Robustification of Multilingual Language Models to Real-world Noise in Crosslingual Zero-shot Settings with Robust Contrastive Pretraining

Asa Cooper Stickland[⋆¶†], **Sailik Sengupta**[⋆‡], **Jason Krone**[¶‡], **He He**[‡♢], **Saab Mansour**[‡]

[†]University of Edinburgh, [‡]AWS AI Labs, [♢]New York University

a.cooper.stickland@ed.ac.uk,{sailiks,saabm,hehea}@amazon.com

Abstract

Advances in neural modeling have achieved state-of-the-art (SOTA) results on public natural language processing (NLP) benchmarks, at times surpassing human performance. However, there is a gap between public benchmarks and real-world applications where *noise*, such as typographical or grammatical mistakes, is abundant and can result in degraded performance. Unfortunately, works which evaluate the robustness of neural models on noisy data and propose improvements, are limited to the English language. Upon analyzing noise in different languages, we observe that noise types vary greatly across languages. Thus, existing investigations do not generalize trivially to multilingual settings. To benchmark the performance of pretrained multilingual language models, we construct noisy datasets covering five languages and four NLP tasks and observe a clear gap in the performance between clean and noisy data in the zero-shot cross-lingual setting. After investigating several ways to boost the robustness of multilingual models in this setting, we propose Robust Contrastive Pretraining (RCP). RCP combines data augmentation with a contrastive loss term at the pretraining stage and achieves large improvements on noisy (& original test data) across two sentence-level (+3.2%) and two sequence-labeling (+10 F1-score) multilingual classification tasks.

1 Introduction

Recently, multilingual pre-trained language models like mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020) and various others (Chi et al., 2021; Xue et al., 2021; Chi et al., 2022) have improved multilingual language understanding by pretraining large Transformer models on web-scale corpora (such as Wikipedia, CommonCrawl). These models achieve state-of-the-art performance on cross-lingual transfer and many multilingual NLP tasks (Wu and Dredze, 2019; Pires et al., 2019). However, a real-world system will encounter *real-world noise*, such as linguistic variations and common errors observed in textual data, that are often absent from benchmark datasets.

While prior works focused on this issue of robustness in monolingual settings (Peng et al., 2021; Sengupta et al., 2021; Tan et al., 2020), investigation has been scarce for multilingual settings. In this paper, we study the effect of realistic noise in multilingual settings and propose methods to boost the robustness of multilingual language models across four NLP tasks: Intent Classification (IC), Slot Labeling (SL), Named Entity Recognition (NER) and Natural Language Inference (NLI).

Due to the lack of multilingual noisy *evaluation* data, we synthesize benchmarks by mining noise from publicly available corpora and injecting them into the test sets associated with each of the four tasks. We conduct human validation to ensure that this noised data is indeed realistic (see examples from MultiATIS++ in Figure 1) and identify the variety of noise-types seen across languages (in §3). These analyses highlight the potential of our test-set in evaluating (and motivating future research on) multilingual robustness.

To benchmark the performance of multilingual systems, we consider accuracy metrics on two utterance-level tasks (IC% and NLI%) and F1-scores on two token-level classification tasks (SL-F1 and NER-F1). Specifically, we seek to evaluate the model's performance on the noised version of the test datasets in a *zero-shot cross-lingual setting*. In this scenario, we have training data for a task available only in one language (in our case, English) and test-data in various languages (Liu et al., 2019, 2020).

While training data augmentation increases model robustness for monolingual (i.e. English) settings, it is not immediately obvious if these robustness gains can transfer across languages, as error types can often be language-specific. For ex-

[⋆] Equal Contribution, [¶] Work done while at Amazon.

Language	Noise Injection Ratio	Realistic Utt. %	Realistic Examples (test-set)	Unrealistic Examples (test-set)
French (fr)	0.1	95.4%	Me montré les vols directs de Charlotte à Minneapolis mardi matin . Quelle compagnie aérienne fut YX	Me montré des vols entre Détroit er St. Louis sur Delta Northwest US Air est United Airlines . Lister des vols de Las Vegas à Son Diego
German (de)	0.2	94.5%	Zeige mir der Flüge zwischen Housten und Orlando Welche Flüge gibt es vom Tacoma nach San Jose	Zeige mit alle Flüge vor Charlotte nach Minneapolis zum Dienstag morgen Zeige mit Flüge an Milwaukee nach Washington DC v. 12 Uhr
Spanish (es)	0.1	96.9%	qué aerolíneas vuelan de baltimore a san francesc muéstrame vuelos entr toronto y san diego	necesito información de un vuelo y la tarifa de oakland a salt lake city para el jueves antes e sus 8 am de nuevo york a las vegas el domingo con la tarde
Hindi (hi)	0.05	95.4%	मुझे डेल्टा उड़ानों के बारे में बताइए जो कोच के यात्रियों को नाश्ता देता हों मुझे मेम्फिस से लास वेगास तक उड़ान की जरूरत है	सोमवार दोपहर ने लॉस एंजिल्स से पिट्सबर्ग रविवार दोपहर को मियामी में क्लीवलैंड
Japanese (jp)	0.1	**92.3%**	来F水曜日にカンザスシティ初シカゴ行きでシカゴ の 午後 7 時ごろ到着して、F りのフライトが木曜日のフライト ワシントン を コロンバス間のすべてのフライトの運賃はいくら	シャFロット空港 の 土曜日 err 午后 1 時に出Fする US エアFのフライトをリストアップして 水曜日のフェニックスFミルウォFキF逝き
Chinese (zh)	0.1	**86.2%**	我需要 4 点 后 在 达拉斯起飞飞往旧金山的联程航班 请列出从纽瓦克飞往 洛杉机 的航班	然而 每天上午 10 点之前从密尔沃基飞往亚特兰大 拉瓜迪亚 了 豪华轿车服务要多少钱

Figure 1: MultiATIS++ test set injected with real-world noise mined from Wikipedia edits. The highest error injection ratio found to be realistic by human experts is shown alongside the realistic utterance percentage. We do not include the noisy test sets for Chinese and Japanese in our analysis owing to low ($< 95\%$) realism.

ample, typos in Devanagari script can differ from those seen in Latin scripts (e.g. स्कूल → सकउल in Devanagari showcases that a joined character is incorrectly separated into two characters in the word 'school').

Thus, to improve the robustness of pretrained multilingual models across noise in all languages, we propose Robust Constrastive Pretraining (RCP) that couples multilingual noisy data-augmentation with a contrastive learning loss term during pretraining; this encourages the model to develop similar representations for the original and the noised version of a sentence.

On the noisy test sets, our method improves the multilingual model performance across all metrics and multilingual tasks– IC% by 4.9% on MultiATIS++, 4.1% on MultiSNIPS; SL-F1 by 18.4 on MultiATIS++, 8.6 on MultiSNIPS; NER-F1 by 2.9 on WikiANN; NLI% by 0.7% on XNLI. In summary, our primary contributions are:

1. We construct multilingual test data to evaluate the robustness of NLP models to noise (§3).

2. We show that the performance of existing multilingual language models deteriorates on four tasks when tested on the noisy test data (§5.1).

3. We introduce Robust Contrastive Pretraining (RCP) to boost the robustness of existing multilingual language models (§5.2).

Our code and data is available on *Github (repo: amazon-science/multilingual-robust-contrastive-pretraining)* .

2 Related Work

Many prior works demonstrate the brittleness of neural models on different noise phenomena such as misspellings (Belinkov and Bisk, 2017; Karpukhin et al., 2019; Moradi et al., 2021), casing variation (van Miltenburg et al., 2020), paraphrases (Einolghozati et al., 2019), morphological variance (Tan et al., 2020), synonyms (Sengupta et al., 2021), and dialectical variance (Sarkar et al., 2022). A popular approach to improve the robustness to noise is fine-tuning models with data augmentation (Feng et al., 2021) at either the pretraining (Tan et al., 2020; Sarkar et al., 2022) or the task-training stage (Peng et al., 2021). These works consider monolingual pre-trained models and primarily focus on English. While recent works on token-free models motivate robustness in multilingual settings (Clark et al., 2021; Xue et al., 2022; Tay et al., 2021), *examining the robustness of SOTA*

multilingual pre-trained models (and improving them) remains unexplored. Hence, we investigate– (1) are multilingual models robust to noise seen in different languages (that may be dissimilar to noise types seen in English)? (2) can we get and leverage multi-lingual noise data to improve multilingual models? and (3) do automatic data-augmentation methods designed for English improve robustness to multilingual noise?

To boost the robustness of multilingual models to diverse multilingual noise, we leverage multilingual data augmentation at the pretraining stage and use contrastive learning. Our effort complements work in computer vision that showcases contrastive learning with adversarial learning at task-training (Fan et al., 2021; Ghosh and Lan, 2021) and pre-training time (Jiang et al., 2020; Kim et al., 2020) can improve model robustness. NLP has also seen a plethora of work that leverages contrastive learning, but seldom to alleviate robustness concerns (Jaiswal et al., 2020). Similar concepts, such as Adversarial Logit Pairing (Einolghozati et al., 2019), used at task-training time have proven to be less effective than data augmentation approaches (Sengupta et al., 2021) in boosting robustness.

All the aforementioned works lack in at least one of the two novel aspects of this paper– robustness to real-world (as opposed to adversarial) noise, and/or multilinguality. Lastly, the aspect of cross-lingual knowledge transfer has been studied in the context of different NLP tasks; typically, from a high-resource language to a low-resource one, as exemplified by the XTREME benchmark (Hu et al., 2020). In this paper, we investigate the cross-lingual transferability of robustness to real-world noise.

3 Constructing Noisy Test Data

As no existing benchmarks exist to evaluate the robustness of multilingual models, we construct noisy test sets in multiple languages for four tasks. First, we construct a word-level error-and–correction dictionary by leveraging the Wikipedia edit corpora. Then, we sample replacements from this dictionary and inject them into the test data for the various multilingual tasks, focusing on replacements that only affect individual words but do not change word order. Finally, we conduct human evaluation to filter out test sets that are not deemed to be realistic by language experts.

3.1 Wiki-edit Mining

Wikipedia[2] is a public encyclopedia available in multiple languages. Wikipedia editors create and iteratively edit its contents. We leverage these edits to construct error-correction word dictionaries (later used to create noisy test data). Our approach to mining edits is similar to Tanaka et al. (2020), but we consider multiple languages (as opposed to only Japanese), and additionally create dictionaries of word-level edits.

To isolate likely useful edits, we first consider each revision page of an article and split it into a list of sentences using NLTK (Bird et al., 2009). Second, we filter out sentence pairs from two consecutive edit versions ensuring both sentences have (1) 2-120 tokens, (2) a difference if < 5 tokens, and (3) a relative edit-distance within 30% of the shorter sentence. Third, we leverage language-specific tokenizes `difflib`[3] to extract exact token-level deltas between the sentence pair. At last, we ensure word pairs (in these deltas) that have at least one character-level Levenshtein edit-distance from each other[4] and none of words are only numbers or punctuation tokens. Note that edits to Wikipedia involve changes to factual information, such as dates, rather than incorrect spelling or grammar; thus, the last step is necessary.

We can finally create a *noise dictionary* of correct-to-incorrect words that has frequency information about the different errors. For example, an element of the dictionary (in Spanish) looks like `{de: [(del, 0.52), (se, 0.32), (do, 0.1), (dë, 0.04), (en, 0.02)]}`.

3.2 Injecting Noise into Test sets

We use the noise dictionaries to create a noised version of the original test data for the four tasks– MultiATIS++ (Xu et al., 2020), MultiSNIPS, WikiANN (Pan et al., 2017) and XNLI (Conneau et al., 2018). After tokenization, we sample tokens randomly without replacement. In each sampling step, we sample based on a uniform probability distribution over the individual tokens and then check if the token exists in the noise dictionary. If so, we replace it with a noised version from the dic-

[2] https://meta.wikimedia.org/wiki/List_of_Wikipedias

[3] https://docs.python.org/3/library/difflib.html

[4] For Chinese characters, including Kanji, even a single character distance could imply a different word.

tionary; the noised version is sampled based on its probability in the noise dictionary (that is proportional to the frequency of its occurrence in the noisy corpora). This procedure continues till we noise a particular number of tokens, precisely between 1 and $\min(4, pL)$ where p a controllable fraction (chosen as a hyperparameter at first, and finalized based on human evaluation described in §3.3), and L is the number of words in the sentence.

3.3 Human Verification of Noised Test-sets

During human evaluation, we analyse the noisy data created for the MultiATIS++ dataset. We asked the language expert to assume that a user who may not be a native speaker, or in a hurry, or sloppy, was trying to find out flight information via text chat, and evaluate realism with this in mind. Note that analysis of noise types for MultiATIS++ generalizes well to other datasets as we use the same error-correction dictionaries for injecting noise into all the test-sets.

Our language experts have graduate/doctoral degrees in linguistics, computational linguistics, or natural language processing and are fluent/native speakers of the respective languages. We employed the human experts and compensated them fairly to conduct this study (see §7 for details). The experts are given 45 examples without being told that 15 examples have 5%, 15 have 10%, and 15 have 20% noised tokens and asked three questions about each example. (1) Is the noised sentence realistic, moderately realistic, or unrealistic? (2) What type of noise is present in the sentence (we supply an initial list and let them add more)? and (3) Are the intent and slot labels unchanged? Based on their initial feedback, we choose the most realistic noise fraction (i.e. 5, 10 or 20%) and provide them with 60 more examples from that set. We considered 15 utterances enough to determine the noise fraction, but used the ratings on 75 utterances for evaluating realism (see realistic utterance % in Figure 1).

In Figure 1, we summarize the results of the human evaluation. Column two shows the error injection ratio that was deemed to have more than 95% realistic utterances. We set a high cut-off of 95% to ensure we can make confident statements about the robustness of multilingual models to realistic alterations exhibited in our benchmarks. Hence, Chinese and Japanese (with a realism of 86.2% and 92.3% resp.) are omitted in our benchmarks. The last two columns highlight examples deemed

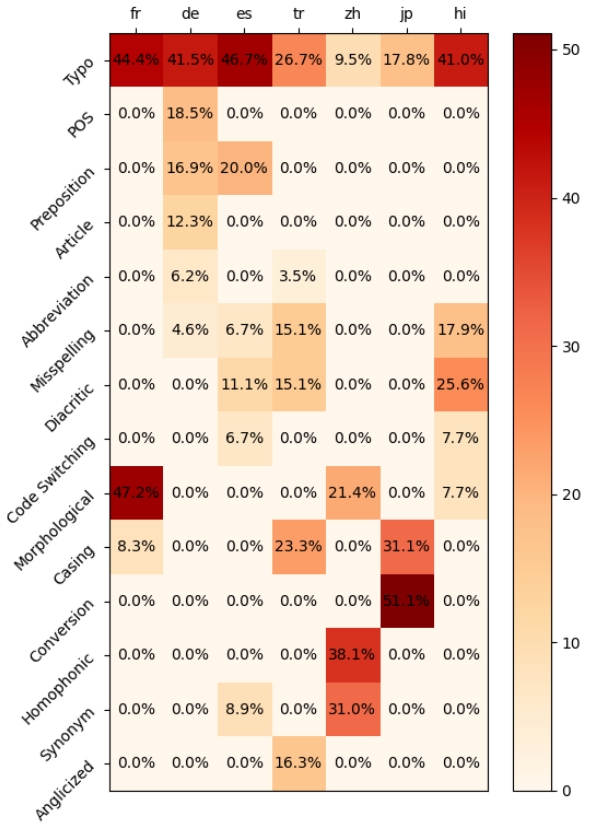

Figure 2: The column-wise color density (which adds up to one) shows the percentage of a different noise types observed for a particular language. The row-wise values show that some noise types (eg. homophononic) is only present for a single language (eg. zh).

as realistic and unrealistic by human experts with the noised tokens highlighted in orange.

Given the sentence length and similarity in task types, we use the error injection percentage determined to be the most realistic for MultiATIS++ as the error injection percentage for MultiSNIPS and Wiki-ann. For XNLI, experts deemed higher noise injection ratios (of > 0.05) to be unrealistic (15% for 0.1, 27% for 0.2) because (1) the premise, usually much longer than sentences in MultiATIS++, had (impractically high) number of noise tokens, and (2) the classification label (implies/neutral/-contradicts) sometimes changed with large noise additions. Thus, for XNLI, we choose 0.05 to be the default noise injection ratio. Finally, one expert noted the Turkish data for MultiATIS++ lacked many diacritic characters, muddling the distinction between noise injected by our procedure and existing misspellings; hence, it was ignored.

In Figure 2, we list the noise-types identified by our experts in different languages. While certain noise-types, such as typographical errors, misspellings are common across multiple languages,

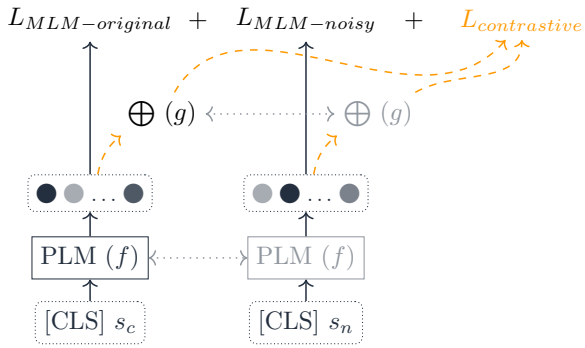

Figure 3: Loss function for fine-tuning a Pretrained Language Model (PLM) using Robust Contrastive Pre-training (RCP).

there are various language-specific noise-types, such as homophonic errors (for zh), Kanji conversion errors (for ja), anglicization (for tr) (we showcase some examples in Appendix A). Given disjoint noise types across languages, we expect that augmentation with errors seen in English (using approaches proposed by prior works) will generalize better to languages that share error types.

4 Robust Contrastive Pre-training (RCP)

Motivation and Approach While *task-time* data augmentation (aka adversarial training) has been effective to boost the robustness of pre-trained models for English, we face two major challenges– (1) lack of supervised multilingual training data in our zero-shot setting, and (2) lack of approaches to synthetically generate noise data for non-English languages. We overcome these with a multilingual data-augmentation approach at *pre-training time* that uses the multilingual Wikipedia edit corpus to expose our models to human errors during pre-training. Here, the need of ex-situ injection of noise (for test-data creation §3) is unnecessary as our edit corpus contains pairs of similar sentences, i.e. a version of the sentence before and after revision by a Wikipedia contributor (§3.1). To encourage the model to align the representations of these two sentences in the encoder's output space, we use a contrastive loss term (see Figure 3). Building on previous work on contrastive learning (Giorgi et al., 2021), Robust Contrastive Pre-training (RCP) considers the original and edited version of a sentence as positive examples and other unrelated sentences as the negative examples.

Similar to Giorgi et al. (2021) and Reimers and Gurevych (2019)), we map variable length sentences to fixed-length embeddings with a pooler $e_i = g(f(s_i))$, where $f(\cdot)$ is a transformer encoder, and $g(\cdot)$ is the mean of the token-level embeddings. Given a batch of N (noisy, clean) sentence tuples, we set our original sentence s_c as the anchor and the noisy version s_n as the corresponding positive pair. [5] Other sentences in the batch (i.e. $\neq s_n$) are deemed to be negative examples. We consider the InfoNCE/NT-Xent loss (Sohn, 2016) for our per-example contrastive loss:

$$\ell(i,j) = -\log \frac{\exp(\text{sim}(e_i, e_j))}{\sum_{i \neq k} \exp(\text{sim}(e_i, e_k)/\tau)} \quad (1)$$

where $\text{sim}(u,v) = u^T v / ||u||_2 ||v||_2$ denotes the cosine similarity of two vectors u and v and $\tau > 0$ denotes the temperature hyper-parameter. Thus, our final contrastive loss function is

$$L_{\text{contrastive}} = \sum_{i=1}^{N} \ell(c,n) + \ell(n,c)$$

We additionally use the standard MLM loss at pre-training time, masking 15% of the input tokens of every sentence (i.e. both noisy and clean) independently. Therefore, our final loss function is

$$L = L_{\text{contrastive}} + L_{\text{MLM-noisy}} + L_{\text{MLM-original}}$$

$L_{\text{MLM-original}}$ is the MLM loss on original sentences, and ensures the model does not 'forget' its original pre-training task. $L_{\text{MLM-noisy}}$ is the MLM loss on noisy sentences, and can be thought of as data-augmentation at pre-training time.

Pre-training Details Following the Domain Adaptive Pre-Training (DAPT) approach (Gururangan et al., 2020), we start with an existing multilingual pre-trained model and fine tune it with our RCP objective. Unlike DAPT, we are not interested in specializing in a particular domain, but in increasing robustness to errors. As mentioned before, we use (unfiltered) pairs of correct/incorrect sentences from the multilingual Wikipedia archive and include sentences from the Lang8 corpus.[6] The Lang8 corpora consists of a smaller number of sentences compared to the Wikipedia corpus, but proves to be apt for our purpose; it consists of pairs of sentences– one written by a non-native speaker who is learning the language (eg. "As the winter

[5] One obvious choice would be for clean sentence with index $2i$, the noisy sentence has index $2i-1$.

[6] https://sites.google.com/site/naistlang8corpora/

Dataset	Task	Size (training)	Languages	Epochs	Learning Rate	Seeds
MultiATIS++ (Xu et al., 2020)	IC/SL	5k	de,en,es,fr,hi	80	1E-04	5
+ training data aug.		18k	de,en,es,fr,hi	20	1E-04	5
MultiSNIPS	IC/SL	13k	en,es,fr,hi	40	1E-04	5
+ training data aug.		72k	en,es,fr,hi	10	1E-04	5
WikiANN (Pan et al., 2017)	NER	20k	de,en,es,fr,hi,tr	3	2E-05	5
XNLI (Conneau et al., 2018)	NLI	392k	de,es,fr,hi,tr	5	2E-05	5

Table 1: Data-set characteristics and hyper-parameters for our experiments.

Model	Original/ Noisy	MultiATIS++		MultiSNIPS		Wiki-ann	XNLI
		IC%	SL-F1	IC%	SL-F1	NER-F1	NLI%
XLM-R_{base}	Original	**90.68**	**71.45**	**92.93**	**68.01**	**74.14**	**76.69**
	Noisy	**89.65**	**62.3**	**90.46**	**61.63**	**69.48**	**74.38**
mBERT	Original	86.29	64.95	78.65	59.05	73.92	70.82
	Noisy	85.42	55.17	75.35	53.71	69.38	68.44

Table 2: Performance of pre-trained multilingual models on the four multilingual datasets averaged across languages and 5 seeds. XLM-R_{base} outperforms mBERT on Original and Noisy test data across all metrics.

is coming, I'm getting to feel better.") and a re-write of this sentence by a native speaker (eg. "as the winter is coming, I'm starting to feel better."). More details about the individual corpora can be found in Appendix C.

We note that the pre-training corpus is not exactly the same set of sentences used to construct our noise dictionaries in §3.1. In this case, the only criteria for inclusion is a length difference of < 5 tokens, and a relative edit-distance of 30% of the shorter sentence (see appendix C for more details). Hence, we incorporate training data from the corpora that exhibit changes beyond simple typos (such as paraphrasing, sentence-level morphological variance) in the pre-training stage.[7]

Similar to Gururangan et al. (2020), we fine tune for 25k steps with a batch size of 2048 sentences to create two pretrained models– one with $\mathcal{L}_{\text{contrastive}} + \mathcal{L}_{\text{MLM-noisy}} + \mathcal{L}_{\text{MLM-clean}}$ (referred to as Robust Contrastive Pre-training or RCP) and an ablation without the contrastive term, i.e. $\mathcal{L}_{\text{MLM-noisy}} + \mathcal{L}_{\text{MLM-clean}}$. The latter setting represents a pure (pre-training time) data augmentation approach such as Tan et al. (2020) (termed *p(aug)* in Table 3). See Appendix D for more hyper-parameters and settings.

5 Experiments and Results

We divide this section into three parts. In §5.1, we analyze the robustness of popular multilingual language models in the zero-shot cross-lingual setting. In §5.2, we show that Robust Contrastive Pre-training (RCP) improves the robustness of existing baselines on noisy test-data for all tasks– joint intent classification and slot labeling (IC-SL), Slot-Labeling (SL) Named Entity Recognition (NER) and Natural Language Inference (NLI)– and (not only maintains but) improves performance on the original test data. Finally, in §5.3, we conduct failure mode analysis for MultiATIS++ and discover that the model trained with RCP makes more explicable sequence-labeling errors (for slot-value prediction) in comparison to existing baselines.

Setup We consider four datasets (shown in Table 1) and four metrics for evaluation. Two of these metrics consider sentence classification accuracy– Intent classification Accuracy (IC%) for the goal-oriented dialog text datasets MultiATIS++ and MultiSNIPS, and classification accuracy (NLI%) for XNLI. We also consider F-score for sequence-labeling tasks– Slot Labelling (SL-F1) for MultiATIS++ and Multi-SNIPS++ and Named Entity Recognition (NER-F1) for Wiki-ann. Table 1 shows the languages present in the noisy test data and the size of the English training data used in our zero-shot cross-lingual setting. Note that for task-time data augmentation, we follow the strategy of *aggregate noise augmentation* proposed in (Sengupta et al., 2021) for English, which involves augmenting training data with a variety of synthetic noise types such as typos, making words ALL-CAPS, abbreviations etc. As this augmentation procedure increases the size of the training data-set ≈ 3.5 times for MultiATIS++ and ≈ 5.5 times for

[7] Unfortunately, the benefit of including sentence-level noise in the pre-training phase is not directly examined by our benchmarks, which focus more on word-level noise.

Task	Metric	XLMR	XLMR +p(aug)	XLMR +t(En-aug)	XLMR +RCP (Ours)	XLMR +RCP+t (Ours)	Gain
MultiATIS++	IC%	89.65	93.10	91.26	93.80	**94.57**	+4.92
	SL-F1	62.30	67.47	74.62	67.45	**80.68**	+18.38
MultiSNIPS	IC%	90.46	93.98	91.60	93.79	**94.53**	+4.07
	SL-F1	61.63	66.67	66.44	67.69	**70.20**	+8.57
Wiki-ann	NER-F1	69.48	72.32	-	**72.37**	-	+2.89
XNLI	NLI%	74.38	74.83	-	**75.06**	-	+0.68

Table 3: Average performance across languages and five seeds. We abbreviate the baselines, multi-lingual pre-training time augmentation as p(aug), and English task-time (aggregate) data augmentation as t(En-aug). 'RCP' stands for 'Robust Contrastive Pre-training', and 'RCP + t' means combining RCP with task-time data augmentation. 'Gain' refers to the increase in performance of the best method vs. XLM-R_{base}.

MultiSNIPS, we find that training for fewer epochs yields the best results.

5.1 Robustness of Multilingual Models

We compare the robustness of two popular pre-trained language models– XLM-R_{base} and multi-lingual BERT in the zero-shot cross-lingual setting.[8] In this setup, we fine-tune the pretrained language models on the task-training data in English and test (zero-shot) on multilingual test sets. The results reported in Table 2 are averaged across multiple languages for brevity (and provide a detailed breakdown in Appendix E). A secondary goal of this experiment was to decide which pre-trained model to use for further experiments and we base our judgements on twelve metrics across four datasets.

Noise always leads to a decrease in performance. On average, the accuracy of both models decreases by $\approx 2\%$ for sentence-level tasks (IC%, NLI%), and by ≈ 6.6 F1-points on sequence-labeling tasks (SL, NER), on noisy data compared to clean data. This can perhaps be explained by the ability to ignore a particular token for sentence-level tasks, whereas every token, including noisy ones, need to be assigned a label for sequence-labeling tasks.

We observe that XLM-R_{base} outperforms mBERT on all the twelve metrics. For sentence-level tasks (i.e. IC%, NLI%), XLM-R_{base} outperforms mBERT by 8.43% on average on the noisy test-sets and for sequence-tagging tasks (i.e. SL, NER), XLM-R_{base} outperforms mBERT by 5.1 F1-points. In general, XLM-R_{base} also seems to be a model better suited for these tasks in the zero-shot cross-lingual setting, as we also see similar gains when using XLM-R_{base} on the clean data.

[8] We also considered Canine-c (Clark et al., 2021), a token-free baseline, but observed poor performance compared to XLM-R_{base} and BERT on IC-SL tasks (see Table 10).

Task	Metric	XLMR	Ours	Gain
MultiATIS++	IC%	90.68	95.32	+4.64
	SL-F1	71.45	84.07	+12.62
MultiSNIPS	IC%	92.93	95.66	+2.73
	SL-F1	68.01	74.39	+6.38
Wiki-ann	NER-F1	74.14	76.34	+2.2
XNLI	NLI%	76.69	76.75	+0.06

Table 4: Comparison of our RCP method with the baseline XLM-R_{base} model on the original (clean) test data.

Breaking the results down by language (see Appendix E for detailed results), XLM-R_{base} outperforms mBERT on average across all languages. Specifically XLM-R_{base} outperforms mBERT on German (in 6/8 metrics), on Spanish (10/10), on French (8/12), on Hindi (12/12), and on Turkish (4/4). As German is missing in MultiATIS++ and Turkish is only present in WikiANN and XNLI among the four datasets, the overall number of metrics is less than 12 for these two languages. Given these results, we consider XLM-R_{base} as the baseline multilingual language model in the rest of our experiments.

5.2 Robust Contrastive Pre-training Results

To showcase the efficacy of our RCP approach, we compare our approach to a popular multilingual model XLM-R_{base}, which performed best in the previous section, and two augmentation solutions that were proposed earlier and shown to improve robustness of English language models to real-world noise. First, we consider a pre-training time data augmentation approach, similar to Tan et al. (2020), by continuing to pre-train XLM-R_{base} on *noisy* multilingual data; see section 4. Next, we consider augmenting task-time data with a combination of various noise types, following Sengupta et al. (2021) that shows using this aggregate data augmentation during task-time finetuning improved

Error Type	Utterance	Slot-lables
Hallucination	*Ichs* brauche einen Flug von Memphis nach Tacoma, der ***uber*** Los Angeles fliegt	✔ O (über) ❗ airline_code
Contextual	Zeige *mit der* Erste-Klasse und Coach-Flüge vom **JFK** nach Miami	✔ fromloc.airport_code ❗ toloc.airport_code

Table 5: Examples of slot labeling errors in German– errors are in *italics*; misclassified tokens are **bold**.

performance on both noisy and clean data for IC-SL tasks like ATIS and SNIPS. For the latter, we treat it as a baseline for zero-shot cross-lingual transfer for the dialog-datasets–MultiATIS++ and MultiSNIPS– and also combine it with our pre-training time approaches.

As shown in Table 3, our approach can improve the performance of current multilingual models across all 4 tasks and datasets. For the multilingual goal-oriented dialog datasets, our approach coupled with task-time augmentation outperforms all the other methods. We observe that the gain for SL tasks is higher than that obtained for IC tasks. Although we analyze the SL results further in §5.3, we highlight that IC accuracy is less affected by noise than SL F1; this provides more headroom for improving SL metrics. The highest gains are observed for Hindi where the XLM-R_{base} model has the worst SL performance on noisy data (42.86 for MultiATIS++, 36.93 for MultiSNIPS). Likewise, we also observe improvement on XNLI% and NER-F1; the largest improvement is again seen on the noisy data for Hindi. Overall, the gain on sequence-labelling tasks is larger than the gain on sentence-level classification tasks.

Does this improvement on noisy data come at the cost of worse performance on clean data? In Table 4, we show that the best performing models shown in Table 3 (XLMR+RCP+t for MultiATIS++ and MultiSNIPS, and XLMR+RCT for WikiANN and XNLI) also improve the performance on clean test data. Further, the magnitude of growth seen on clean data is like the ones seen on the noisy test data. For slot-labeling errors, we observe a particular kind error which occurs on both clean and noisy data that our model mitigates; we provide more details on this in the next section. For IC and XNLI, we found no specific error pattern that distinguishes between XLM-R_{base} and our model. Thus, we believe that our approach mostly improves the overall quality of the model's representation rather than just its downstream robustness. In the future, one can consider if an upper bound on model quality exists beyond which the tension between accuracy on clean data and robustness to real-world noise emerges (Tsipras et al., 2018).

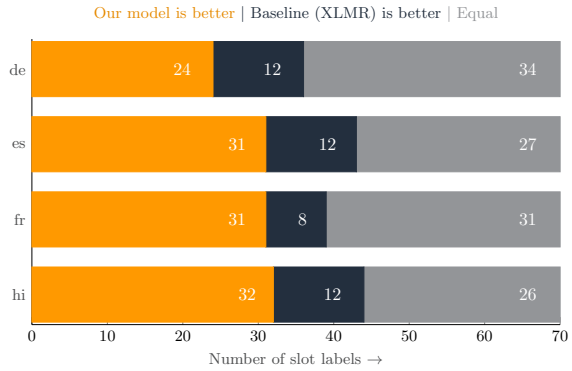

Figure 4: Comparing the number of slot labels for which our model *vs.* the baseline performs better.

Finally, we note that beyond improving performance on clean and noisy data, our approach reduces the *disparity* in performance between the clean and noisy test sets. For MultiATIS++, the disparity reduces by 0.3% for IC% and 5.76 for SL-F1; for MultiSNIPS, it reduces by 1.34% for IC% and 2.19 for SL-F1; for WikiANN, it reduces by 0.68 for NER-F1; and for XNLI, it reduces by 0.9% for NLI%.

5.3 Result Analysis

Given the large improvement seen on sequence labeling tasks, we zoom in on the SL metrics for MultiATIS++. In Figure 4, we show the number of slot labels on which our method outperforms (has fewer misclassifications than) the baseline, vice versa, and where they perform equally. Our method clearly out-performs the baseline on at least twice the number of slot-labels– $2\times$ better on German, $\approx 2.6\times$ times on Spanish and on Hindi, and $\approx 4\times$ on French. Across all languages, our model always outperforms XLM-R_{base} on eight slot-labels. These slots correspond to relative times ('leaves in the *evening*'), relative dates ('traveling the *day after tomorrow*'), relative costs ('*cheap* flights'), meal names ('flights that offer *breakfast*'), and carrier tokens/non-slot values ('*that offer* breakfast'). We postulate these slot values are more common in the

N/O	Model	de	es	fr	hi
Noisy	XLMR	315	358	413	671
	XLMR+RCP+t	21	123	33	204
Original	XLMR	208	262	334	460
	XLMR+RCP+t	19	106	22	180

Table 6: Reduction in hallucination error (i.e. model identifies irrelevant tokens as a slot value) counts.

Languages	de	es	fr	hi
(r1) Top-confusion changes to no-label (w/ RCP)	7	8	6	17
(r2) Confusions becomes more explicable (w/ RCP)	8	3	3	4

Table 7: Number of slot-labels that our model misclassified to (r1) a no-slot or (r2) a more explicable slot-label.

pre-training data compared to proper nouns such as airline, airport or city names and thus, understood in noisy contexts. In turn, variations of these words are mapped closer in the embedding space and the classifier is more robust to such errors.

Upon further analysis, we observe two distinct patterns– (1) reduction in *hallucination errors*, i.e. errors where an irrelevant carrier phrase token is labeled to be a slot value, and (2) errors become more contextual– misclassification is to related classes (see examples in Table 5).

In Table 6, we highlight the distribution of hallucination errors and observe that the number of carrier phrase tokens that the baseline XLM-R_{base} misclassifies as a slot-value reduces (by $> 10\times$ for German and French, and $\approx$2-3$\times$ for Hindi and Spanish) with our approach on both the original and the noisy test data. This observation aligns with our initial reasoning that the contrastive loss term at pre-training time helps the model develop a better understanding of non-slot words as the model learns to identify such words (and their noisy forms) in both linguistically correct and noisy contexts. Note that the latter signal is missing for the XLM-R_{base} baseline.

For a subset of the slot labels, the class to which it was misclassified (with the highest frequency) differed between the XLM-R_{base} baseline and our model. In Table 7, we highlight two scenarios where the most-confused label changed from (r1) an incorrect slot label (eg. *meal_code* $\rightarrow$ *airline_code*) to no-label (i.e. *meal_code* $\rightarrow$ O), and (r2) from an inexplicable slot label (*state_code* $\rightarrow$ *transport_type*) to a more explicable one (*state_code* $\rightarrow$ *state_name*) when the RCP method is used (we use the explicable/inexplicable terminology of Olmo et al. (2020)). Thus, our approach inadvertently improves the explicability of the failures made during slot-labeling.

6 Conclusion

In this paper, we investigate the robustness of pretrained multilingual models in the zero-shot cross-lingual setting on four tasks– intent classification, slot labeling, named entity recognition, and natural language inference. Given the dearth of existing datasets to benchmark the robustness of existing multilingual models, we develop noisy test data by injecting errors mined from an edit corpus (and conduct expert evaluation for quality assurance). Our identification of noise types across various languages motivates the necessity of language specific investigation in the future. Finally, demonstrate existing baselines perform poorly in the presence of noise in the test data and propose Robust Contrastive Pretraining to boost the robustness of these multilingual models.

7 Ethical Considerations

For the human annotation tasks of (1) identifying language-specific noise types, and (2) ranking their realism, we leveraged the effort of full-time employees at Amazon. The annotators had advanced degrees in linguistics or natural language processing, and were fluent/native in the languages they annotated. Amazon compensated them under a competitive industry rate, which is above the minimum hourly pay rate, for their particular job role (which included Applied/Research Scientists, Software/Language Engineers, Linguists, and Language Consultants).

Acknowledgements A special thanks to Saab, Batool Haider and M. Saiful Bari for sharing with us the MultiSNIPS dataset. In addition, we want to express our gratitude to members of the AWS AI Lab for their valuable comments, suggestions, and participation in our pilot and human labeling studies (in no particular order)– Sebastien Jean, Volha Belash, Arshit Gupta, Berk Sarioz, Maansi Shandilya, Raphael Shu, Abhilash Panigrahi, Lorenzo Lambertino, and Yi Zhang. Finally, we are grateful to the anonymous reviewers who have helped us improve this paper.

8 Limitations

8.1 The Umbrella of Realistic Noise

'Realistic noise' is too abstract a category. We mostly concern ourselves with real-world errors and their corrections appearing in existing corpora (with criteria like a small character-level edit distance). But this could include things like better paraphrasing, use of more appropriate synonyms or morphology that can be viewed as language variation rather than noise; this could be one reason we notice improvements on the original (i.e. unnoised) test data. Yet, to distinguish ourselves from the terminology of synthetic or adversarial noise, we choose this (imperfect) terminology of real-world/realistic noise as in Sengupta et al. (2021) to bracket all our noise types under a single class.

8.2 Language Choice and Diversity

This work considers (relatively) high-resource languages. This makes it easier for us to find publicly available corpora from where we can mine error/-correction data and use it to improve the model's understanding of errors and, in turn, boost their robustness to real-world noise. But this is only the first step towards developing an understanding of noise phenomena in languages beyond English, bench-marking multi-lingual model performance in such settings, and improving their robustness. Further, we do notice that Hindi (and, to some extent, Turkish) are relatively low resource languages when it comes to pre-training data (see Table 8 in Appendix). We hope future work builds on this and explores a greater variety of languages.

8.3 Zooming-in on Individual Tasks

Many of our human studies are based on a subset of datasets (eg. MultiATIS, XNLI). It is possible individual tasks and further, individual datasets need more fine-grained human attention. Given language expertise for several datasets and several languages is difficult/costly, we made the choice to concentrate on a smaller number of datasets in order to provide a more rigorous analysis. We hope future work can expand the number of tasks and datasets covered so we have a more comprehensive analysis of how multilingual noise affects pre-trained models.

References

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Bo Zheng, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2022. XLM-E: Cross-lingual language model pre-training via ELECTRA. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6170–6182, Dublin, Ireland. Association for Computational Linguistics.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2021. CANINE: pre-training an efficient tokenization-free encoder for language representation. *CoRR*, abs/2103.06874.

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

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Arash Einolghozati, Sonal Gupta, Mrinal Mohit, and Rushin Shah. 2019. Improving robustness of task oriented dialog systems. *arXiv preprint arXiv:1911.05153*.

Lijie Fan, Sijia Liu, Pin-Yu Chen, Gaoyuan Zhang, and Chuang Gan. 2021. When does contrastive learning preserve adversarial robustness from pretraining to finetuning? *Advances in Neural Information Processing Systems*, 34.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Aritra Ghosh and Andrew Lan. 2021. Contrastive learning improves model robustness under label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2703–2708.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. 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 879–895, Online. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2.

Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. 2020. Robust pre-training by adversarial contrastive learning. *Advances in Neural Information Processing Systems*, 33:16199–16210.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. *arXiv preprint arXiv:1902.01509*.

Minseon Kim, Jihoon Tack, and Sung Ju Hwang. 2020. Adversarial self-supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:2983–2994.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Zihan Liu, Jamin Shin, Yan Xu, Genta Indra Winata, Peng Xu, Andrea Madotto, and Pascale Fung. 2019. Zero-shot cross-lingual dialogue systems with transferable latent variables. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1297–1303, Hong Kong, China. Association for Computational Linguistics.

Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2020. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8433–8440.

Pooya Moradi, Nishant Kambhatla, and Anoop Sarkar. 2021. Measuring and improving faithfulness of attention in neural machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2791–2802, Online. Association for Computational Linguistics.

Alberto Olmo, Sailik Sengupta, and Subbarao Kambhampati. 2020. Not all failure modes are created equal: Training deep neural networks for explicable (mis) classification. *ICML Workshop on Uncertainty and Robustness in Deep Learning*.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Zhu Zhang, Chenguang Zhu, Jinchao Li, and Jianfeng Gao. 2021. RADDLE: An evaluation benchmark and analysis platform for robust task-oriented dialog systems. 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 4418–4429, Online. Association for Computational Linguistics.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Soumajyoti Sarkar, Kaixiang Lin, Sailik Sengupta, Leonard Lausen, Sheng Zha, and Saab Mansour. 2022. Parameter and data efficient continual pre-training for robustness to dialectal variance in arabic. In *NeurIPS 2022 Workshop on Efficient Natural Language and Speech Processing (ENLSP)*.

Sailik Sengupta, Jason Krone, and Saab Mansour. 2021. On the robustness of intent classification and slot labeling in goal-oriented dialog systems to real-world noise. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 68–79, Online. Association for Computational Linguistics.

Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29.

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

Yu Tanaka, Yugo Murawaki, Daisuke Kawahara, and Sadao Kurohashi. 2020. Building a Japanese typo dataset from Wikipedia's revision history. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 230–236, Online. Association for Computational Linguistics.

Yi Tay, Vinh Q Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2021. Charformer: Fast character transformers via gradient-based subword tokenization. *arXiv preprint arXiv:2106.12672*.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2018. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*.

Emiel van Miltenburg, Chris van der Lee, Thiago Castro-Ferreira, and Emiel Krahmer. 2020. Evaluation rules! on the use of grammars and rule-based systems for NLG evaluation. In *Proceedings of the 1st Workshop on Evaluating NLG Evaluation*, pages 17–27, Online (Dublin, Ireland). Association for Computational Linguistics.

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

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5052–5063, Online. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

A Examples of Noise/Errors in the test set

In this section, we highlight an example of some of the unique noise types observed for certain languages shown in Figure 5.

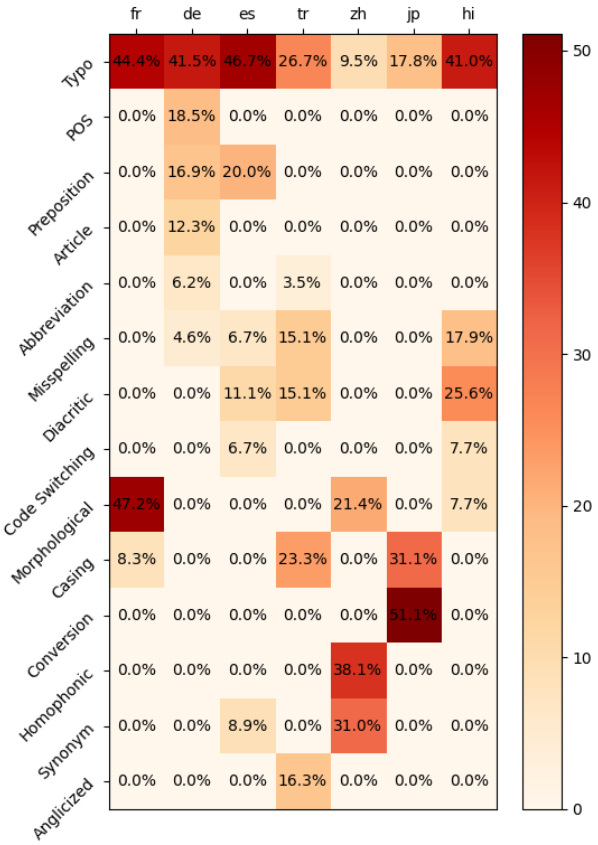

Figure 5: Noise types seen across various languages.

A.1 Typographic Errors (Typos)

Two examples follow for Hindi and Chinese, where experts evaluated based on the Indic Script and the Pinyin keyboards (which is what they use regularly) respectively.

Language	Examples
Hindi (hi)	सोमवार को बरबैंक [सेके] मिल्वौकी तक उड़ानें
Chinese (zh)	列出美国航空周三从密尔沃基飞往圣何塞的航班 **[列出 | 例出]** 美国航空周三从密尔沃基飞往圣何塞的航班

A.2 Preposition Errors

We noticed language experts tagged preposition errors for French and German. Examples follow:

Language	Examples
French (fr)	Je veux un vol aller-retour [de | à] Memphis à Seattle .
German (de)	Wie sieht es [am | im] Mittwoch morgen mit Flügen von DC nach Oakland aus

A.3 Diacritic Errors

Some languages use diacritic characters; although even these diacritics may greatly differ depending on script. Examples from Hindi and Spanish follow.

Language	Examples
Spanish (es)	¿puedo tomar el vuelo [más | mas] corto de milwaukee a orlando ?
Hindi (hi)	d9s किस [तरह | तरह्] का विमान हैं

A.4 Conversion Errors

Kanji conversion error. This error was unique to the Japanese language. Examples follow.

Language	Examples
Japanese (ja)	アトランタからセントルイスまでの火曜日午後 2 時 30 分 [以前 | 依然] のフライト

A.5 Homophonic Errors

This error was unique to Chinese. Words with the same pronunciation (potentially with different tones), but different spelling. Examples follow.

Language	Examples
Chinese (zh)	请列出从 [洛杉矶 | 洛杉机] 飞往夏洛特的航班

A.6 Synonym

Experts marked these as use of a different synonym in Spanish and Chinese only. Note that such variations may not be erroneous but is still considered a noise given they are not used in the original training/testing data in the given context as much. Examples follow.

Language	Examples
Spanish (es)	el próximo miércoles , me gustaría salir de kansas city en [un | el] viaje a chicago que llegue a chicago alrededor de las 7 p m.
Chinese (zh)	请列出从 ewr [到 | 直到] 纽约市的地面交通

A.7 Anglicized

We observed this errors only for Turkish and noticed that experts marked scenarios where an alphabet in the native script was replaced with a particular one in the latin script. Examples follow (note that Turkish examples are drawn from the XNLI dataset, while the others were drawn from MultiATIS++).

Language	Examples
Turkish (tr)	Sonrasında, ilk ziyareti yapmış olan aynı temsilci, soruları cevaplamak ve şikayet örneğinde not edilen sorunları tartışmak [için | icin] yeni sağlayıcıyı yeniden ziyaret eder.
	Konfederasyonun hukuk felsefesi, hem maddi hem de üslupla [karşı | karsi] karşıya geldi.

B Chinese and Japanese Edit Mining

Our two character edit distance criteria for obtaining word-level correct-noisy pairs of words does not work well for Chinese characters, including Kanji for Japanese. This is because words are made up of only a small number of characters (relative to e.g. latin scripts). So we can completely change the semantics with only a small character-level edit distance. We therefore used different noise types: Homophonic and Synonym errors for Chinese and Kanji Conversion errors for Japanese, with brief descriptions and examples in Appendix A. In order to collect homophonic errors we converted words to pinyin[9] (without tone markers) and checked if they were the same in pinyin but different in Chinese characters. To collect synonym noise we labelled words with part-of-speech (POS) tags [10], and kept words that weren't labeled as nouns, verbs, adverbs, keeping e.g. prepositions and conjunctions, with the hope that these would be less likely to involve the kind of big semantic changes you might get with changes to e.g. proper nouns like place names.

However this process was largely driven by trial and error and more work is needed to create a principled pipeline that creates a realistic noise dictionary for these languages.

Finally for Kanji we re-use the criteria of Tanaka et al. (2020) as we re-use their dataset of sentence pairs: checking if the two sentences (containing Kanji) have the same reading.

C Data Details

Table 8 shows the number of Wikipedia and Lang8 sentences (in Millions) we used for fine-tuning the multilingual models in the pre-training stage (§4). As stated earlier, the proportion of data obtained from the Lang8 corpus is less than Wikipedia for most languages except English (where it is comparable) and Japanese (where Lang8 has $\approx 4x$ the data compared to the Wikipedia corpus). In general, Hindi (and Turkish) stand out as a relatively low-resource language in our investigation with less than 0.5 Million sentences.

Language	Lang8	Wikipedia	Total
en	2.5	3.8	6.3
de	0.2	13	13.2
es	0.2	7.6	7.8
fr	0.2	10.7	10.9
hi	0.001	0.1	0.101
ja	4.2	1	5.2
tr	0.02	0.4	0.42
zh	0.6	1.9	2.5

Table 8: Number of sentences (in millions) used for pre-training.

Language	# Pairs (in Millions)
en	0.13
de	0.33
es	0.21
fr	0.27
hi	0.04
ja	0.05
tr	0.25
zh	0.01

Table 9: Number of Error pairs by language.

Table 9 lists the number of correct/incorrect pairs (in Millions) used for noise dictionaries to create the test-sets for the various languages (§3). Here too, we can observe that the number of corrections are relatively less for Hindi. Interestingly, the number of errors for Chinese are the least although it representation is significantly more compared to Hindi. This low number of errors is inline with our human studies where even the 5% error injection was deemed to be unrealistic; futher, such low pairs of errors also reduced the diversity of our test set, which would eventually results in a lower-quality test-set. Hence, we drop it from our evaluation benchmarks.

D Pre-training Settings

For our experiments with Robust Contrasting Pretraining (§4) and variants we use the following hyperparameters and setup. We train on 4 Nvidea V100 GPUs, with a per-gpu batch size of 8 sentences with a maximum sequence length of 128 tokens, and 64 gradient accumulation steps, for an overall batch size of $64 \times 8 \times 4 = 2048$ sentences. We use a masked language modeling mask probability of 15% and a

[9] Using the `pinyin` Python package https://pypi.org/project/pinyin/

[10] With the `jieba` Python package https://pypi.org/project/jieba/.

learning rate of 1e-4 with the Adam optimizer (Kingma and Ba, 2015), and used 16-bit floating point operations. See below for the arguments of the Huggingface transformers (Wolf et al., 2020) masked language modelling script which we modified[11]

```
python -m torch.distributed.launch --nproc_per_node 4 run_mlm.py \
    --model_name_or_path xlm-roberta-base \
    --gradient_accumulation_steps 64 \
    --validation_split_percentage 1 \
    --per_gpu_train_batch_size 8 \
    --dataloader_num_workers 32 \
    --model_type xlm-roberta \
    --mlm-probability 0.15 \
    --learning_rate 1e-4 \
    --num_train_epochs 5 \
    --max_seq_length 128 \
    --line_by_line \
    --do_train \
    --do_eval \
    --seed 42 \
    --fp16
```

E Per-language Results

Table 10 shows the performance of multilingual models like m-BERT and XLM-R_{base} on individual languages. We note that the reduction in performance for high-resource language (e.g. German, French, English) is higher than low-resource languages for several settings. To explain this seemingly surprising result, first notice that the metrics on low-resource languages are already bad, even on clean data. Second, the variety of noise seen for low resource languages is less (see Table 9) compared to high-resource settings. Hence, the effect of less diverse noise in low-resource languages doesn't have as large an adverse effect on already poorly performing models.[12]

Another hypothesis, pending future investigation, is that multi-lingual models trained on more high-resource language data overfit to clean test-sets for these languages and fail to generalize better when faced with noise. For low resource languages, the performance on clean data is already poor because of a lack of sufficient language understanding that prevents over-fitting.

[11] https://github.com/huggingface/transformers/blob/main/examples/pytorch/language-modeling/run_mlm.py

[12] We are told a saying goes (coincidentally) in Hindi, *mare hue ko kya maroge, saheb?*. It implies you cannot do much (by adding noise) to kill the (model that is already) dead.

Dataset	Model	Metric	C/N	de	en	es	fr	hi	tr	Avg.
MultiATIS++	*XLMR*	IC%	C	92.4	98.7	92.0	90.6	79.6	-	90.7
			N	90.9	97.6	91.8	89.5	78.4	-	89.6
		SL-F1	C	74.4	96.0	73.6	70.4	42.9	-	71.5
			N	67.3	82.2	68.2	65.6	38.2	-	62.3
	mBERT									
		IC%	C	83.3	98.3	84.7	88.8	76.3	-	86.3
			N	81.2	97.6	84.3	87.9	76.1	-	85.4
		SL-F1	C	59.9	96.0	65.1	69.8	33.9	-	65.0
			N	51.6	78.5	60.2	64.3	31.3	-	55.2
	(XLMR vs mBERT)			4,0	1,3	4,0	4,0	4,0		
	Canine-c	IC%	C	66.32	96.51	78.41	76.06	71.55	-	77.77
			N	65.13	95.90	78.08	75.06	71.15	-	77.06
		SL-F1	C	31.56	92.19	19.52	23.67	22.81	-	37.95
			N	32.42	78.51	20.25	24.41	22.45	-	35.61
MultiSNIPS++	*XLMR*	IC%	C	-	98.8	94.0	91.3	87.6	-	92.9
			N	-	98.4	92.4	87.0	84.1	-	90.5
		SL-F1	C	-	96.9	72.0	66.2	36.9	-	68.0
			N	-	92.7	63.3	57.7	32.8	-	61.6
	mBERT									
		IC%	C	-	98.9	88.0	88.5	39.3	-	78.6
			N	-	98.2	84.1	82.9	36.2	-	75.4
		SL-F1	C	-	96.5	65.4	59.9	14.5	-	59.1
			N	-	91.3	58.1	52.4	13.0	-	53.7
	(XLMR vs mBERT)				3,1	4,0	2,2	4,0		
	Canine-c	IC%	C	-	69.39	32.88	36.39	23.28	-	40.48
			N	-	69.30	32.57	34.99	23.68	-	40.13
		SL-F1	C	-	0.89.31	24.09	23.06	6.93	-	35.85
			N	-	87.86	22.3	21.49	7.02	-	34.67
WikiANN	*XLMR*	NER-F1	C	74.9	-	75.2	77.2	67.5	75.9	74.1
			N	71.6	-	70.0	71.1	65.1	69.5	69.1
	mBERT									
		NER-F1	C	78.6	-	72.1	79.5	66.2	73.1	73.9
			N	75.4	-	67.1	74.2	63.0	67.3	69.4
	(XLMR vs mBERT)			0,2		2,0	0,2	2,0	2,0	
XNLI	*XLMR*	NLI%	C	76.4	84.6	78.8	77.9	69.7	72.9	76.7
			N	72.6	80.7	76.4	75.7	70.3	70.6	74.4
	mBERT									
		NLI%	C	71.1	82.0	74.9	74.2	60.5	62.2	70.8
			N	67.5	77.9	73.1	71.8	61.5	59.1	68.4
	(XLMR vs mBERT)			2,0	2,0	2,0	2,0	2,0	2,0	

Table 10: Per-language results of cross-lingual transfer from English data (average of 5 random seeds) across 4 datasets analyzed in §5.1 to compare between existing pre-trained multilingual models.