Making Sentence Embeddings Robust to User-Generated Content

Lydia Nishimwe, Benoît Sagot, Rachel Bawden

Making Sentence Embeddings Robust to User-Generated Content

Lydia Nishimwe, Benoît Sagot, Rachel Bawden
Inria
2 rue Simone Iff, 75012 Paris, France
{firstname.lastname}@inria.fr

Abstract

NLP models have been known to perform poorly on user-generated content (UGC), mainly because it presents a lot of lexical variations and deviates from the standard texts on which most of these models were trained. In this work, we focus on the robustness of LASER, a sentence embedding model, to UGC data. We evaluate this robustness by LASER's ability to represent non-standard sentences and their standard counterparts close to each other in the embedding space. Inspired by previous works extending LASER to other languages and modalities, we propose RoLASER, a robust English encoder trained using a teacher-student approach to reduce the distances between the representations of standard and UGC sentences. We show that with training only on standard and synthetic UGC-like data, RoLASER significantly improves LASER's robustness to both natural and artificial UGC data by achieving up to $2\times$ and $11\times$ better scores. We also perform a fine-grained analysis on artificial UGC data and find that our model greatly outperforms LASER on its most challenging UGC phenomena such as keyboard typos and social media abbreviations. Evaluation on downstream tasks shows that RoLASER performs comparably to or better than LASER on standard data, while consistently outperforming it on UGC data.

Keywords: sentence embeddings, robustness, user-generated content (UGC)

1. Introduction

Most Natural Language Processing (NLP) models are trained on "standard" texts, which are edited and well written. When applied to user-generated content (UGC), these models struggle due to the high lexical variance induced by the presence of "non-standard" phenomena such as irregular spelling choices, evolving slang and marks of expressiveness (Seddah et al., 2012; Eisenstein, 2013; van der Goot et al., 2018; Sanguinetti et al., 2020). Table 1 illustrates some examples of non-standard sentences with their standardised versions. UGC has been shown to have a negative impact on NLP model performance in various tasks such as machine translation (Belinkov and Bisk, 2018; Rosales Núñez et al., 2021a), dependency parsing (van der Goot, 2019), sentiment analysis (Kumar et al., 2020) and named entity recognition (Plank et al., 2020).

This performance drop of NLP models is due to their semantic vector representations (or *embeddings*) not being robust to UGC, *i.e.* non-standard words and their standard counterparts do not have similar embeddings, even if they have the same meaning in the same context. Furthermore, common UGC phenomena such as acronyms (*e.g. btw* $\rightarrow$ *by the way*) and misspellings can greatly modify the tokenisation of a sentence, making it hard to represent the tokens of a UGC sentence and its normalised version in the same space. Therefore, we propose to tackle the problem at the sentence level: we consider each sentence as a whole and aim for a robust embedding that is not as affected by local, surface-level lexical variations. We frame the question of robustness to UGC as a bitext alignment problem in the sentence embedding space: *how well can a sentence encoder align a standard text with its non-standard counterpart and how close are the two sentences in the embedding space?*

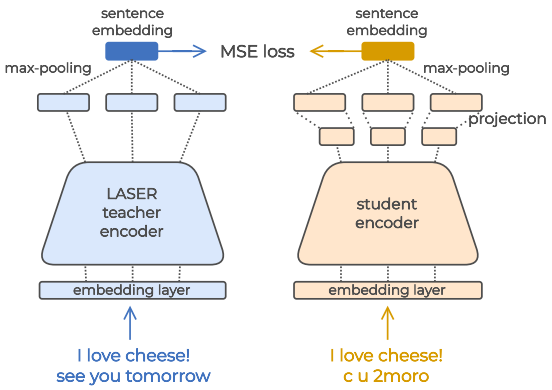

Figure 1: Teacher-Student approach.

Inspired by previous works extending the LASER sentence encoder (Artetxe and Schwenk, 2019b) to low-resource languages and the speech modality (Heffernan et al., 2022; Duquenne et al., 2022), our approach is to train a student of LASER which learns to map non-standard English sentences and their standard versions close to each other in the embedding space (see Figure 1). We compare two model architectures (one token-level and one character-aware), trained using artificially generated parallel UGC data, and we use popular bitext

Corpus	UGC sentence	Standard(ised) sentence
MultiLexNorm$^{\diamond}$	if i **cnt** afford the real deal , i ain't buying **nuffin** fake .. i just won't have it	if i **can't** afford the real deal , i ain't buying **nothing** fake .. i just won't have it
RoCS-MT‡	**Umm idk**, maybe **its bc** we're **DIFFERENT PEOPLE** with **DIFFERENT BODIES???**	**Um, I don't know**, maybe **it's because** we're **different people** with **different bodies?**
FLORES† `abr2 + fing + abr1`	**"** Luckily **nthing** happened **2** me **,** but I saw a macabre scene **,** as **ppl triwd 2** break windows in order **2 gt** out **.**	**"**Luckily **nothing** happened **to** me**,** but I saw a macabre scene**,** as **people tried to** break windows in order **to get** out**.**

Table 1: Example non-standard sentences from 3 different UGC corpora and their standardised versions. $\diamond$: Twitter, ‡: Reddit, †: artificially augmented with UGC phenomena.

mining metrics for intrinsic evaluation. We also conduct an analysis of the robustness of LASER and the student models to natural and artificial UGC data in general and to each UGC phenomenon type. Finally, we analyse the performance of the models on standard data and downstream tasks such as sentence (pair) classification and semantic textual similarity.

With our robust English LASER encoder, we open the door to cross-lingual and cross-modal NLP applications on UGC data, thanks to LASER being multilingual, flexible and modular (Duquenne et al., 2022).

Our main contributions are:

1. a simple method to increase sentence-level encoder robustness to UGC by reducing the standard-UGC distance in the embedding space;
2. RoLASER, a LASER student encoder for English more robust to natural and artificial UGC, as well as c-RoLASER, its character-aware equivalent;
3. a fine-grained analysis of model robustness to artificial UGC data by UGC phenomenon type;
4. a simple combination of data augmentation techniques for generating artificial real-life-like UGC for training and evaluation in scenarios where natural parallel UGC data is scarce.

We release our models and code at `https://github.com/lydianish/RoLASER`.

2. Background and Related Work

Language-Agnostic SEntence Representations (LASER) One of the pioneers of large-scale multilingual sentence embedding models, LASER has known many improvements over time. The first LASER model (Artetxe and Schwenk, 2019b) was a multilingual bi-LSTM (Schuster and Paliwal, 1997) encoder-decoder model that was trained using a machine translation objective on 93 languages, pooling the encoder's outputs to obtain a fixed-size sentence embedding. Li and Mak (2020) proposed T-LASER, a version of LASER built on the Transformer architecture (Vaswani et al., 2017) and added a distance constraint to the translation loss to bring parallel sentences closer in the embedding space. After releasing LASER2, which presents some improvements with respect to the original LASER model, Heffernan et al. (2022) observed that one of the major problems with it was the poor representation of low-resource languages in the multilingual sentence space. In order to tackle it, they used a teacher-student approach inspired by knowledge distillation (Hinton et al., 2015) to train Transformer-based encoders (student models) on monolingual and parallel xx→English data to mimic the behaviour of LASER2 (the teacher model). Each of these students, called LASER3, targeted a specific low-resource language. Duquenne et al. (2022) built on this approach to build Translation Modules (T-Modules) for multilingual cross-modal translation. They trained speech and text encoders to learn from LASER2, and also trained decoders from the LASER embedding space. Tan et al. (2023) proposed LASER3-CO, a variant of LASER3 that integrates contrastive learning. In our work, we adapt the teacher-student approach for UGC English using a similar training setup to T-Modules, particularly the training loss.

Improving Model Robustness to UGC Data One solution to recover the performance drop of NLP models is to train or fine-tune them on UGC data. However, the scarcity of parallel annotated UGC data poses a problem. For instance, most available datasets for training or evaluating the machine translation of UGC contain only a few thousand bitexts, *e.g.* MTNT (Michel and Neubig, 2018), PFSMB (Rosales Núñez et al., 2019) and RoCS-MT (Bawden and Sagot, 2023). To mitigate this, data augmentation techniques have been explored to generate synthetic UGC training data. In particular, rule-based techniques consisting of character- and word-level edit operations and perturbations,

as well as dictionary-based techniques, have been used to improve the robustness of NLP models to synthetic and natural UGC (Belinkov and Bisk, 2018; Karpukhin et al., 2019; Matos Veliz et al., 2019; Dekker and van der Goot, 2020; Samuel and Straka, 2021). In our work, we combine various types of such transformations to generate synthetic UGC data from standard data. We also analyse performance by UGC phenomenon type, similarly to Rosales Núñez et al. (2021a). Data augmentation has also been used to train monolingual sentence models with a focus on improving the separation between similar and dissimilar sentences in the embedding space (Yan et al., 2021; Chuang et al., 2022; Tang et al., 2022). Our work, however, aims to bring closer UGC sentences and their standard counterparts on the basis that they are in fact similar. Other works have also shown that character-level models can be more robust to non-standard data in such low-resource scenarios (Rosales Núñez et al., 2021b; Riabi et al., 2021; Libovický et al., 2022), which motivates us to also explore using a student model with a character-level input embedding layer.

3. Proposed Approach: Reducing the Standard-UGC Distance in the Embedding Space

We propose to train a sentence embedding model that is robust to non-standard UGC text, such that the representation assigned to non-standard text is as close as possible to its normalised equivalent without degrading model performance on standard text. We choose to work with the LASER model and aim therefore to encode non-standard text into the LASER embedding space. Although we evaluate on English in this article, this also leaves open the possibility in the future of working with other languages (for which LASER representations are also available).

Inspired by the teacher-student approach in LASER3 (Heffernan et al., 2022) and T-Modules (Duquenne et al., 2022), we train a student model on standard English and UGC English data with LASER2 as the teacher (see Figure 1). The training loss is a mean-squared error (MSE) loss, and the student model learns to minimise the distance between the two output sentence embedding vectors. As a result, it makes both standard and non-standard sentences as close as possible to the teacher's standard embeddings. This should, in theory, make it more robust to UGC phenomena. A similar approach has also been successfully applied to making monolingual sentence embeddings multilingual (Reimers and Gurevych, 2020).

With LASER2 as the teacher model, we separately train two student models. The first is (BPE-based) token-level with the same architecture as RoBERTa (Liu et al., 2019), which we refer to as **RoLASER** (**Ro**bust **LASER**). We also train a character-aware student for comparison. It has a similar architecture to the first one, except for the input embedding layer, which is character-level. We refer to this model as **c-RoLASER**. From this point forward, LASER will be used to refer to LASER2.

Given the scarcity of natural UGC data to train such a model, we artificially generate non-standard data from standard English sentences. We achieve this by applying selected transformations from NL-Augmenter[1] (Dhole et al., 2021), namely:[2]

- insertion of common social media abbreviations, acronyms and slang words (`abr1`, `abr2`, `abr3`, `slng`);
- contraction and expansion of auxiliary verbs (`cont`), *e.g. I am* ↔ *I'm*, and of names of months and weekdays (`week`), *e.g. Mon.* ↔ *Monday*;
- insertion of misspellings such as keyboard typos or "butter fingers" (`fing`); homophone (`homo`) and dyslexia (`dysl`) errors, *e.g. there* ↔ *their, lose* ↔ *loose*; and other common spelling mistakes (`spel`);
- visual and segmentation transformations such as Leet Speak[3] (`leet`), *e.g. love* → *l0V3*; and whitespace insertion and deletion (`spac`).

We also define a `mix_all` transformation that randomly selects and applies a subset of the previous perturbations. For example, the last UGC sentence in Table 1 was obtained via a `mix_all` transformation which applied `abr2`, `fing` and `abr1` to a standard sentence.

4. Evaluating Robustness

Intuitively, the embedding space is robust if variants of the same sentence are embedded into vectors that are close to one another, *i.e.* they ideally have similar representations. However, although designed to be a semantic space, it is natural for non-semantic aspects of sentences to be represented in the space too (*e.g.* syntactic variations, language, formality, etc.), and for semantic equivalents therefore not to have identical embeddings. For the applications we envisage, our aim is for non-standard texts to be assigned embeddings that are as close as possible such that the surface form of the sentences does not impact the embeddings. To

[1] https://github.com/GEM-benchmark/NL-Augmenter

[2] See Appendix A for the detailed list of transformations and random generation techniques.

[3] https://en.wikipedia.org/wiki/Leet

evaluate this, we use several metrics for evaluating embeddings (Section 4.1) and several English normalisation-centric datasets, including both natural and artificial non-standardness (Section 4.2).

4.1. Evaluation Metrics

The metrics we use are pairwise cosine distance as well as xSIM and xSIM++, two metrics previously used for evaluating sentence embeddings through the proxy task of bitext mining.

Average Pairwise Cosine Distance We compute the cosine distances between the embeddings of each non-standard sentence and its normalised version and then average over all sentences in the text. For the sake of brevity, we will subsequently refer to it simply as cosine distance.

xSIM and xSIM++ Cross-lingual similarity search, or xSIM (Artetxe and Schwenk, 2019a), is a proxy metric used for bitext mining. Given a set of parallel sentences in languages A (the source) and B (the target), it aligns sentences via margin-based similarity scores. It then computes the error rate of aligning each language A sentence with its language B translation from the pool of candidates (all language B sentences). xSIM++ is an extended version of the metric that discriminates better between systems and correlates more with performance on downstream tasks. It was proposed by Chen et al. (2023), who noted that xSIM was not challenging enough for many language pairs, given that the sentences in the candidate pool were often too semantically distinct (see Appendix B). xSIM++ relies on augmenting the target set with hard negative examples, created by applying transformations that perturb the meaning of the sentences with minimal alteration to their surface form (causality alternation, number replacement and entity replacement). Note that xSIM was initially designed to be used in conjunction with the FLORES-200 dataset (see Section 4.2) and xSIM++ only augmented the English sets (making them approximately 44 times larger). xSIM++ can therefore currently only be evaluated on xx→English language pairs from FLORES-200.

4.2. Evaluation Data

We evaluate on three English test sets representing different types of parallel non-standard data and their normalised versions.[4] We use two existing datasets of natural UGC (MultiLexNorm and RoCS-MT). However, in order to do a finer-grained analysis, we also create artificial UGC from FLORES-200 by applying multiple transformations. Examples from the three evaluation sets we use are provided in Table 1, and basic statistics are given in Table 2. Note that UGC texts tend to have fewer tokens than their standard counterparts, mainly due to the frequent use of acronyms and abbreviations. The lexical diversity of the datasets is indicated using the type-token ratio (TTR).[5]

MultiLexNorm (van der Goot et al., 2021) is a multilingual dataset created for the lexical normalisation task. We use the English subset, consisting of sentences from Twitter and their manual normalisations. The data is pretokenised and lowercased.

RoCS-MT (Bawden and Sagot, 2023) is a multilingual dataset for the task of machine translation of UGC English into other languages: Czech (cs), German (de), French (fr), Russian (ru) and Ukranian (uk). The source sentences are from Reddit, and manual normalisations are also provided. Unlike MultiLexNorm, the data is not pretokenised nor lowercased. Casing is kept intact in the original sentences, and normalised in the standard ones.

FLORES-200 (NLLB Team et al., 2022) is a multilingual dataset consisting of parallel texts from WikiNews, WikiBooks and WikiVoyage in 200 languages. We artificially transform its English subset with UGC phenomena from NL-Augmenter as described in Section 3. We subsequently refer to the original corpus as FLORES, and to the artificially augmented one as FLORES†.

5. Experimental Setup

Training Data We use 2 million standard English sentences of the unshuffled deduplicated OSCAR[6] dataset (Ortiz Suárez et al., 2019), representing 648MB of text. The data is split into 100 chunks of 20k sentences, each of which is artificially augmented with UGC phenomena using the `mix_all` transformation with probability $p_{all} = 0.1$ (described in Appendix A) and a different random seed, producing a 2M-sentence "bilingual" standard-UGC dataset. The standard sentences are passed to the teacher, while their augmented ones are passed to the student. Note that by setting a probability to

[4] In practice, the definition of *normalised* depends on the annotation guidelines chosen.

[5] The TTR is the number of unique tokens divided by the total token count; the more lexically diverse a text is, the higher the TTR. Previous work has shown that UGC texts tend to have a higher TTR due to multiple variants of the same word (Rosales Núñez et al., 2021a). We compute TTR based on LASER's SentencePiece tokenisation (Kudo and Richardson, 2018).

[6] https://huggingface.co/datasets/oscar/viewer/unshuffled_deduplicated_en

	FLORES		MultiLexNorm						RoCS-MT	
	dev	devtest	train		dev		test		test	
Metric	std	std	std	UGC	std	UGC	std	UGC	std	UGC
# sentences	997	1012	2360	2360	590	590	1967	1967	1922	1922
# tokens	36.7k	38.9k	76.1k	75.8k	19.8k	19.7k	63.3k	63.1k	43.0k	40.8k
TTR	9.10	8.82	5.98	6.06	14.49	14.71	6.86	6.95	6.34	7.16
(TTR ratio)				(1.01)		(1.02)		(1.01)		(1.13)

Table 2: Description of standard (std) and UGC data. TTR=Type-Token Ratio, TTR ratio=$\text{TTR}_{\text{UGC}}/\text{TTR}_{\text{std}}$.

apply transformations, not all sentences are augmented.[7] Furthermore, replacement-based transformations may leave the original sentence unchanged if they find no candidate words to replace. As a result, the student model also sees standard sentences and learns to encode them (Figure 1).

Text Preprocessing When fetching OSCAR data, we replace HTML line-breaking characters, do sentence splitting and filter out sentences with less than 90% of common English characters. Afterwards, we apply the same preprocessing steps as LASER on all data, namely: removal of non-printable characters, punctuation normalisation and lowercasing (Artetxe and Schwenk, 2019b). The teacher input texts are then tokenised with LASER's SentencePiece (Kudo and Richardson, 2018) model (vocabulary size 50,004), and the RoLASER student inputs using RoBERTa's SentencePiece tokeniser (vocabulary size 50,265). As for the c-RoLASER student, the inputs are pre-tokenised on whitespace and punctuation using BERT's pretokeniser (Devlin et al., 2019).[8]

Architectures LASER[9] is a 45M-parameter encoder with 5 bi-LSTM layers and an output embedding dimension of 1,024. RoLASER is a 108M-parameter, 12-layer Transformer encoder with 12 attention heads and a 768-output dimension, similarly to RoBERTa (without the final pooling layer). c-RoLASER is a 104M-parameter encoder with the same architecture as RoLASER except for the input embedding layer, which is a Character-CNN similar to the one used in CharacterBERT (El Boukkouri et al., 2020). Note that the students' output dimension is smaller than LASER's. Therefore, similarly to Mao and Nakagawa (2023), we add a linear layer to the student encoders to project their outputs to the right size. The outputs from the teacher and students are then max-pooled to obtain sentence embedding vectors. Regarding the pooling strategy, Duquenne et al. (2022) showed that max-pooling works better than CLS-pooling for LASER students, probably because LASER itself was trained with max-pooling. While many teacher-student sentence embedding models use mean-pooling (Reimers and Gurevych, 2020; Ham and Kim, 2021; Mao and Nakagawa, 2023), our preliminary experiments showed that max-pooling consistently performs slightly better than mean-pooling during validation. All model implementation and training are done using the Fairseq toolkit (Ott et al., 2019).

Training The teacher model remains frozen during training. Both student models are separately trained on 8 Tesla V100-SXM2 GPUs with a maximum number of 4,000 tokens per batch per GPU (without gradient accumulation); an Adam optimiser with parameters $\beta = (0.9, 0.98)$ and $\epsilon = 10^{-6}$; learning rates of 10^{-4} for RoLASER and 5×10^{-5} for c-RoLASER, both with 1,000 warm-up updates; standard, attention and activation dropouts of 0.1; and a clip norm of 5. Similarly to T-Modules (Duquenne et al., 2022), the training criterion is encoder similarity, and the training loss is an MSE loss with sum reduction. A checkpoint is saved every 30,000 steps. Our preliminary experiments also showed that initialising the student with a pre-trained language model performed better during validation than random initialisation. We therefore initialise RoLASER with RoBERTa,[10] and c-RoLASER with CharacterBERT.[11] Table 3 describes further details of the training checkpoints.

Model	#Params.	#Epochs	#Steps	#Hours
RoLASER	108M	100	683k	86
		98	**669k**	
c-RoLASER	104M	34	750k	170
		32	**726k**	

Table 3: Training details of student models. Best checkpoints are in **bold**. Trained on 8 GPUs.

[7] In our case, 563,343 sentences ($\approx 28.2\%$) are not transformed (see Figure 3 in Appendix A).

[8] https://huggingface.co/google-bert/bert-base-cased

[9] https://github.com/facebookresearch/LASER

[10] https://huggingface.co/FacebookAI/roberta-base

[11] https://huggingface.co/helboukkouri/character-bert

Validation The best checkpoint is selected by taking the student model that minimises the MSE distance between the teacher's representation of standard text and the student's representations of (i) standard text and (ii) UGC text, *i.e.*:

$$\text{loss} = \text{MSE}(L[std], m[std]) + \text{MSE}(L[std], m[ugc]),$$

where $L[x]$ and $m[x]$ refer respectively to the teacher and student's representation of x, where x can either be standard (std) or UGC (ugc) text. Framing it as a sum of two losses allows us to monitor the model's learning to minimise both distances with respect to the same anchor, using the sentence triplet $(L[std], m[std], m[ugc])$. This is different from the training loss which minimises both distances separately, *i.e.* via two separate sentence pairs $(L[std_1], m[std_1])$ and $(L[std_2], m[ugc_2])$. For each saved checkpoint, we compute the validation loss on the dev set of FLORES (which is also augmented with the `mix_all` transformation) and select the checkpoint with the lowest loss.[12]

6. Results and Analysis

We evaluate LASER, RoLASER and c-RoLASER on the MultiLexNorm and RoCS-MT test sets. We also generate artificial data by applying each of the UGC transformations described in Section 3 to the standard FLORES devtest 10 times with different generation seeds, and we evaluate the models on the generated FLORES† sets. We first conduct an intrinsic evaluation of the student models' robustness in Section 6.1 where we analyse whether the student models are better at representing UGC data compared to LASER, and whether they are as good as LASER on standard English. We then conduct an extrinsic evaluation in Section 6.2 where we analyse their performance on downstream tasks such as sentence (pair) classification and semantic textual similarity.

6.1. Intrinsic Evaluation

In theory, a sentence embedding model would be robust to UGC if the cosine distance between standard and non-standard sentence pairs is small enough to ensure a perfect similarity alignment score. In practice, we aim to reduce cosine distances and similarity alignment error rates scores as much as possible. For each model m, we evaluate whether the distance between $m[ugc]$ and $m[std]$ has effectively reduced, and whether that translates into lower search error rates. We perform xSIM (and xSIM++ for FLORES) on UGC→standard English bitexts. We determine the statistical significance of the student model results using an independent 2-sample t-test compared to LASER's scores. We also compute the TTR of generated FLORES† files to gauge their non-standardness level, as well as their t-test compared to the TTR of the original FLORES text, and we indicate the TTR ratio with respect to the standard text. We report results on natural test sets in Section 6.1.1, results on the artificial test sets for each UGC phenomenon type in Section 6.1.2, and on standard data in Section 6.1.3.

	MultiLexNorm		RoCS-MT	
Model	cos dist	xSIM	cos dist	xSIM
LASER	0.03	0.10	0.09	4.06
RoLASER	0.02	**0.05**	0.06	**2.34**
(improv.)		(2.0×)		(1.7×)
c-RoLASER	**0.01**	0.10	**0.05**	3.80
(improv.)		(1.0×)		(1.1×)

Table 4: Cosine distance and xSIM scores on UGC→standard English bitexts from natural UGC test sets. The best score for each metric is in **bold**.

6.1.1. Results on Natural UGC

Table 4 illustrates the cosine distance and xSIM scores of the three models on UGC→standard English bitexts from the MultiLexNorm and RoCS-MT test sets. We observe that both student models reduce the cosine distance across the board. We also note that RoCS-MT is a more challenging evaluation set as it produces much greater distances and error scores, which is consistent with it having the highest TTR ratio (Table 2). While RoLASER outperforms LASER with $\approx 2\times$ better xSIM scores, we observe a contradictory tendency for c-RoLASER: despite having the lowest cosine distances, it produces minimal performance gains over LASER. We will show in Section 6.1.3 that this is because c-RoLASER has, on average, larger distances between its standard embeddings and LASER's.

To visually compare the students' and LASER's sentence representations, we use a 2-component PCA dimension reduction of the LASER sentence space. In Figure 2, we plot the embeddings of the UGC sentence *"I then lost interest in her bc her IG wasn't that interesting."* from RoCS-MT,[13] its normalised version *"I then lost interest in her, because her Instagram wasn't that interesting."*, and its translations in five other languages. We evaluate the distance preservation in the reduced dimensions and obtain a Spearman's correlation of $r = 0.69$ between Euclidean distances in the reduced and original space. We observe that

[12] We use a different random seed from the ones selected for augmenting the training set.

[13] We choose this example because it illustrates the trends observed on the RoCS-MT test set.

both RoLASER and c-RoLASER have a shorter standard-UGC distance than LASER. Furthermore, RoLASER's standard and UGC embeddings are closer to LASER's than any of the other languages. However, c-RoLASER's standard embedding remains far from LASER's, which explains its poor xSIM scores.

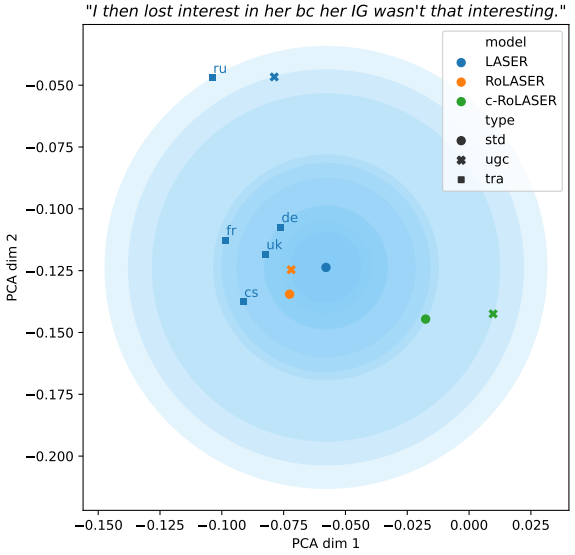

Figure 2: Visualisation of the first 2 principal components of the LASER space. The points represent the embeddings of a UGC sentence from RoCS-MT, its standardised version (std), and its translations into other languages (tra).

6.1.2. Results by UGC Phenomenon Type

Table 5 illustrates the cosine distance and the xSIM++ scores of the three models on the FLORES† devtest for all UGC types, as well as the ratio of TTRs of the UGC texts with respect to the standard text.[14] We report the xSIM results in Table 9 (Appendix C). All of LASER's xSIM++ scores are (highly) significantly[15] different from zero (the expected mean), suggesting a lack of robustness of LASER to artificial UGC types.

We observe that both student models have highly significantly reduced the cosine distances to close to zero. We also note that cosine distance scores for LASER on most UGC types are less than 0.07, which is the minimum, or 0th percentile, for LASER on all xx→English FLORES language pairs (see Figure 5 in Appendix B). In other words, LASER mostly represents UGC English closer to standard English than it does all the other languages, which is reasonable considering UGC English is still English. The UGC type that it embeds the furthest from standard English is `leet` with a cosine distance of 0.22, which is in the 35th percentile. This means LASER considers that 35% of FLORES languages are closer to standard English than Leet Speak English.

With xSIM++, the three most challenging transformations for LASER are `leet`, `space` and `fing`. Intuitively, they are the ones that "shatter" subword tokenisation the most because they perform character-level perturbations. In fact, `leet` and `fing` have the lowest and highest TTR ratios respectively. The next batch of challenging transformations apply more word-level perturbations (`abr2` and `homo`). Other noteworthy transformations are the ones with very low xSIM++ scores and cosine distances of zero: `abr3`, `cont`, `week`. Finally, the results suggest that `mix_all` is challenging enough to be a good attempt at generating comprehensive, real-life-like artificial UGC.

RoLASER outperforms LASER artificial UGC (as shown by the $10.8\times$ better xSIM++ score on `mix_all`). We also observe major performance gains for several UGC transformations: $22.7\times$ better for `leet`, and between $3.9\times$ and $10.6\times$ for most of the other types. `fing` remains the most challenging one for RoLASER as it obtains the highest cosine distance and xSIM++ score (which is still $2.8\times$ better than LASER's). Lastly, RoLASER slightly degrades LASER's performance on `cont` and `abr3`. This is likely because these phenomena are already frequent in standard data, which means that the original LASER has already been trained to deal with them efficiently. It could also be that they perform minimal perturbations on the original text (as shown by their TTR ratio of 1.00).

We also note that all the RoLASER xSIM++ scores are less than 7.21%, which is the minimum score for LASER on all xx→English FLORES language pairs (see Figure 6 in Appendix B). It is akin to saying that RoLASER aligns UGC English to standard English better than LASER does all the other languages.

However, the c-RoLASER results are disappointing: it degrades the performance on all types, except for `leet`, `fing` and `spac`, and it never outperforms RoLASER. This is consistent with the results on natural UGC that c-RoLASER struggles to map its standard embeddings to LASER's.

6.1.3. Results on Standard Data

It is also important to evaluate whether the student models' reduced UGC-standard distances introduce a performance drop on standard data. In theory, this should not be the case since they are also trained to minimise the distance between their standard embeddings $m[std]$ on the one hand, and LASER's standard embeddings $L[std]$ on the other.

[14] See Figure 4 in Appendix A for more interpretation of the TTR ratios.

[15] significant: $p < 0.05$, highly significant: $p < 0.001$.

UGC type (TTR ratio)	abr1 (0.93**)	abr2 (0.98**)	abr3 (1.00**)	cont (1.00**)	dysl (0.99**)	fing (1.05**)	homo (0.98**)	leet (0.76**)	slng (0.98**)	spac (1.01*)	spel (1.02**)	week (1.00**)	mix_all (1.01**)
Average pairwise cosine distance													
LASER	0.03	0.08	**0.00**	**0.00**	0.04	0.07	0.05	0.22	0.02	0.09	0.04	**0.00**	0.05
RoLASER	**0.00****	**0.00****	**0.00**	**0.00**	**0.00****	0.02**	**0.00****	0.01**	**0.00****	**0.01****	**0.01****	**0.00**	**0.00****
c-RoLASER	**0.00****	0.01**	**0.00**	**0.00**	0.01**	**0.01****	0.01**	**0.00****	0.01**	0.03**	**0.01****	**0.00**	0.01**
xSIM++													
LASER	4.01	15.81	**0.10**	**0.20**	5.83	19.60	10.13	68.60	2.39	22.76	7.85	2.08	13.17
RoLASER	**0.75***	**1.58****	0.40	0.40	**0.79****	**7.09***	**0.96****	**3.03****	**0.61****	**2.45****	**2.14****	**0.40****	**1.22****
(improv.)	(5.4×)	(10.0×)			(7.4×)	(2.8×)	(10.6×)	(22.7×)	(3.9×)	(9.3×)	(3.7×)	(5.2×)	(10.8×)
c-RoLASER	13.16	19.37	11.76	12.94	15.71	13.49	16.93	12.56**	14.54	19.72	15.00	11.46	15.74
(improv.)						(1.5×)		(5.5×)		(1.2×)			

Table 5: Cosine distance and xSIM++ scores for all models on UGC→standard English bitext from each UGC type of FLORES† devtest, averaged across 10 data generation seeds. The best score for each type is in **bold**. *: $p < 0.05$, **: $p < 0.001$, statistical significance with respect to LASER's scores.

	cs→en		de→en		fr→en		ru→en		uk→en	
Model [en]	UGC	std	UGC	std	UGC	std	UGC	std	UGC	std
LASER	9.11	**3.28**	6.56	0.83	10.20	**4.68**	11.76	5.93	8.79	2.39
RoLASER	**7.23**	3.33	**4.94**	**0.73**	**9.21**	4.79	**10.15**	**4.89**	**6.61**	**2.34**
c-RoLASER	13.94	7.49	9.78	4.37	15.04	9.31	16.44	10.87	13.53	7.13
	en→cs		en→de		en→fr		en→ru		en→uk	
Model [en]	UGC	std	UGC	std	UGC	std	UGC	std	UGC	std
LASER	9.11	**2.71**	5.83	**0.57**	10.87	**5.10**	11.71	**5.88**	8.79	2.55
RoLASER	**7.02**	3.17	**4.58**	0.83	**8.64**	5.20	**10.20**	6.35	**6.04**	**2.34**
c-RoLASER	18.26	9.16	13.16	5.52	19.25	11.76	23.31	14.36	17.74	8.58

Table 6: xSIM scores on xx→English and English→xx bitexts from RoCS-MT. The results compare all models for embedding UGC and standard (std) English. Only LASER is used to embed the non-English languages. The best score for each language pair is in **bold**.

We evaluate all models on the task of bitext alignment on the five xx-English language pairs of RoCS-MT. Table 6 shows the xSIM scores in both xx→English and English→xx directions,[16] where English is either UGC or standard (std). LASER is used to embed all non-English sentences, while both LASER and the student models are used for the English sentences.

As is expected, standard English consistently produces better results than UGC for all the models. We also observe that RoLASER improves on LASER's performance for standard English in the xx→English direction. This is likely because the student specialised in standard English as the target language during training. In the English→xx direction however, RoLASER only surpasses LASER about half the time. As for UGC English, we observe that RoLASER produces the best results in both directions, while c-RoLASER degrades LASER's performance.

To better understand these results, we compare the standard English embeddings from the student models with LASER's on all test sets. We illustrate in Table 7 the average pairwise cosine distance between them. They show that RoLASER has managed to effectively minimise the distance between its standard embeddings and LASER's, which manifests as performance gains observed in the bilingual alignments (Table 6). However, c-RoLASER struggles to map its standard embeddings to LASER's, especially on RoCS-MT. This explains its poor performance in general. In other words, c-RoLASER has successfully reduced the distance between its UGC and standard embeddings to almost zero (see Table 5), but it lags behind when bridging the gap between its standard embeddings and LASER's. One reason for this could be that character-level tokenisation results in very long

Model	FLORES	MultiLexNorm	RoCS-MT
RoLASER	0.02	0.04	0.05
c-RoLASER	0.05	0.09	0.13

Table 7: Cosine distance between the students' and LASER's standard embeddings.

[16] xSIM is not symmetrical: scores are not comparable across both language pair directions (Chen et al., 2023).

sequences, making it a difficult task for the model pool their representations into one fixed-sized vector capturing all semantic information. Nonetheless, we suspect that c-RoLASER could benefit from longer and better optimised training.

6.2. Extrinsic Evaluation

To support the results of the intrinsic evaluation (Section 6.1), we evaluate our models' performance on downstream tasks from MTEB, the Massive Text Embedding Benchmark (Muennighoff et al., 2023). We select four tasks spanning three types:

1. **Sentence classification**, which predicts labels from sentence embeddings, *e.g.* sentiment labels: `TweetSentimentExtractionClassification` (T-SentExt).
2. **Sentence pair classification**, which predicts a binary label from sentence embeddings, *e.g.* whether two sentences are paraphrases: `TwitterSemEval2015` (T-SemEval) and `TwitterURLCorpus` (T-URL).
3. **Semantic textual similarity**, which examines the degree of semantic equivalence between two sentences: `STSBenchmark` (STS).

Note that the first three tasks are evaluated on UGC, specifically Twitter data. The last one is evaluated on more standard texts from image captions, news headlines and user forums.

Model	T-SentExt$^{\diamond}$	T-SemEval†	T-URL†	STS‡
LASER	50.64	59.57	81.48	**69.77**
RoLASER	**51.96**	**60.68**	**81.79**	69.61
c-RoLASER	49.29	55.32	76.80	68.13

Table 8: Scores (%) on 4 MTEB tasks. The best score for each metric is in **bold**. $\diamond$: accuracy, †: average precision on cosine similarity, ‡: Spearman's correlation on cosine similarity.

Table 8 shows the scores of our models on the four tasks (along with their corresponding evaluation metrics). RoLASER consistently outperforms LASER on the first three tasks on Twitter data, while it is almost as good as LASER on the standard STS task. This is in agreement with our findings in Section 6.1 that RoLASER is better than LASER at encoding non-standard data and achieves comparable performance on standard data. On the other hand, c-RoLASER remains the worst across all tasks and greatly degrades LASER's performance.

7. Conclusion

In this work, we frame the question of LASER's robustness to UGC as a bitext alignment problem where we aim to align standard sentences and their non-standard equivalents. We propose RoLASER, a Transformer-based encoder student of LASER, trained with the objective of minimising the distances between standard and non-standard sentence pairs in the embedding space. The model is trained solely on standard and synthetic UGC-like English data. We also consider a character-aware student, c-RoLASER, and find that the token-level RoLASER performs best overall while the c-RoLASER struggles to map its standard embeddings to LASER's.

We find that RoLASER is significantly more robust than LASER on natural UGC, achieving up to $2\times$ better xSIM scores. We also evaluate it on standard data and downstream tasks and show that it improves, or at least matches, LASER's performance. Furthermore, we perform a fine-grained analysis of the models' robustness with respect to artificially generated data by type of UGC phenomena. We show that RoLASER achieves roughly $11\times$ better xSIM++ scores than LASER on artificial UGC, and up to $23\times$ better on Leet Speak, the most difficult UGC type for LASER. We also find that the most challenging phenomena are those with character-level perturbations that shatter subword tokenisation.

For future work, we plan to extend RoLASER to more languages and their corresponding UGC phenomena. We will also consider ways to improve c-RoLASER, such as using a thin-deep architecture (Tay et al., 2022), or a token-level model with a small enough vocabulary size to be close to the character level.

8. Limitations

The ambiguity introduced by non-standard words in language could be problematic. For example, *smh* could mean *shaking my head* or *so much hate*, and our approach would try to map both to the same space. One way to resolve this ambiguity would be to use the surrounding sentences as context. Though it is an interesting line of research to pursue, it is outside the scope of this article. Thankfully, such cases are rare and the model has proved to do well in general across multiple UGC types.

There is also a possible domain mismatch between the type of data used to train the our models and the data on which we test. RoLASER is trained and validated on standard data artificially augmented with UGC phenomena and is evaluated on (scarce) parallel UGC data from social media. However, the results show that the model is able to generalise well on natural UGC data without having been trained or fine-tuned on it.

9. Acknowledgements

We thank the anonymous reviewers for their constructive feedback, and Paul-Ambroise Duquenne for his insights on LASER. This work was granted access to the HPC resources of IDRIS under the allocations 2023-AD011012254R1 and 2023-AD011013674R1 made by GENCI. This work was funded by the last two authors' chairs in the PRAIRIE institute funded by the French national agency ANR as part of the "Investissements d'avenir" programme under the reference ANR-19-P3IA-0001.

10. Bibliographical References

Mikel Artetxe and Holger Schwenk. 2019a. Margin-based Parallel Corpus Mining with Multilingual Sentence Embeddings. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019b. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. In Transactions of the Association for Computational Linguistics, volume 7, pages 597–610, Cambridge, MA. MIT Press.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada.

Mingda Chen, Kevin Heffernan, Onur Çelebi, Alexandre Mourachko, and Holger Schwenk. 2023. xSIM++: An improved proxy to bitext mining performance for low-resource languages. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 101–109, Toronto, Canada. Association for Computational Linguistics.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4207–4218, Seattle, United States. Association for Computational Linguistics.

Kelly Dekker and Rob van der Goot. 2020. Synthetic data for English lexical normalization: How close can we get to manually annotated data? In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 6300–6309, Marseille, France. European Language Resources Association.

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.

Kaustubh D. Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, Saad Mahamood, Abinaya Mahendiran, Simon Mille, Ashish Srivastava, Samson Tan, Tongshuang Wu, Jascha Sohl-Dickstein, Jinho D. Choi, Eduard Hovy, Ondrej Dusek, Sebastian Ruder, Sajant Anand, Nagender Aneja, Rabin Banjade, Lisa Barthe, Hanna Behnke, Ian Berlot-Attwell, Connor Boyle, Caroline Brun, Marco Antonio Sobrevilla Cabezudo, Samuel Cahyawijaya, Emile Chapuis, Wanxiang Che, Mukund Choudhary, Christian Clauss, Pierre Colombo, Filip Cornell, Gautier Dagan, Mayukh Das, Tanay Dixit, Thomas Dopierre, Paul-Alexis Dray, Suchitra Dubey, Tatiana Ekeinhor, Marco Di Giovanni, Rishabh Gupta, Rishabh Gupta, Louanes Hamla, Sang Han, Fabrice Harel-Canada, Antoine Honore, Ishan Jindal, Przemyslaw K. Joniak, Denis Kleyko, Venelin Kovatchev, Kalpesh Krishna, Ashutosh Kumar, Stefan Langer, Seungjae Ryan Lee, Corey James Levinson, Hualou Liang, Kaizhao Liang, Zhexiong Liu, Andrey Lukyanenko, Vukosi Marivate, Gerard de Melo, Simon Meoni, Maxime Meyer, Afnan Mir, Nafise Sadat Moosavi, Niklas Muennighoff, Timothy Sum Hon Mun, Kenton Murray, Marcin Namysl, Maria Obedkova, Priti Oli, Nivranshu Pasricha, Jan Pfister, Richard Plant, Vinay Prabhu, Vasile Pais, Libo Qin, Shahab Raji, Pawan Kumar Rajpoot, Vikas Raunak, Roy Rinberg, Nicolas Roberts, Juan Diego Rodriguez, Claude Roux, Vasconcellos P. H. S., Ananya B. Sai, Robin M. Schmidt, Thomas Scialom, Tshephisho Sefara, Saqib N. Shamsi, Xudong Shen, Haoyue Shi, Yiwen Shi, Anna Shvets, Nick Siegel, Damien Sileo, Jamie Simon, Chandan Singh, Roman Sitelew, Priyank Soni, Taylor Sorensen, William Soto, Aman Srivastava, KV Aditya Srivatsa, Tony Sun, Mukund Varma T, A Tabassum, Fiona Anting Tan, Ryan Teehan, Mo Tiwari, Marie Tolkiehn,

Athena Wang, Zijian Wang, Gloria Wang, Zijie J. Wang, Fuxuan Wei, Bryan Wilie, Genta Indra Winata, Xinyi Wu, Witold Wydmański, Tianbao Xie, Usama Yaseen, M. Yee, Jing Zhang, and Yue Zhang. 2021. NL-Augmenter: A Framework for Task-Sensitive Natural Language Augmentation.

Paul-Ambroise Duquenne, Hongyu Gong, Benoît Sagot, and Holger Schwenk. 2022. T-Modules: Translation Modules for Zero-Shot Cross-Modal Machine Translation. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 5794–5806, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 359–369, Atlanta, Georgia. Association for Computational Linguistics.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun'ichi Tsujii. 2020. CharacterBERT: Reconciling ELMo and BERT for Word-Level Open-Vocabulary Representations From Characters. In Proceedings of the 28th International Conference on Computational Linguistics, pages 6903–6915, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jiyeon Ham and Eun-Sol Kim. 2021. Semantic alignment with calibrated similarity for multilingual sentence embedding. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 1781–1791, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext Mining Using Distilled Sentence Representations for Low-Resource Languages. In Findings of the Association for Computational Linguistics: EMNLP 2022, pages 2101–2112, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. In Proceedings of the 2014 NIPS Deep Learning and Representation Learning Workshop, Montreal, Quebec, Canada.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on Synthetic Noise Improves Robustness to Natural Noise in Machine Translation. In Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), pages 42–47, Hong Kong, China. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Ankit Kumar, Piyush Makhija, and Anuj Gupta. 2020. Noisy Text Data: Achilles' Heel of BERT. In Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020), pages 16–21, Online. Association for Computational Linguistics.

Wei Li and Brian Mak. 2020. Transformer based multilingual document embedding model. CoRR, abs/2008.08567.

Jindřich Libovický, Helmut Schmid, and Alexander Fraser. 2022. Why don't people use character-level machine translation? In Findings of the Association for Computational Linguistics: ACL 2022, pages 2470–2485, Dublin, Ireland. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692.

Zhuoyuan Mao and Tetsuji Nakagawa. 2023. LEALLA: Learning Lightweight Language-agnostic Sentence Embeddings with Knowledge Distillation. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 1886–1894, Dubrovnik, Croatia. Association for Computational Linguistics.

Claudia Matos Veliz, Orphee De Clercq, and Veronique Hoste. 2019. Benefits of Data Augmentation for NMT-based Text Normalization of User-Generated Content. In Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), pages 275–285, Hong Kong, China. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In

Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Barbara Plank, Kristian Nørgaard Jensen, and Rob van der Goot. 2020. DaN+: Danish nested named entities and lexical normalization. In Proceedings of the 28th International Conference on Computational Linguistics, pages 6649–6662, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4512–4525, Online. Association for Computational Linguistics.

Arij Riabi, Benoît Sagot, and Djamé Seddah. 2021. Can Character-based Language Models Improve Downstream Task Performances In Low-Resource And Noisy Language Scenarios? In Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021), pages 423–436, Online. Association for Computational Linguistics.

José Carlos Rosales Núñez, Djamé Seddah, and Guillaume Wisniewski. 2021a. Understanding the Impact of UGC Specificities on Translation Quality. In Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021), pages 189–198, Online. Association for Computational Linguistics.

José Carlos Rosales Núñez, Guillaume Wisniewski, and Djamé Seddah. 2021b. Noisy UGC translation at the character level: Revisiting open-vocabulary capabilities and robustness of char-based models. In Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021), pages 199–211, Online. Association for Computational Linguistics.

David Samuel and Milan Straka. 2021. ÚFAL at MultiLexNorm 2021: Improving Multilingual Lexical Normalization by Fine-tuning ByT5. In Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021), pages 483–492, Online. Association for Computational Linguistics.

Manuela Sanguinetti, Cristina Bosco, Lauren Cassidy, Özlem Çetinoğlu, Alessandra Teresa Cignarella, Teresa Lynn, Ines Rehbein, Josef Ruppenhofer, Djamé Seddah, and Amir Zeldes. 2020. Treebanking user-generated content: A proposal for a unified representation in Universal Dependencies. In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 5240–5250, Marseille, France. European Language Resources Association.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11):2673–2681.

Djamé Seddah, Benoit Sagot, Marie Candito, Virginie Mouilleron, and Vanessa Combet. 2012. The French Social Media Bank: a treebank of noisy user generated content. In Proceedings of COLING 2012, pages 2441–2458, Mumbai, India. The COLING 2012 Organizing Committee.

Weiting Tan, Kevin Heffernan, Holger Schwenk, and Philipp Koehn. 2023. Multilingual representation distillation with contrastive learning. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 1477–1490, Dubrovnik, Croatia. Association for Computational Linguistics.

Zilu Tang, Muhammed Yusuf Kocyigit, and Derry Tanti Wijaya. 2022. AugCSE: Contrastive Sentence Embedding with Diverse Augmentations. In Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 375–398, Online only. Association for Computational Linguistics.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Prakash Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2022. Charformer: Fast character transformers via gradient-based subword tokenization. In Proceedings of the The Tenth International Conference on Learning Representations, Virtual.

Rob van der Goot. 2019. An In-depth Analysis of the Effect of Lexical Normalization on the Dependency Parsing of Social Media. In Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), pages 115–120, Hong Kong, China. Association for Computational Linguistics.

Rob van der Goot, Rik van Noord, and Gertjan van Noord. 2018. A Taxonomy for In-depth Evaluation of Normalization for User Generated Content. In International Conference on

Language Resources and Evaluation, Miyazaki, Japan. European Language Resources Association (ELRA).

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

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. CoRR, abs/2212.03533.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 5065–5075, Online. Association for Computational Linguistics.

11. Language Resource References

Bawden, Rachel and Sagot, Benoît. 2023. RoCS-MT: Robustness Challenge Set for Machine Translation. Association for Computational Linguistics.

Michel, Paul and Neubig, Graham. 2018. MTNT: A Testbed for Machine Translation of Noisy Text. Association for Computational Linguistics.

Muennighoff, Niklas and Tazi, Nouamane and Magne, Loic and Reimers, Nils. 2023. MTEB: Massive Text Embedding Benchmark. Association for Computational Linguistics.

NLLB Team and Costa-jussà, Marta R. and Cross, James and Çelebi, Onur and Elbayad, Maha and Heafield, Kenneth and Heffernan, Kevin and Kalbassi, Elahe and Lam, Janice and Licht, Daniel and Maillard, Jean and Sun, Anna and Wang, Skyler and Wenzek, Guillaume and Youngblood, Al and Akula, Bapi and Barrault, Loïc and Mejia Gonzalez, Gabriel and Hansanti, Prangthip and Hoffman, John and Jarrett, Semarley and Ram Sadagopan, Kaushik and Rowe, Dirk and Spruit, Shannon and Tran, Chau and Andrews, Pierre and Ayan, Necip Fazil and Bhosale, Shruti and Edunov, Sergey and Fan, Angela and Gao, Cynthia and Goswami, Vedanuj and Guzmán, Francisco and Koehn, Philipp and Mourachko, Alexandre and Ropers, Christophe and Saleem, Safiyyah and Schwenk, Holger and Wang, Jeff. 2022. No Language Left Behind: Scaling Human-Centered Machine Translation. FAIR (META).

Ortiz Suárez, Pedro Javier and Sagot, Benoît and Romary, Laurent. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Leibniz-Institut für Deutsche Sprache.

Rosales Núñez, José Carlos and Seddah, Djamé and Wisniewski, Guillaume. 2019. Comparison between NMT and PBSMT Performance for Translating Noisy User-Generated Content. Linköping University Electronic Press.

van der Goot, Rob and Ramponi, Alan and Zubiaga, Arkaitz and Plank, Barbara and Muller, Benjamin and San Vicente Roncal, Iñaki and Ljubešić, Nikola and Çetinoğlu, Özlem and Mahendra, Rahmad and Çolakoğlu, Talha and Baldwin, Timothy and Caselli, Tommaso and Sidorenko, Wladimir. 2021. MultiLexNorm: A Shared Task on Multilingual Lexical Normalization. Association for Computational Linguistics.

Appendices

A. Transformations for Artificial UGC Generation

Below is the detailed list of transformations selected from NL-Augmenter for artificial UGC generation.

1. `abr1` *(abbreviation_transformation)*:[17] replaces words or phrases with their abbreviated counterpart using a web-scraped slang dictionary (with default probability $p = 0.1$)
2. `abr2` *(insert_abbreviation)*: replaces words or phrases with their abbreviated counterpart from a list of common generic and social media abbreviations
3. `abr3` *(replace_abbreviation_and_acronyms)*: swaps the abbreviated and expanded forms of words and phrases from a list of common abbreviations and acronyms in business communications
4. `cont` *(contraction_expansions)*: swaps commonly used contractions and expansions, *e.g. I am* $\leftrightarrow$ *I'm*
5. `dysl` *(dyslexia_words_swap)*: replaces words with their counterparts from a list of frequently misspelled words for dyslexia, *e.g. lose* $\leftrightarrow$ *loose*

[17] Name of the transformation module in NL-Augmenter.

6. `fing` *(butter_fingers_perturbation)*: swaps letters with one of their QWERTY keyboard neighbours ($p = 0.05$)

7. `homo` *(close_homophones_swap)*: replaces words with one of their homophones ($p = 0.5$), *e.g. there* ↔ *their*

8. `leet` *(leet_letters)*: replaces letters with their Leet[18] equivalents ($p = 0.1$), *e.g. love* → *l0V3*

9. `slng` *(slangificator)*: replaces words (in particular, nouns, adjectives, and adverbs) with their corresponding slang from a dictionary of English slang and colloquialisms

10. `spac` *(whitespace_perturbation)*: adds or remove a whitespace at random positions ($p_{add} = 0.05, p_{remove} = 0.1$)

11. `spel` *(replace_spelling)*: replaces words with their counterparts from corpora of frequently misspelled words ($p = 0.2$)

12. `week` *(weekday_month_abbreviation)*: abbreviates or expands the names of months and weekdays, *e.g. Mon.* ↔ *Monday*

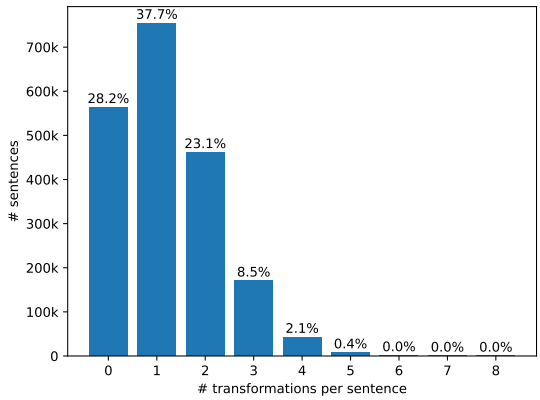

Figure 3: Distribution of transformations obtained by applying `mix_all` on 2M training sentences.

We also implement a `mix_all` transformation that combines perturbations from some of the 12 transformations. Firstly, a subset of the transformations is uniformly selected with probability $p_{all} = 0.1$. Then they are shuffled, ensuring that they are not always applied in the same order. Lastly, for the transformations that depend on a probability parameter p, let p_d denote its default value. The value of p is randomly selected between $\{\frac{1}{2}p_d, p_d, \frac{3}{2}p_d\}$, with probabilities of $\{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$ respectively. A different random seed is used for each transformation. Figure 3 illustrates the distribution of the number of perturbations applied to each sentence as a result of executing the `mix_all` transformation on 2 million training sentences from the OSCAR dataset.

[18] https://en.wikipedia.org/wiki/Leet

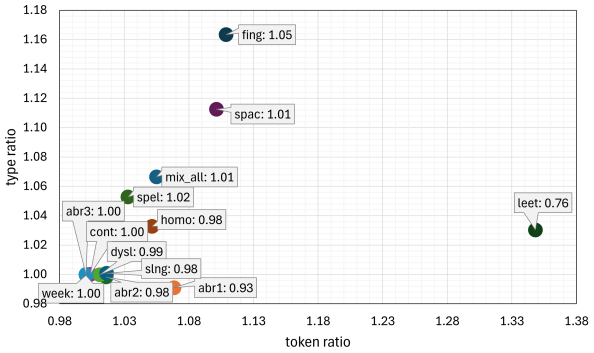

Figure 4: Visualisation of UGC phenomena of the FLORES† devtest by their type and token ratios. The data point labels indicate TTR ratios. All ratios are with respect to the standard English text.

All these transformations produce artificial UGC texts with varying levels of non-standardness. Figure 4 illustrates the ratios of number of types, number of tokens and TTR of the FLORES† devtest texts generated by each transformation with respect to the standard English text. The perturbations with the highest and lowest TTR ratios are `fing` and `leet`, respectively. `fing` also has the highest type ratio while `leet` has the highest token ratio. Both transformations perform character-level substitutions that shatter LASER's SentencePiece tokenisation. `spac` also has a high type ratio as a result of inserting and deleting whitespaces. In theory, the closer a transformation is to the lower-left corner of the plot, the more standard-like the UGC text is. For instance, `abr3`, `cont` and `week` fall into this category with all three ratios equal to 1.00. Conversely, the farther the transformation is from the lower-left corner, the more non-standard it is (and therefore more challenging for LASER).

B. Comparison of Cosine Distance, xSIM and xSIM++ across Languages

The FLORES dataset has n-way parallel texts in 200 languages. We produce LASER embeddings of the devtest and compute average pairwise cosine distance, xSIM and xSIM++ for all 199 xx-English language pairs. Figure 5 shows the quantiles of the cosine distance, while Figure 6 shows those of xSIM and xSIM++. The minimum values (or 0^{th} percentiles) are 0.07, 0% and 7.21% for cosine distance, xSIM and xSIM++ respectively.

Notably, Figure 6 supports the observation made by Chen et al. (2023) that the xSIM scores for many language pairs "quickly saturate at 0%". Indeed,

UGC type	abr1	abr2	abr3	cont	dysl	fing	homo	leet	slng	spac	spel	week	mix_all
LASER	0.15	0.30	**0.00**	0.10	0.10	0.35	0.11	10.35	0.10	0.36	0.16	**0.00**	0.38
RoLASER	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00****	**0.00**	**0.00**	**0.00**	**0.00**	**0.00****
c-RoLASER	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00****	**0.00**	**0.00**	**0.00**	**0.00**	**0.00****

Table 9: xSIM scores for all models on UGC→standard English bitext from each UGC type of FLORES† devtest data, averaged across 10 data generation seeds. The best score for each type is in **bold**. **: $p < 0.001$, statistical significance with respect to LASER's scores.

the xSIM value remains at 0 until the 20% quantile (20th percentile). This means that for the top 20% language pairs, LASER has a perfect xSIM score in aligning the sentences. We see that xSIM++ is a better metric because it is not easy to get a perfect score. It is therefore deemed more "challenging".

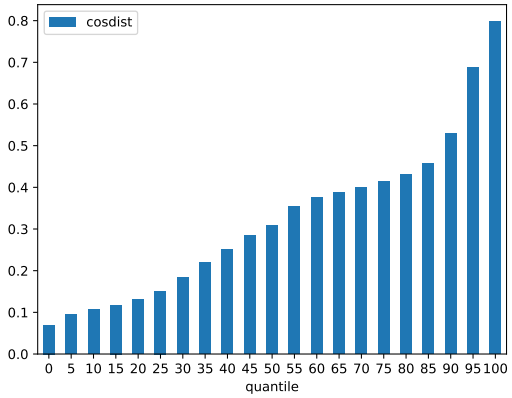

Figure 5: Quantiles of average pairwise cosine distance on FLORES devtest for all 199 xx→English language pairs.

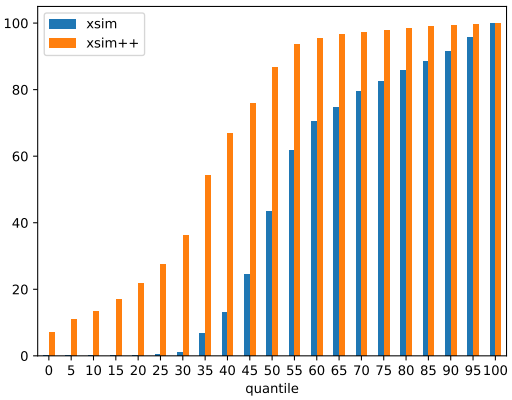

Figure 6: Quantiles of xSIM and xSIM++ scores on FLORES devtest for all 199 xx→English language pairs.

C. xSIM Scores on Artificial UGC

Table 9 shows the xSIM scores of the three models on the artificial UGC texts from FLORES† devtest. Both RoLASER and c-RoLASER get a consistent score of zero across all UGC types. As it has already been stated that xSIM is not challenging enough on FLORES (see Appendix B), these results are not informative enough to make further conclusions on their performance, other than that they improve on LASER's.