. 2007. *Grammatical borrowing in cross-linguistic perspective*, volume 38. Walter de Gruyter.

Paul Michel, Xian Li, Graham Neubig, and Juan Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. 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 3103–3114, Minneapolis, Minnesota. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41.

Benjamin Muller, Benoit Sagot, and Djamé Seddah. 2020. Can multilingual language models transfer to an unseen dialect? a case study on north african arabizi. *arXiv preprint arXiv:2005.00318*.

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

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 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain. Association for Computational Linguistics.

Wannaphong Phatthiyaphaibun, Korakot Chaovavanich, Arthit Suriyawongkul Charin Polpanumas, Lalita Lowphansirikul, and Pattarawat Chormai. 2016. PyThaiNLP: Thai Natural Language Processing in Python.

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

Pavlin G. Poličar, Martin Stražar, and Blaž Zupan. 2019. openTSNE: a modular Python library for t-SNE dimensionality reduction and embedding. *bioRxiv*.

M. Popel, M. Tomková, J. Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondrej Bojar, and Z. Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11.

Shana Poplack. 1980. Sometimes i'll start a sentence in spanish y termino en español: toward a typology of code-switching. *Linguistics*, 18(7-8):581–618.

Shana Poplack, David Sankoff, and Christopher Miller. 1988. The social correlates and linguistic processes of lexical borrowing and assimilation. *Linguistics*, 26(1):47–104.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Byron Reeves, Nilam Ram, Thomas N Robinson, James J Cummings, C Lee Giles, Jennifer Pan, Agnese Chiatti, MJ Cho, Katie Roehrick, Xiao Yang, et al. 2019. Screenomics: A framework to capture and analyze personal life experiences and the ways that technology shapes them. *Human–Computer Interaction*, pages 1–52.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J. Mielke, Cibu Johny, Işin Demirşahin, and Keith Hall. 2020. Processing South Asian languages written in the Latin script: the Dakshina dataset. In *Proceedings of The 12th Language Resources and Evaluation Conference (LREC)*, pages 2413–2423.

Uma Roy, Noah Constant, Rami Al-Rfou, Aditya Barua, Aaron Phillips, and Yinfei Yang. 2020. LAReQA: Language-agnostic answer retrieval from a multilingual pool. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5919–5930, Online. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, and Hinrich Schütze. 2020. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. *arXiv preprint arXiv:2004.08728*.

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*.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.

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

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations*, Banff, AB, Canada.

Peck Tung Tan. 1988. A description of patterns of code-mixing and code-switching in a multilingual household. In Joseph Foley, editor, *New Englishes: The Case of Singapore*. NUS Press.

Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. 2020a. It's morphin' time! Combating linguistic discrimination with inflectional perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2920–2935, Online. Association for Computational Linguistics.

Samson Tan, Shafiq Joty, Lav Varshney, and Min-Yen Kan. 2020b. Mind your inflections! Improving NLP for non-standard Englishes with Base-Inflection Encoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online. Association for Computational Linguistics.

Mary W. J. Tay. 1989. Code switching and code mixing as a communicative strategy in multilingual discourse. *World Englishes*, 8(3):407–417.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. 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)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

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

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. 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)*, pages 833–844.

Kofi Yakpo. 2015. Code-switching and social change: Convergent language mixing in a multilingual society. *Code-switching Between Structural and Sociolinguistic Perspectives*, 43:259.

Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. 2020. Code-switching pre-training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online. Association for Computational Linguistics.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2019. Generating fluent adversarial examples for natural languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5569, Florence, Italy. Association for Computational Linguistics.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(3).

A Examples of Real Code-Mixed Text

English + Spanish (Sridhar and Sridhar, 1980)

- The man que vino ayer (who came yesterday) wants ito buyun carro nuevo (a new car).
- El (The) old man esta enojide (is mad).
- Me lleve chile ya roasted y peeled ... para hacerlo ells. (I picked up the chile already roasted and peeled for making it there.)

Hindi + English (Bali et al., 2014)

- Befitting reply to mere papa ne maaraa (My father gave a befitting reply)
- ... and the party workers [will] come with me without virodh (protest/objection)

Sarnami + Sranan + Dutch (Yakpo, 2015)

- dus gewoon calat jaiye, tab ego kerki links ki rechts. (So just keep on walking, then [there's] achurch, left or right.)
- kaun wálá damrú, ego haigá jaun men ná verfi bhail, ma ego wel hai. (Which [kind of] damru drum, there's one which is not coloured inside but one actually is.)

B Attack Implementation Details

B.1 PolyGloss

Algorithm 2 PolyGloss

```
Require: Clean example-label pair $(x, y)$, Target Model $\mathcal{M}$,
    Embedded languages $\mathbb{L}$
Ensure: Adversarial example $x'$
  $T \leftarrow$ TRANSLATE($x$, target-languages $= \mathbb{L}$)
  $\mathcal{L}_x \leftarrow$ GETLOSS($\mathcal{M}, x, y$)
  $B \leftarrow \{(\mathcal{L}_x, x, 0)\}$                    ▷ Initialize beam
  while NOTEMPTY($B$) do
      $\mathcal{L}_{x_c}, x_c, i \leftarrow$ POLL($B$)
      $C \leftarrow$ GETCANDIDATES($x_c$, token-id $= i$)
      $C \leftarrow$ FILTERCANDIDATES($C, T$)
      $\mathcal{L} \leftarrow$ GETLOSS($\mathcal{M}, C, y$)          ▷ Losses for $C$
      $i \leftarrow i + 1$
      UPDATEBEAM($B, \mathcal{L}, C, i$)
  end while
  $x' \leftarrow$ POLL($B$)
  return $x'$
```

To reduce the practical running time of our attack, we make use of cross-lingual dictionaries released by Lample et al. (2018) for generating candidate perturbations instead of translating the words in an online fashion. We also use the gold translations of the clean examples when they are available (such as in XNLI), and use the models released by Tiedemann and Thottingal (2020) in the `transformers` library (Wolf et al., 2019) to translate the examples to other languages. We also cache them in hashtables for fast retrieval.

B.2 Bumblebee

We use the gold translations where available and Tiedemann and Thottingal (2020)'s translation models for the other languages, and align sentences with a neural word aligner (Sabet et al., 2020) backed by XLM-R$_{\text{base}}$ in our implementation of Bumblebee. Although Sabet et al. (2020) found the "Itermax" algorithm to yield the best performance for their experimental settings, we suggest using the high recall ("Match") algorithm for candidate generation. We inspected the output of both algorithms and found that while Itermax generates more candidates, it also tends to generate noisier alignments compared to Match, which we found to be more conservative.

C PolyGloss and CAT Samples

Almost random samples for PolyGloss and CAT (we tried to include the sentences with Thai and Hindi characters but did not manage to get them to render with pdflatex).

Rockefeller έδωσε to السرطان forschung.
The 发音列表 само included the most basic things.
He vardı yedi hermanas and no brothers in his family.
But même хотя as a boy I lived on a çiftlik right on the Mexican حدود, I تذكر being mystified by ranching términos that crept into Western songs du north of us, cayuse, for example.
We should nie think of human equality when we consider social and political justice.

Table 7: PolyGloss adversaries for XLM-R$_{\text{base}}$ on XNLI-13.

Not much has sich innerhalb des Jahrzehnts verändert.
Ese كان el most historic weather يوم in la historia registrada para el el clima severo en el country
Como ustedes γνωρίζετε, last enero hemos issued un νέο όγκο de reports, la Performance και λογοδοσίας serie de, en los que se esbozan los de gestión desafíos που αντιμετωπίζουν οι nuestras mayores federal agencies and the substantial opportunities para mejorar su rendimiento.
because a lot trong số họ are are similar
um-hum bueno that's increíble como i used to cuando i was в college solía la have el stereo en all през time or i tenía en MTV or something but ever que i've been out de college

Table 8: Code-mixed adversarial training examples.

D Experiment Details

D.1 Datasets

Standard XNLI[3] comprises 7,500 *parallel* examples (2,490 dev., 5,010 test) in fifteen languages: English (en), Spanish (es), German (de), Greek (el), Russian (ru), Turkish (tr), Arabic (ar), Vietnamese (vi), Thai (th), Chinese (zh), Hindi (hi), French (fr), Bulgarian (bg), Swahili (sw), and Urdu (ur). The labels are uniformly distributed between the three classes (contradiction, neutral, entailment).

The machine-translated training set[4] of standard XNLI comprises the MNLI training examples in addition to their translations in the same fourteen non-English languages as the dev. and test sets.

XNLI-13 comprises all standard XNLI languages except Swahili and Urdu due to the lack of suitable dictionaries for PolyGloss. XNLI-31 comprises all languages in XNLI-13, in addition to another eighteen: Afrikaans (af), Albanian (sq), Catalan (ca), Czech (cs), Danish (da), Dutch (nl), Estonian (et), Filipino (tl), Finnish (fi), Hebrew (he), Hungarian (hu), Indonesian (id), Italian (it), Macedonian (mk), Romanian (ro), Slovak (sk), Swedish (sv), and Ukrainian (uk).

MNLI[5] comprises 392,702 examples in English with the following label distribution: 130,899 entailment, 130,900 neutral, 130,903 contradiction.

XQuAD[6] comprises 1,190 question-answer pairs with guaranteed answers in eleven languages: English (en), Spanish (es), German (de), Greek (el), Russian (ru), Turkish (tr), Arabic (ar), Vietnamese (vi), Thai (th), Chinese (zh), and Hindi (hi). XQuAD examples are drawn from the SQuAD 1.1 development set.

SQuAD 1.1[7] comprises 87,599 question-answer pairs with guaranteed answers in English.

D.2 Metrics

The metric used for XNLI is simple accuracy:

$$\text{Accuracy} = \frac{\#\,\text{true positive}}{\#\,\text{total}} \tag{2}$$

The metrics used for SQuAD are F_1:

$$\text{F}_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

and exact match:

$$\text{Exact Match} = \frac{\#\,\hat{y} = y}{\#\,\text{total}} \tag{4}$$

D.3 XQuAD Preprocessing

We use Phatthiyaphaibun et al. (2016) to tokenize Thai text, `jieba`[8] for Chinese text, and split all other languages on whitespace.

D.4 Training Details

Model	Params.	Lr	Bsz	Epochs
XLM-R_{large}	550M	1e-06	64	10
XLM-R_{base}	270M	5e-06	64	10
$\text{mBERT}_{\text{base}}$	172M	5e-05	64	2
$\text{Unicoder}_{\text{base}}$	270M	5e-06	64	10

Table 9: Hyperparameters for model fine-tuning on XNLI (MNLI). Number of parameters reproduced from Conneau et al. (2020a).

Model	Learning rate	Batch Size	Epochs
XLM-R_{large}	1e-05	32	3
XLM-R_{base}	3e-05	64	2

Table 10: Hyperparameters for model fine-tuning on XQuaD (SQuAD 1.1).

Hyperparameters. Table 9 contains the hyperparameters we used to fine-tune our models on MNLI. We used the hyperparameters suggested by Devlin et al. (2019)[9] for mBERT, the hyperparameters suggested by Bari et al. (2020) for the XLM-R models, and the hyperparameters from Liang et al. (2020) for Unicoder.

Table 10 contains the hyperparameters we used to fine-tune our models on SQuAD 1.1. We used the default SQuAD hyperparameters from the `transformers` library[10] for XLM-R_{base} and adjusted the hyperparameters for XLM-R_{large} to fit it onto the GPU. The mBERT model from the HuggingFace model repository[11] was used instead of fine-tuning our own. The clean scores reported in Table 4 are similar to those reported in the XQuAD GitHub repository.

[3] cims.nyu.edu/~sbowman/xnli

[4] Same link as XNLI dev/test set.

[5] cims.nyu.edu/ sbowman/multinli/

[6] github.com/deepmind/xquad

[7] rajpurkar.github.io/.../train-v1.1.json

[8] github.com/fxsjy/jieba

[9] https://github.com/google-research/.../multilingual.md

[10] github.com/huggingface/.../question-answering

[11] huggingface.co/salti/bert-base-multilingual-cased-finetuned-squad

D.5 Validation Performance for CAT Exps.

Model	Clean	Adv$_{\S4\text{-dev}}$
Cross-lingual transfer (from §4)	73.90	4.17
Translate-train-n	76.99	29.27
DANN (Ganin et al., 2016)	51.81	34.89
Code-mixed adv. training (CAT)	**77.13**	**52.32**

Table 11: Accuracy on the XNLI dev. set and BUMBLEBEE adversaries generated from the dev. set.

D.6 Infrastructure Details

Models were trained on single V100 GPUs. Attacks were run on 8 V100 GPUs, parallelized with `ray`[12] to make full use of GPU memory. On the standard XNLI test set (15 languages, 5,010 examples), BUMBLEBEE takes 60-90 minutes in total. With 1 embedded language, BUMBLEBEE runs on the test set in under 10 minutes. POLYGLOSS is generally much faster (under 30 minutes on XNLI-31) due to not needing a neural aligner.

E Extra BUMBLEBEE Experiments

Variable (v)	$v = 1$	$v = 2$	$v = 3$
Beam width	48.52%	48.95%	49.67%
Embedded lgs.	48.52%	69.09%	75.73%

Table 12: Effect of increasing the beam width vs. the number of embedded languages on the attack success rate (%) while holding the other variable constant at 1. We use Swahili, Swahili and French, and Swahili, French, and Spanish as the embedded languages when $v = 1, 2, 3$, respectively. Rates are computed relative to the average clean XNLI score on the languages involved.

Beam search. In our experiments, we found that increasing the beam width yielded a higher attack success rate. However, this increases running time with only minor improvements (Table 12). We found increasing the number of embedded languages (and hence candidates) to be a more efficient method of increasing the success rate with a minor increase in running time. Although the time complexity (in number of model queries) is $O(|B||C||\mathbb{L}||S|)$ where $|S|$ is the sentence length, increasing $|\mathbb{L}|$ had a greater impact on the success rate than increasing $|B|$ by the same number.

Model	Clean	Supervised	Unsupervised
XLM-R$_{\text{base}}$	81.32	49.00	46.78

Table 13: Accuracy after running BUMBLEBEE on XNLI in supervised and unsupervised settings (matrix: English, embedded: French).

Fully unsupervised adversaries. A potential drawback of BUMBLEBEE is that it requires translations of the clean example, which may be challenging to obtain for low-resource languages. However, it is possible to use unsupervised MT models for this purpose. We use Song et al. (2019)'s unsupervised English-French model to generate translations as a proof of concept and find that they achieve similar results (Table 13).

F Plausible Language Combinations

To explore the effect of different combinations of languages on a multilingual model's performance, we run BUMBLEBEE with different sets of embedded languages that could be plausibly spoken by (adversarial) polyglots around the world. English is used as the matrix language for all experiments.

We observe a general trend of decreasing model accuracy as we increase the number of mixed languages (Table 14), and XLM-R$_{\text{base}}$'s robustness to an embedded language appears to be generally more dependent on the size of its pretraining dataset than language typology. For example, Russian (*en+ru*) and Indonesian (*en+id*), languages with notably high accuracies, were the two most resourced languages after English in the corpus used for pretraining (Conneau et al., 2020a), while Swahili and Filipino were among the lower-resourced languages. A notable outlier is Afrikaans (*en+af*), which was the lowest resourced language in our experiments yet XLM-R$_{\text{base}}$ was quite robust to adversaries constructed from it and English. A possible explanation is that Afrikaans' language family, Indo-European, is highly represented in the pretraining corpus. Another notable outlier is Vietnamese, which was the fourth most resourced language in the pretraining corpus, yet the model was more vulnerable to adversaries constructed from English and Vietnamese than adversaries constructed from English and Filipino, one of the lowest resourced languages. A *possible* explanation for this is the use of Latin characters combined with little vocabulary overlap between English and Vietnamese, and differing adjective positions.

[12] github.com/ray-project/ray

Lgs. (en+)	Exemplar Region	Acc.	Size (GB)
tr	Turkey	55.76	20.9
de	Germany	55.42	66.6
bg	Bulgaria	54.87	57.5
ru	Russia	54.33	278.0
th	Thailand	52.43	71.7
el	Greece	52.41	46.9
zh	China	52.17	46.9
ar	Middle East	51.53	28.0
es	Spain	50.07	53.3
fr	France	49.00	56.8
hi	India	48.62	20.2
ur	Pakistan	42.09	5.7
sw	Kenya	38.54	1.6
vi	Vietnam	36.32	137.3
ro	Romania	56.26	61.4
id	Indonesia	50.91	148.3
af	Namibia	49.30	1.3
sq	Albania	40.13	5.4
tl	Philippines	36.96	3.1
ru+uk	Ukraine	44.37	362.6
ar+he	Israel	37.24	59.6
af+de	Namibia	36.46	67.9
de+fr	Switzerland	35.84	123.4
id+zh	Indonesia	35.30	195.2
hu+ro	Transylvania	34.11	119.8
ar+fr	Morocco	32.15	84.8
hi+ur	Kashmir	29.12	25.9
ar+sw	Tanzania	24.37	29.6
fi+sv+ru	Finland	30.27	344.4
cs+hu+sk	Slovak	30.21	97.2
es+fr+it	S. Europe	26.58	140.3
mk+sq+tr	Albania	23.53	31.1
da+de+nl+sv	Denmark	27.74	153.6
id+th+tl+vi	S.E. Asia	12.53	360.4

Table 14: How XLM-R$_{\text{base}}$ *might* fare against real-life adversarial polyglots and where they might be found. Scores are accuracy on the XNLI test set with English as the matrix language. Corpora sizes reproduced from Conneau et al. (2020a).

We also find XLM-R$_{\text{base}}$ to be twice as robust in the *en+da+de+nl+sv* condition as the *en+id+th+tl+vi* condition. It is likely that the structural similarity of the mixed languages also plays an important role in determining the model robustness, reinforcing K et al. (2020)'s similar findings on cross-lingual transfer. Investigating the effects of typology, vocabulary and orthographic overlap, pretraining corpus size, and interaction effects between different sets of languages on a model's robustness to code-mixed adversaries could lead to new insights into how different languages interact in the multilingual embedding space and we leave this to future work.

G More Tables, Figures, and Algorithms

Algorithm 3 CAT Example Generation

```
Require: Original examples $X$, Embedded languages $\mathbb{L}$,
  Num. perturbed examples $k$, Adversarial distribution $\mathcal{P}_{adv}$,
  Max. langs. per example $n$, Phrase perturbation prob. $\rho$
Ensure: Adversarial training set $X'$
  $X' \leftarrow \{\varnothing\}$
  for $x$ in $X$ do
      $\mathbb{S} \leftarrow \textsc{SampleLanguages}(\mathbb{L}, n, \mathcal{P}_{adv})$
      $T \leftarrow \textsc{Translate}(x, \text{target-languages} = \mathbb{S})$
      $\mathbb{P} \leftarrow \textsc{AlignAndExtractPhrases}(x, T)$
      for $i = 1$ to $k$ do
          $x' \leftarrow \textsc{Perturb}(x, \mathbb{P}, \rho)$
          $X' \leftarrow X' \cup \{x'\}$
      end for
  end for
  return $X \cup X'$
```

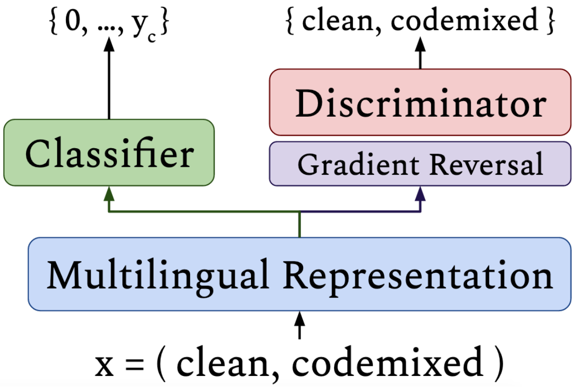


Figure 3: Domain Adversarial Neural Network

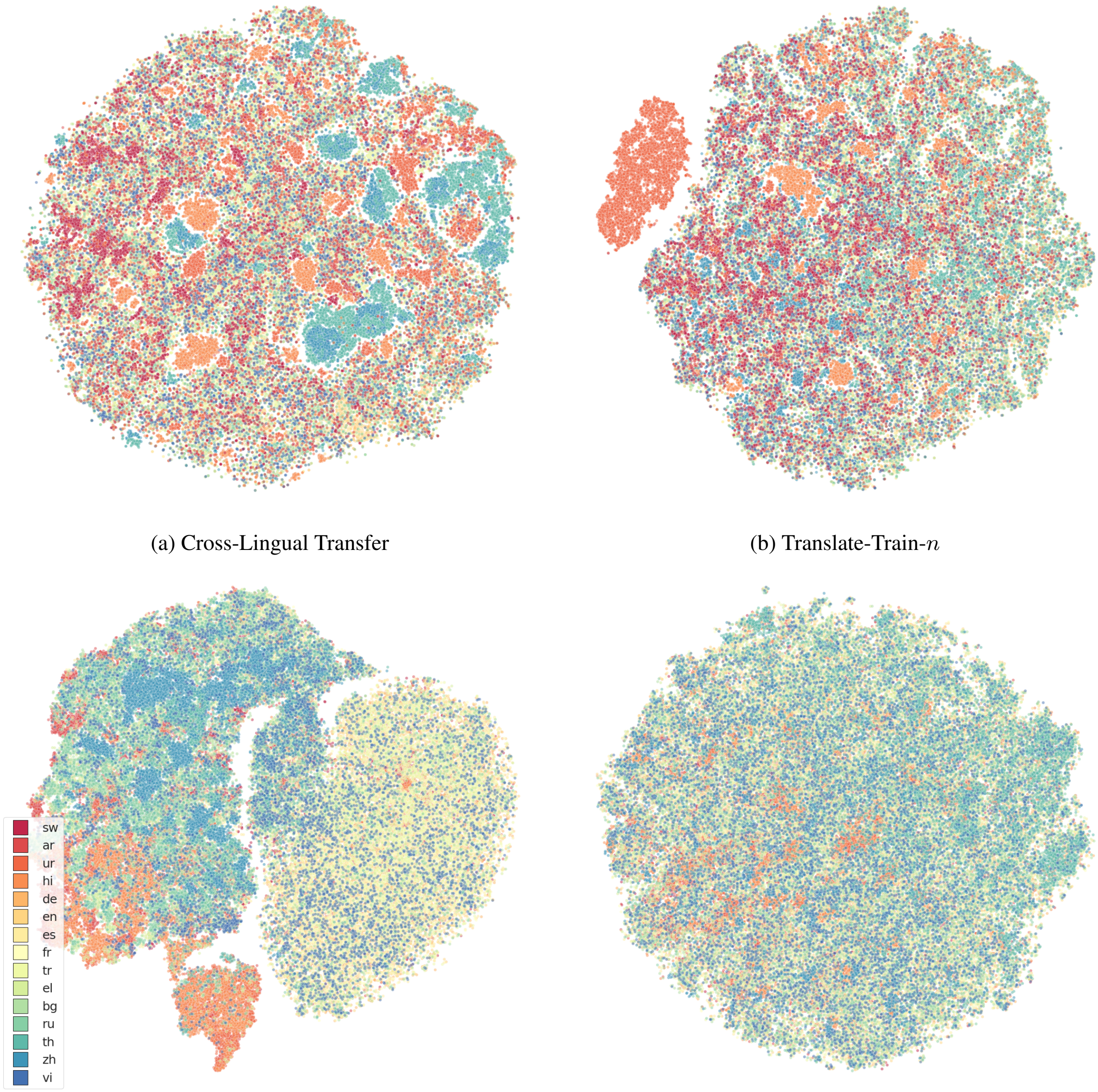
Section continues on next page.

English	**P:** Americans should also consider how to do it-organizing their government in a different way. **H:** The American government might be organized in a different way.
French (R)	**P:** Les Américains devraient aussi réfléchir to de le faire, en organisant in la different way. **H:** The gouvernement américain might be organized in a different way.
French (BB)	**P:** Americans should also penser à comment organiser leur gouvernement differemment. **H:** Le gouvernement americain est maybe organized differently.
Hindi (R)	**P:** Americans ko yeh bhi sochna hai ki kaise karna hai-organizing their government in a different way. **H:** America ki sarkar might be organized in a different way.
Hindi (BB)	**P:** amerikiyon chahiye ki also vichar kese to karna it-organizing their government in a different way. **H:** The American government might be organized in a different way.
Chinese (R)	**P:** Americans应该consider下怎样去重组他们的government **H:** 美国government可以有其它的organization方式
Vietnamese (R)	**P:** Người Mỹ cũng nên consider how to do it-organizing their government theo một cách khác. **H:** Chính phủ Mỹ có thể được organized theo một cách khác.
English	**P:** When that occurs, the lending fund sacrifices interest from Treasury securities on its invested balances and instead receives interest from the borrowing fund on the amount of the loan. **H:** The lending fund doesn't get all the interest in some cases.
French (R)	**P:** Lorsque cela se produit, le lending fonds de sacrifie interest des Trésor securities sur ses investis balances et plutôt receives intérêts du borrowing fund sur le montant du les loan. **H:** The lending fund ne reçoit pas all the interest in some cases.
French (BB)	**P:** Quand ça arrive, le lending fund sacrifie l'intéret des Treasury securities sur ses investissements et recoit des interest from the borrowing fund on the amount du prêt. **H:** Le fending fund ne reçoit pas tous les interest in some cases.
Hindi (R)	**P:** Jab aisa hota hai, the lending fund sacrifices interest from Treasury securities on its invested balances and instead receives interest from the borrowing fund on the amount of the loan. **H:** Aise mamlo mein, the lending fund doesn't get all the interest.
Chinese (R)	**P:** 这种情况下，lending fund会损失从treasury securities的investment interests，但是可以从borrowing fund那里拿到interest **H:** lending fund在某些case下不会拿到所有的interest
Chinese (BB)	**P:** 当 that occurs, the lending fund sacrifices 利息从 Treasury securities 在其投资余额，而 receives 利息 from the borrowing 基金 on the 金额的 the loan. **H:** The lending 基金并 doesn't get all the interest in some cases.
Vietnamese (R)	**P:** Khi điều ấy xảy ra, the lending fund sacrifices interest from Treasury securities on its invested balances và thay vào đó receives interest from the borrowing fund on the amount of the loan. **H:** The lending fund doesn't get all the interest trong một số trường hợp.
Vietnamese (BB)	**P:** Khi đó occurs, điều quỹ cho vay hy suất từ chứng khoán mình invested số dư và thay đó nhận được lãi borrowing tiền trên the số of tiền loan. **H:** The lending fund doesn't get all cả sự quan tâm some cases.
English	**P:** It's the truth, you fool. **H:** Everything I say is true.
French (R)	**P:** It's the truth, you fool. **H:** Tout ce que je true.
French (BB)	**P:** C'est la truth, abruti. **H:** Tout ce que je dis is true.
Hindi (R)	**P:** Yaha sach hai, you fool. **H:** Everything I say, sach hai.
Hindi (BB)	**P:** It's the truth, you fool. **H:** jo I say is true.
Chinese (R)	**P:** 傻瓜，这就是truth **H:** 我说的都是truth
Chinese (BB)	**P:** 这是 the truth, you fool. **H:** 我说的一切 true.
Vietnamese (R)	**P:** Đó là sự thật, you fool. **H:** Everything I say là đúng.
Vietnamese (BB)	**P:** It's the truth, you fool. **H:** Tất I say is true.

Table 15: A comparison of code-mixed examples produced by bilinguals (R) and BUMBLEBEE (BB). Since the humans were only provided with the English examples, some differences in phrasing is to be expected.

Model	Clean	PG$_{\text{unfilt.}}$	PG$_{\text{filt.}}$	Bumblebee	Bumblebee$_{\text{unconstr}}$
XLM-R$_{\text{large}}$	81.10	6.06	28.28	11.31	5.22
XLM-R$_{\text{base}}$	75.42	2.17	12.27	5.08	1.47
mBERT$_{\text{base}}$	67.54	2.15	9.24	6.10	1.19
Unicoder$_{\text{base}}$	74.98	1.99	11.33	4.81	1.29

Table 16: PolyGloss (PG) and Bumblebee results on the XNLI-13 test set with a beam width of 1. PG$_{\{\text{fil., unfilt.}\}}$ indicates whether the candidate substitutions were filtered using reference translations. Bumblebee$_{\text{unconstr}}$ refers to the setting without the equivalence constraint (§3.2).



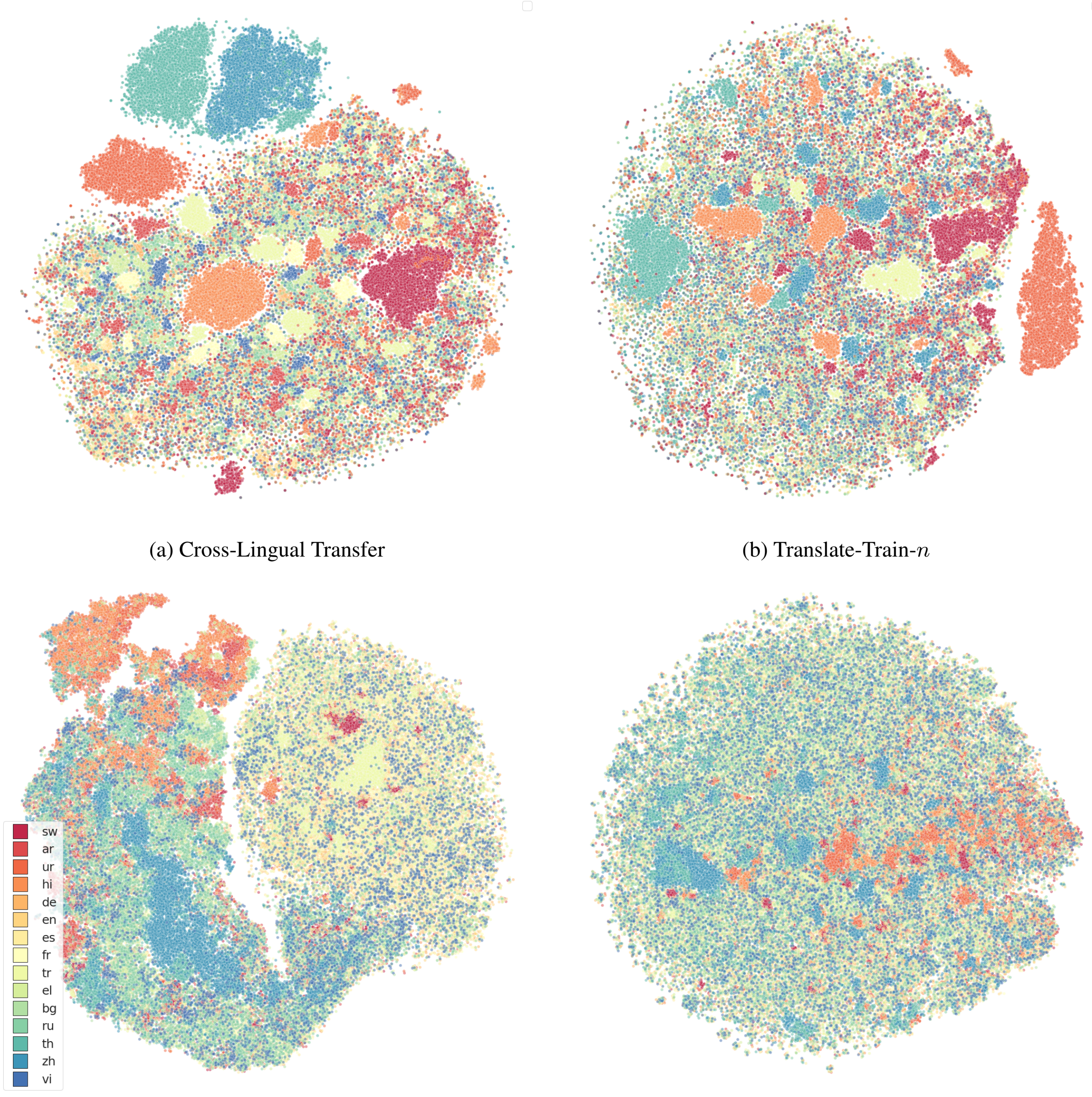
(a) Cross-Lingual Transfer

(b) Translate-Train-n

(c) Domain Adversarial Neural Network

(d) Code-Mixed Adversarial Training

Figure 4: t-SNE visualizations of XLM-R$_{\text{base}}$ representations fine-tuned using different methods. We use Linderman et al. (2019)'s t-SNE algorithm implemented in openTSNE (Poličar et al., 2019) and tried to arrange related languages close to each other on the color spectrum (to the extent possible on one dimension) so it would be obvious if similar languages were getting clustered together, as in (c).



(a) Cross-Lingual Transfer

(b) Translate-Train-n

(c) Domain Adversarial Neural Network

(d) Code-Mixed Adversarial Training

Figure 5: t-SNE visualizations of XLM-R$_{\text{base}}$ representations fine-tuned using different methods. Here, we average all the token embeddings in the sentence instead of just using the <s> embedding.