Offensive Content Detection Via Synthetic Code-Switched Text

Anonymous ACL submission

Abstract

The prevalent use of offensive content in social media has become an important reason for concern for online platforms (customer service chat-boxes, and social media platforms). Classifying offensive and hate-speech content in online settings is an essential task in many applications that needs to be addressed accordingly. However, online text from online platforms can contain code-switching, a combination of more than one language. The non-availability of labeled code-switched data for a low-resourced code-switching combinations adds difficulty to this problem. To overcome this, we release a synthetic code-switched textual dataset containing around 29k samples for training and a real-world dataset containing around 10k samples for testing for three language combinations en-fr, en-es, and en-de[1]. In this paper, we describe our algorithm for creating synthetic code-switched offensive content data and the process for creating the human generated data. We also introduce the results of a keyword classification baseline and a multi-lingual transformer based classification model.

1 Introduction

The use of offensive content in online settings such as chat-boxes, and social media platforms continues to be a growing problem that requires addressing. It can have negative effects on the psycho-emotional state of people (Saha et al., 2019). Offensive content and hate-speech continue to be a challenge to people world. As such, it is important to keep social media and other communication platforms free from offensive content. Considerable research has been conducted on deep-learning techniques for detecting offensive language (Pitsilis et al., 2018; Mehra and Hasanuzzaman, 2020). One of the growing challenges in the field of content detection is code-switching (Aguilar et al., 2020; Qin et al., 2020; Tang et al., 2020; Chakravarthi et al., 2020). Code-switching refers to the use of two or more languages in a single conversation. Code-switching can occur inter-sententially (across sentences) and intra-sententially (within sentences). The combination of code-switching and offensive content increases the complexity of the classification task. As code-switching is a combination of multiple languages, resources for these various combinations are extremely low. This causes researchers to find ways to create viable synthetic data that can serve in place of real-world data for training purposes. However, the real world benchmark test set still remains scarce.

To stimulate the research, we create human-annotated testsets written in three pairs of languages (en-fr, en-es, and en-de). Further, we propose a method for creating a synthetic train set and show its applicability to detect human-annotated code-switched text.

2 Related Works

Researchers have attempted to solve the problem of synthetic data generation for various code-switching tasks.

Theory Based Synthetic Code-switching Data Generation: Equivalency Theory (EC Theory) explains a range of interesting code-switched patterns beyond lexical substitution. The EC Theory describes a CM sentence as a constrained combination of two sentences that are equivalent. Pratapa et al. (2018) use EC Theory to generate meaningful artificial code-switched sentences.

Code-switched Offensive Content Datasets: Code-switching produces low resourced language combinations which presents many challenges for researchers in this field. Jose et al. (2020b) conducts a survey on currently available data-sets for various nlp tasks for code-switching. They mention data-sets for code-switching shared tasks Jose et al. (2020a), named entity recognition Singh et al. (2018), sentiment analysis Sitaram et al. (2015),

[1]Dataset/code links are removed for anonymity

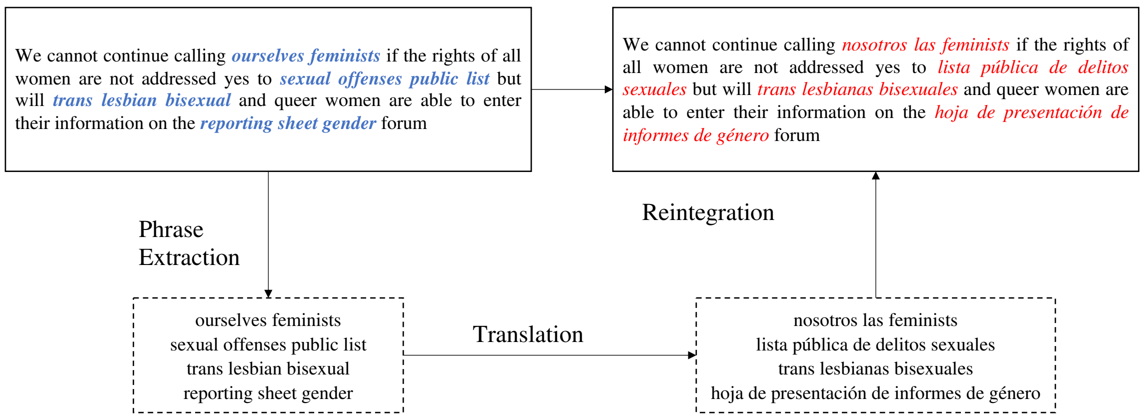

We cannot continue calling ourselves feminists if the rights of all women are not addressed yes to sexual offenses public list but will trans lesbian bisexual and queer women are able to enter their information on the reporting sheet gender forum
Phrase Extraction
ourselves feminists
sexual offenses public list
trans lesbian bisexual
reporting sheet gender
Translation
nosotros las feminists
lista pública de delitos sexuales
trans lesbianas bisexuales
hoja de presentación de informes de género
Reintegration
We cannot continue calling nosotros las feminists if the rights of all women are not addressed yes to lista pública de delitos sexuales but will trans lesbianas bisexuales and queer women are able to enter their information on the hoja de presentación de informes de género forum

Figure 1: Synthetic Code Switching Generation Framework

conversational systems Banerjee et al. (2018), machine translation (Dhar et al., 2018). While these data-sets offer a resource for code-switching related tasks, there still is a significant shortage in available data for tasks based on data for our code-switching combinations used in this research (en-fr,en-es,en-de). Our synthetic and human generated data-sets offer a set of resources to address this shortage.

For the task of Hate/Offensive speech detection, various approaches have been utilized. Dadu and Pant (2020) propose splitting a code-switched sentence into its constituent high resource languages to exploit both monolingual and cross-lingual settings. Kapoor et al. (2018) utilize apply transfer learning and design an LSTM based model of hate speech detection for hate speech and offensive speech in Hinglish (Hindi-english). Work performed on code-switched Tamil-English and Malayalam-English text includes corpus created for sentiment analysis for these two languages (Gupta et al., 2021).

However, these approaches to synthetic data generation and classification require a larger amount of data to work effectively. Our work seeks to implement an algorithm for creating synthetic code-switched data and testing the efficacy of using that synthetic data for fine-tuning a multi-lingual language model for binary offensive content detection.

3 Dataset

The language combinations produced from code-switching can be seen as low-resourced languages in their own right. To stimulate research in this domain and directly tackle the code-switched abusive language detection task, we create and release around 10k sentence test-set created by human annotators. Additionally, we generate and release around 30k synthetic dataset to train a model (see the statistics in Table 1).

	EN-FR	EN-ES	EN-DE
synthetic train-set			
Number of samples	9,926	9,926	9,926
Average length (word-level)	27.55	27.07	28.71
human-generated SWAP test-set			
Number of samples	1,080	1,564	1,668
Average length (word-level)	21.53	24.01	25.10
human-generated REWRITE test-set			
Number of samples	2,000	2,000	1,948
Average length (word-level)	16.99	16.57	15.69

Table 1: Statistics of the synthetic train-set and human-generated dataset. Annotators are asked to rewrite the code-mixed sentence (REWRITE) or only translate the pre-detected abusive word (SWAP).

3.1 Synthetic dataset generation

The overall flow of our synthetic data creation framework can be seen in Fig. 1. Our synthetic code-switching generation framework occurs in three stages.

Phrase Identification The first stage in generating the synthetic data is the identification of phrases in the mono-lingual source text. We analyze existing real word abusive speech datasets, which are written in mixed languages (Bohra et al., 2018; Patwa et al., 2020) and find that one of the salient patterns is switching "noun phrase" in the sentence (Couto and Gullberg, 2019; Dorota et al., 2021). To specify the salient phrases in the sentences, we employ a pretrained language model-based phrase tagging method (Gu et al., 2021). The original texts are passed into the tagging model to generate spans corresponding to the phrases in the sentences. Sentences that are not tagged with phrases are dis-

Type	Original text	Annotation	Label
SWAP (en/de)	backpedaling fails to point out exactly my comments and resumes to nonsensical aussie retarded babbles go choke on a cock you useless wrinkly aussie slag you are the descendants of the bottom feeders among limeys loll	backpedalismus versagt, um genau meine Kommentare und Wiederholungen zu unsinnigen aussie retarded babbles gehen auf einem Hahn Sie nutzlos faltig aussie Schlampe wurgen Sie sind die Nachkommen der Bottom Feeder unter limeys loll	1
SWAP (en/fr)	if your humor is based on racism homosexualphobia sexism and rape you are not fucking funny go home	if your humor is based on racism homosexual phobia sexism and rape, no eres jodidamente divertido, go home	1
REWRITE (en/es)	This little fake ass meeting is making me anxious	This little fake culo meeting is making me anxious	1
REWRITE (en/fr)	I thought I was the only one who noticed his ass be saying a whole lot of nothing	I thought I was the only one who noticed que son cul ne disait pas grand-chose	1

Table 2: Examples of human-generated code-mixed text. In SWAP, the highlight shows the target code switches.

carded from the dataset.

Phrase Translation Each phrase tagged in a sentence is then translated using the automatic machine translation model. We employ EasyNMT (Fan et al., 2021), an open-source state-of-the-art neural machine translation model that can translate 100+ languages. The phrases are fed into the translation model and translated to the destination language of our choice.

Phrase Reintegration After the phrases have been translated into the destination language of our choice, we then replace the tagged phrases in the source text with the new translated phrases. After the phrases have been reintegrated into the source text, the synthetically generated is ready to be utilized for training purposes.

To test the efficacy of our synthetic data generation framework, we first generate three hate-speech code-switching combinations, English-French (EN-FR), English-Spanish (EN-ES), and English-German (EN-DE). The source text for this data is HateXplain (Mathew et al., 2021), a dataset of binary (hate or not) labeled hate-speech sentences sourced from the internet. We use the training subset of this data to generate our training data synthetically. Statistics of the synthetic data created can be seen in Table 1.

3.2 Real dataset creation

Creating a benchmark test-set is an essential task for this study since it can stimulate the research further. To make the benchmark test reflect the real-world usage, we build the dataset from monolingual hate speech data created from real user text. We first take HateXplain (Mathew et al., 2021) data, which has fine-grained labels indicating the span related to the abusiveness. We request annotators to change given English sentences into a mix of English and the national language of their country (German, French, or Spanish) by focusing the switching on a provided list of offensive words. If the sentence has no offensive words, we request the annotator to create the mixed version at their own discretion.

To have diverse benchmark data-sets, we further create a test-set by asking annotators to rewrite a code-mixed text from the existing abusive text. We request annotators to REWRITE given sentences (Mandl et al., 2020) into a mix between English and their secondary language (German, French, Spanish). We ask annotators to maintain hateful/offensive translations as much as possible.

We utilize MTurk[2] and Upwork[3] platforms for SWAP and REWRITE tasks, respectively to work with bi-lingual annotators and translators to generate diverse code-switched sentences. Table 2 shows examples of the input and output from the workers (We provide further information in Appendix A).

4 Method

We employ a human-annotated lexicon dictionary for abusive language and build a binary classification model as a baseline model. Furthermore, we explore the performance of the recently proposed multilingual neural network-based model.

4.1 Baseline model

We leverage offensive and abusive speech lexicons sourced from (Hatebase) to develop a keyword-based classification algorithm. Specifically, we compiled four dictionary lexicons of hate-speech

[2]https://www.mturk.com/
[3]https://www.upwork.com/

Model	SWAP testset						REWRITE testset					
	Eng-FR		Eng-ES		Eng-DE		Eng-FR		Eng-ES		Eng-DE	
	f1	WA	f1	WA	f1	WA	f1	WA	f1	WA	f1	WA
Dictionary	0.290	0.300	**0.540**	**0.570**	0.460	0.460	**0.660**	**0.680**	**0.670**	**0.690**	0.370	0.510
XML-R$_{syn}$	**0.550**	**0.580**	0.530	0.550	**0.670**	**0.670**	0.530	0.580	0.590	0.620	**0.580**	**0.610**

Table 3: Experimental results on the benchmark testset. Each model is trained with synthetic dataset.

Model	Eng-FR		Eng-ES		Eng-DE	
	f1	WA	f1	WA	f1	WA
XML-R$_{syn}$	0.530	0.580	0.590	0.620	**0.580**	**0.610**
XML-R$_{SWAP}$	**0.610**	**0.620**	0.520	0.530	0.430	0.510
XML-R$_{syn+SWAP}$	0.580	**0.620**	**0.600**	**0.630**	**0.580**	**0.610**

Table 4: Model is trained with synthetic, SWAP, or synthetic+SWAP and evaluated on REWRITE testset.

words from each language present (English, French, German, and Spanish). Each lexicon is used as a look-up table to determine if words present in given sentences are considered hate/offensive or not.

4.2 Transformer Based Model

To leverage the pretrained language model (PLM), we employ a multilingual model, XLM-RoBERTa (XLM-R), and build the abusive content clasifier (Conneau et al., 2019)[4]. In implementing the model, we feed the code-switched sentence to the XLM-R, and the "[CLS]" token is further passed through a two-layer fully-connected network. The final output is compared with the label, and loss is computed using the cross-entropy function (We provide more details in Appendix B).

5 Experimental results and discussion

To fine-tune the XML-R model, we perform a learning rate schedule. We base the scheduling on the validation split macro F1 scores instead of using the loss from the validation. We adopt this approach from (Roy et al., 2021) where the authors focus on the validation scores at the end of each training iteration instead of using early-stopping to prevent over-fitting. If the validation performance decreases through an iteration, we backtrack to the previous model weights and decrease our learning rate. Training ends when the learning rate reaches a significantly small value. Even though this approach is extremely expensive, this type of scheduling guarantees that the Macro F1 score is maximized on the validation split.

We ran three types of experiments for both our dictionary and transformer-based models; (1) training on a synthetic dataset and testing on SWAP/REWRITE datasets, (2) training on SWAP, and testing on REWRITE, (3) training on synthetic and SWAP datasets, and testing on REWRITE. Table 3 and Table 4 show the experimental results in terms of F1 score and weighted accuracy (WA). An interesting observation in our experiment is the different results on our SWAP & REWRITE testsets. For instance, when code-switching semantics tend towards the swapping of offensive words between languages (SWAP testset), an PLM trained on our synthetic can perform better than dictionary-based detection (EN-DE). This is primarily due to the fact that our synthetic data generation algorithm is most similar to these types of occurrences. We also find that our synthetic dataset shows strong utility even better than human-annotated data (see Table 4). In other cases, we can see a decrease in performance when the structure of the code-switched sentences is more complex.

Based on some of these observations, we believe this algorithm can be useful in extending model training sets by mixing both synthetic data with real-world training data.

6 Conclusion

In this paper, we introduced an algorithm for generating synthetic code-switched data for training purposes. Using this algorithm, we generated a synthetic offensive-content dataset comprised of 30k entries for en-fr, en-de, en-es code-switching combinations. Additionally, we release two human-annotated testsets of the under-resourced en-fr, en-de, en-es language combinations (approximately 10k). We create two baselines models and report their performance on the human-annotated testsets. We expect this resource will enable the researchers to address new and exciting problems in code-mixed research.

[4]We also test other variants of multilingual models such as multilingual BERT (Devlin et al., 2019) and multilingual-DistilBERT (Sanh et al., 2019) on the downstream tasks and find the XLMR consistently shows superior performance.

References

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. Lince: A centralized benchmark for linguistic code-switching evaluation. *arXiv preprint arXiv:2005.04322*.

Suman Banerjee, Nikita Moghe, Siddharth Arora, and Mitesh M. Khapra. 2018. A dataset for building code-mixed goal oriented conversation systems. In *COLING*.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset of Hindi-English code-mixed social media text for hate speech detection. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 36–41, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John P McCrae. 2020. Corpus creation for sentiment analysis in code-mixed tamil-english text. *arXiv preprint arXiv:2006.00206*.

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

Maria Carmen Parafita Couto and Marianne Gullberg. 2019. Code-switching within the noun phrase: Evidence from three corpora. *International Journal of Bilingualism*, 23(2):695–714.

Tanvi Dadu and Kartikey Pant. 2020. Towards code-switched classification exploiting constituent language resources. *arXiv preprint arXiv:2011.01913*.

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.

Mrinal Dhar, Vaibhav Kumar, and Manish Shrivastava. 2018. Enabling code-mixed translation: Parallel corpus creation and mt augmentation approach.

Gaskins Dorota, Oksana Bailleul, Anne Marie Werner, and Antje Endesfelder Quick. 2021. A crosslinguistic study of child code-switching within the noun phrase: A usage-based perspective. *Languages*, 6(1).

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

Xiaotao Gu, Zihan Wang, Zhenyu Bi, Yu Meng, Liyuan Liu, Jiawei Han, and Jingbo Shang. 2021. Ucphrase: Unsupervised context-aware quality phrase tagging. *arXiv preprint arXiv:2105.14078*.

Akshat Gupta, Sargam Menghani, Sai Krishna Rallabandi, and Alan W. Black. 2021. Unsupervised self-training for sentiment analysis of code-switched data. *CoRR*, abs/2103.14797.

Hatebase. Hatebase, a collaborative, regionalized repository of multilingual hate speech. Accessed: 2021-10-29.

Navya Jose, Bharathi Chakravarthi, Shardul Suryawanshi, Elizabeth Sherly, and John McCrae. 2020a. A survey of current datasets for code-switching research. pages 136–141.

Navya Jose, Bharathi Raja Chakravarthi, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. 2020b. A survey of current datasets for code-switching research. *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 136–141.

Raghav Kapoor, Yaman Kumar, Kshitij Rajput, Rajiv Ratn Shah, Ponnurangam Kumaraguru, and Roger Zimmermann. 2018. Mind your language: Abuse and offense detection for code-switched languages. *CoRR*, abs/1809.08652.

Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. 2020. Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german. In *Forum for Information Retrieval Evaluation*, pages 29–32.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14867–14875.

Sidharth Mehra and Mohammed Hasanuzzaman. 2020. *Detection of Offensive Language in Social Media Posts*. Ph.D. thesis.

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. *CoRR*, abs/2008.04277.

Georgios K. Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Detecting offensive language in tweets using deep learning. *CoRR*, abs/1801.04433.

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.

Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP. *CoRR*, abs/2006.06402.

Sayar Ghosh Roy, Ujwal Narayan, Tathagata Raha, Zubair Abid, and Vasudeva Varma. 2021. Leveraging multilingual transformers for hate speech detection. *CoRR*, abs/2101.03207.

Koustuv Saha, eshwar chandrasekharan, and Munmun Choudhury. 2019. Prevalence and psychological effects of hateful speech in online college communities. volume 2019.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Named entity recognition for hindi-english code-mixed social media text. In *NEWS@ACL*.

Dinkar Sitaram, Savitha Murthy, Debraj Ray, Devansh Sharma, and Kashyap Dhar. 2015. Sentiment analysis of mixed language employing hindi-english code switching. *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, 1:271–276.

Tiancheng Tang, Xinhuai Tang, and Tianyi Yuan. 2020. Fine-tuning bert for multi-label sentiment analysis in unbalanced code-switching text. *IEEE Access*, 8:193248–193256.

A Data Collection

A.1 Sentence Generation

We generate code-switched sentences from the test-split of HateXplain and HASOC. The test-split of the HateXplain data-set contains sentences with words tagged by annotators that convey hate-speech and offensive content. These sentences and words are given to code-switching annotators on Amazon Mechanical Turk (Mturk) platform to perform the SWAP method as described in section 3.2. HASOC sentences do not contain annotated hate and offensive words and so this data is sent to bi-lingual translators on the Upwork platform.

MTurk is a crowdsourcing marketplace that simplifies the outsourcing of tasks to a distributed workforce who can perform these tasks virtually. Mturk allows individuals and businesses to post batches of assignments for workers.

On Upwork, three job posting are created with the following criteria:

- Fluency in English & (German, French or Spanish)
- Familiarity with colloquial terminology

Freelancers are then chosen based on the above criteria. The freelancers perform the REWRITE method of sentence generation as described in section 3.2.

A.2 Instructions to Annotators

Annotators for both the SWAP & REWRITE methods, are given instructions on how to complete the annotation tasks. An example of the SWAP annotation instructions and an example of a task on mturk can be seen in Fig. 2 and Fig. 3 respectively.

For SWAP, we request annotators to change a given English sentence into a mix of English and their native language (German, French, or Spanish) by focusing the switching on the provide list of words that are pre-determined to be hateful or offensive. If the sentence provided is not offensive, we then request that the annotator create a mixed version of the sentence based on their own discretion.

For REWRITE, we request annotators to rewrite the given sentences into a mix between English and their native language (German, French, or Spanish) based on their own discretion. We ask the annotators to maintain hateful or offensive translations as much as possible.

A.3 Workers Pool & Pay

For MTurk, we hire the annotators whose locations is either France, Germany, Mexico, Spain. This restriction of location helps to ensure the annotators speak both the national language of the country as well as English We restrict the workers whose HIT approval rates are higher than 95%. We pay workers around 12 USD per hour.

For Upwork, we hire translators who are professionally fluent in either German, French, or Spanish. We choose the translators who best showcase the ability to create a code-switched rewrite by rewriting a few test examples. Each translator is paid according to a negotiated fee based on the number of sentences to REWRITE. We pay annotators 10 USD per 30 sentences, which is above the average rate for a similar task on Upwork.

B Reproducibility Checklist

- **Source code with specification of all dependencies, including external libraries**: The source code is included in the submission. It provides information about the dependencies including external libraries and instructions on how to run the proposed models.

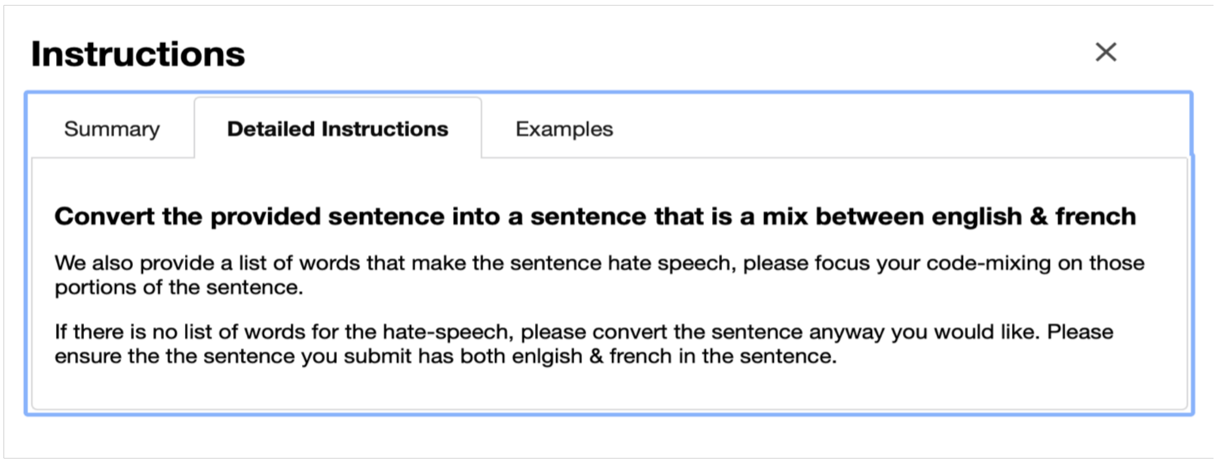
- **Description of computing infrastructure used**: We use a single Tesla V100 GPU with 16GB memory in this work. PyTorch 1.1 is used to implement the models.

- **Average run-time for each approach**: Each epoch of the XLMR models, on average, takes 2 minutes for binary offensive classification. We train the model until learning rate reaches an very small value.

- **Number of parameters in the model**: We use XLMR in our in our experiments. This model has 2.7 million parameters to be optimized during training.

- **Explanation of evaluation metrics used**: To evaluate the performance of the model, we use the the weighted average and F1 scores for prediction.

- **Hyper-parameter configurations for best-performing models**: Our model has 768 hidden layers. The Adamw optimizer learning rate is set to 2e-5 and the batch size is 16.


Instructions

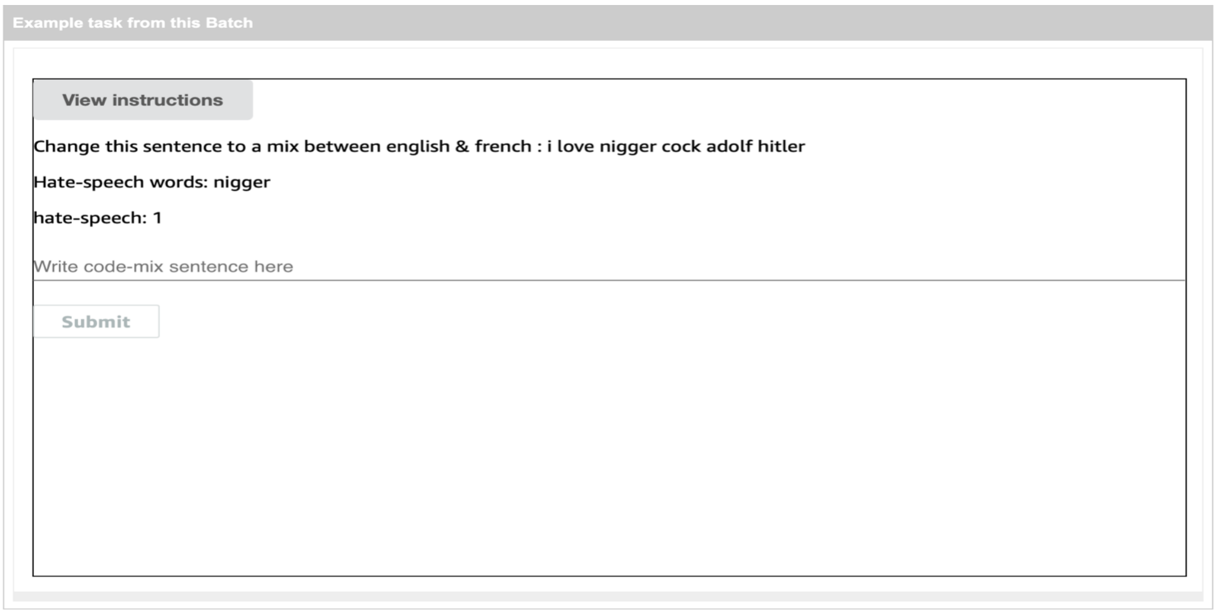
Summary | Detailed Instructions | Examples

Convert the provided sentence into a sentence that is a mix between english & french

We also provide a list of words that make the sentence hate speech, please focus your code-mixing on those portions of the sentence.

If there is no list of words for the hate-speech, please convert the sentence anyway you would like. Please ensure the the sentence you submit has both enlgish & french in the sentence.

Figure 2: Annotator instructions on SWAP task


Example task from this Batch

View instructions

Change this sentence to a mix between english & french : i love nigger cock adolf hitler

Hate-speech words: nigger

hate-speech: 1

Write code-mix sentence here

Submit

Figure 3: Interface for human code-switching annotation task for SWAP method

- **The method of choosing hyper-parameter values and the criterion used to select among them**: Random search is used to determine the hyper-parameters. The selection is determined F1 scores and the selected hyper-params are used across experiments for uniformity.