XGLUE: A New Benchmark Dataset for Cross-lingual Pre-training, Understanding and Generation

Yaobo Liang, ✉Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou,
Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti,
Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, Ming Zhou
{yalia,nanduan,yegong,t-niwu,v-fengu,v-weqi,migon,lisho,djiang,gucao,xiafan,bzhang,rahul.agrawal,edwac,
sinwei,tbharti,yiqia,jiuche,winniew,shuguanl,fanyang,Campos.Daniel, ranganm,mingzhou}@microsoft.com

Abstract

In this paper, we introduce **XGLUE**, a new benchmark dataset that can be used to train large-scale cross-lingual pre-trained models using multilingual and bilingual corpora and evaluate their performance across a diverse set of cross-lingual tasks. Comparing to GLUE (Wang et al., 2019), which is labeled in English for natural language understanding tasks only, XGLUE has two main advantages: (1) it provides 11 diversified tasks that cover both natural language understanding and generation scenarios; (2) for each task, it provides labeled data in multiple languages. We extend a recent cross-lingual pre-trained model Unicoder (Huang et al., 2019) to cover both understanding and generation tasks, which is evaluated on XGLUE as a strong baseline. We also evaluate the base versions (12-layer) of Multilingual BERT, XLM and XLM-R for comparison. [1]

1 Introduction

Pre-training + Fine-tuning has become a new NLP paradigm, where the general knowledge are firstly learnt from large-scale corpus by self-supervised learning and then transferred to downstream tasks by task-specific fine-tuning. Three different types of pre-trained models are explored recently, including *monolingual pre-trained models* (Radford et al., 2018; Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019b; Dong et al., 2019; Lewis et al., 2019a), *multilingual and cross-lingual pre-trained models* (Devlin et al., 2019; Conneau and Lample, 2019; Huang et al., 2019; Conneau et al., 2019) and *multimodal pre-trained models* (Lu et al., 2019; Li et al., 2020; Chen et al., 2019; Zhou et al., 2020). In this paper, we focus on the cross-lingual pre-trained models, due to their importance to alleviating the low-resource issue among languages, where an NLP task often has rich training data in one language (such as English) but has few or no training data in other languages (such as French and German). In order to further advance the development of cross-lingual pre-trained models for various downstream tasks in different languages, this paper introduces **XGLUE**, a new benchmark dataset that can be used to: (i) train large-scale cross-lingual pre-trained models using multilingual and bilingual corpora, (ii) evaluate generalization capabilities of the cross-lingual pre-trained models across a diverse set of cross-lingual tasks.

The contribution of XGLUE is two-fold. First, it provides 11 diversified cross-lingual tasks covering both understanding and generation scenarios. XTREME (Hu et al., 2020) is a concurrent work of XGLUE. But it includes cross-lingual understanding tasks only. Besides, XGLUE introduces 6 new tasks selected from Search, Ads and News scenarios,which makes XGLUE have more practical values. Second, an extended version of Unicoder (Huang et al., 2019) is described and evaluated as a strong cross-lingual pre-trained model baseline on XGLUE for both understanding and generation tasks. We also evaluate the base versions (12-layer) of Multilingual BERT (Devlin et al., 2019), XLM (Conneau and Lample, 2019) and XLM-R (Conneau et al., 2019) for comparison.

2 XGLUE Benchmark

2.1 Pre-training Corpus

We collect two corpora, *Small Corpus* and *Large Corpus*, with different sizes for cross-lingual pre-training. Table 1 lists the data statistics.

2.1.1 Small Corpus (SC)

Multilingual Corpus We extract raw sentences from Wikipedia using *WikiExtractor*. It leads to a 101G multilingual corpus covering 100 languages.

[1] The dataset is available at `https://microsoft.github.io/XGLUE/`, The code and model is available at `https://github.com/microsoft/Unicoder`

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6008–6018,
November 16–20, 2020. ©2020 Association for Computational Linguistics

	Type	# of Languages	Size
Small Corpus	Multilingual	100	101G
	Bilingual	27	146G
Large Corpus	Multilingual	100	2,500G+101G
	Bilingual	27	146G

Table 1: The statistics of two pre-training corpora.

Bilingual Corpus We use an in-house pipeline to extract bilingual sentence pairs from the Web, which leads to a 146G bilingual corpus covering 27 languages, including *Arabic*, *Bulgarian*, *Danish*, *German*, *Greek*, *English*, *Spanish*, *Finnish*, *French*, *Hebrew*, *Hindi*, *Hungarian*, *Indonesian*, *Italian*, *Japanese*, *Korean*, *Dutch*, *Polish*, *Portuguese*, *Russian*, *Swedish*, *Swahili*, *Thai*, *Turkish*, *Urdu*, *Vietnamese* and *Chinese*. All the bilingual pairs are English to another language.

2.1.2 Large Corpus (LC)

Multilingual Corpus Following Wenzek et al. (2019), we construct a clean version of Common Crawl (CC)[2] as the multilingual corpus. First, we use a language identification model trained based on Wikipedia to classify the language of each page in CC. Then, we train a language model for each language using the corresponding part of the Wikipedia corpus, and use it to filter documents as Wenzek et al. (2019) did. We use one CC dump for English and twelve CC dumps for other languages. It leads to a 2,500G multilingual corpus covering 89 languages. We also include the 101G multilingual corpus described in Section 2.1.1.

Bilingual Corpus We reuse the bilingual corpus described in Section 2.1.1. We will add CCMatrix (Schwenk et al., 2019) in the future.

2.2 Downstream Tasks

We select 11 cross-lingual tasks in XGLUE, which are categorized into 3 groups: single-input understanding tasks, pair-input understanding tasks, and generation tasks. For each task, training set is only available in English. In order to obtain a good performance on XGLUE, a model should be able to learn how to do a task well using its English training set, and then transfer this ability to test sets in other languages. Table 2 gives the dataset statistics and Table 3 lists languages covered by all tasks.

2.2.1 Single-input Understanding Tasks

NER We select a subset of the following two NER tasks, CoNLL-2002 NER (Sang, 2002) and CoNLL-2003 NER (Sang and De Meulder, 2003), to form this cross-lingual NER dataset. It covers 4 languages, including *English*, *German*, *Spanish* and *Dutch*, and 4 types of named entities, including *Person*, *Location*, *Organization* and *Miscellaneous* entities that do not belong to the previous three types. F1 score is used as the metric.

POS Tagging (POS) Following (Kim et al., 2017), we select a subset of Universal Dependencies (UD) Treebanks (v2.5) (Zeman et al., 2019), which covers 18 languages. Accuracy (ACC) of the predicted POS tags is used as the metric.

News Classification (NC) This task aims to predict the category given a news article. It covers 5 languages, including *English*, *Spanish*, *French*, *German* and *Russian*. Each labeled instance is a 3-tuple: <news title, news body, category>. The category number is 10. We crawl this dataset from Microsoft News (MSN). Accuracy (ACC) of the multi-class classification is used as the metric.

2.2.2 Pair-input Understanding Tasks

MLQA The MLQA (Lewis et al., 2019b) is a multilingual machine reading comprehension task, which contains QA annotations labeled in 7 languages, including *English*, *Arabic*, *German*, *Spanish*, *Hindi*, *Vietnamese* and *Chinese*. F1 score of the predicted answers is used as the metric.

XNLI We reuse the original XNLI dataset (Conneau et al., 2018) in XGLUE.

PAWS-X The PAWS-X (Yang et al., 2019a) is a paraphrase identification dataset, which extends the Wikipedia portion of the PAWS (Zhang et al., 2019) evaluation to more languages. We select 4 languages, including *English*, *Spanish*, *French* and *German*, from the original dataset and use them in XGLUE. Accuracy (ACC) of the binary classification is used as the metric.

Query-Ad Matching (QADSM) This task aims to predict whether an advertisement (ad) is relevant to an input query. It covers 3 languages, including *English*, *French* and *German*. Each labeled instance is a 4-tuple: <query, ad title, ad description, label>. The label indicates whether the ad is relevant to the query (Good), or not (Bad). We con-

[2] https://commoncrawl.org/.

Task	# of Languages	$\lvert\text{Train}\rvert^{en}$	$\lvert\text{Dev}\rvert^{avg}$	$\lvert\text{Test}\rvert^{avg}$	Metric	Data Source
NER	4	15.0K	2.8K	3.4K	F1	ECI Multilingual Text Corpus
POS	18	25.4K	1.0K	0.9K	ACC	UD Tree-banks (v2.5)
NC*	5	100K	10K	10K	ACC	MSN
MLQA	7	87.6K	0.6K	5.7K	F1	Wikipedia
XNLI	15	433K	2.5K	5K	ACC	MultiNLI Corpus
PAWS-X	4	49.4K	2K	2K	ACC	Wikipedia
QADSM*	3	100K	10K	10K	ACC	Bing
WPR*	7	100K	10K	10K	nDCG	Bing
QAM*	3	100K	10K	10K	ACC	Bing
QG*	6	100K	10K	10K	BLEU-4	Bing
NTG*	5	300K	10K	10K	BLEU-4	MSN

Table 2: 11 downstream tasks in XGLUE. For each task, training set is only available in English. $\lvert\text{Train}\rvert^{en}$ denotes the number of labeled instances in the training set. $\lvert\text{Dev}\rvert^{avg}$ and $\lvert\text{Test}\rvert^{avg}$ denote the average numbers of labeled instances in the dev sets and test sets, respectively. $*$ denotes the corresponding dataset is constructed by this paper.

Task	ar	bg	de	el	en	es	fr	hi	it	nl	pl	pt	ru	sw	th	tr	ur	vi	zh
NER			✓		✓	✓				✓									
POS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
NC*			✓		✓	✓	✓						✓						
MLQA	✓		✓		✓	✓		✓										✓	✓
XNLI	✓	✓	✓	✓	✓	✓	✓	✓					✓	✓	✓	✓	✓	✓	✓
PAWS-X			✓		✓	✓	✓												
QADSM*			✓		✓		✓												
WPR*			✓		✓	✓	✓		✓			✓							✓
QAM*			✓		✓		✓												
QG*			✓		✓	✓	✓		✓			✓							
NTG*			✓		✓	✓	✓						✓						

Table 3: The 19 languages covered by the 11 downstream tasks: *Arabic* (ar), *Bulgarian* (bg), *German* (de), *Greek* (el), *English* (en), *Spanish* (es), *French* (fr), *Hindi* (hi), *Italian* (it), *Dutch* (nl), *Polish* (pl), *Portuguese* (pt), *Russian* (ru), *Swahili* (sw), *Thai* (th), *Turkish* (tr), *Urdu* (ur), *Vietnamese* (vi), and *Chinese* (zh). All these 6 new tasks with $*$ are labeled by human, except es, it and pt datasets in QG (80+% accuracy) are obtained by an in-house QA ranker.

struct this dataset based on Bing. Accuracy (ACC) of the binary classification is used as the metric.

Web Page Ranking (WPR) This task aims to predict whether a web page is relevant to an input query. It covers 7 languages, including *English*, *German*, *French*, *Spanish*, *Italian*, *Portuguese* and *Chinese*. Each labeled instance is a 4-tuple: <query, web page title, web page snippet, label>. The relevance label contains 5 ratings: Perfect (4), Excellent (3), Good (2), Fair (1) and Bad (0). We construct this dataset based on Bing. Normalize Discounted Cumulative Gain (nDCG) is used as the metric.

QA Matching (QAM) This task aims to predict whether a <question, passage> pair is a QA pair. It covers 3 languages, including *English*, *French* and *German*. Each labeled instance is a 3-tuple: <question, passage, label>. The label indicates whether the passage is the answer of the question (1), or not (0). We construct this dataset based on Bing. Accuracy (ACC) of the binary classification is used as the metric.

2.2.3 Generation Tasks

Question Generation (QG) This task aims to generate a question for a given passage. We collect <passage, question> pairs from Bing. It covers 6 languages, including *English*, *French*, *German*, *Spanish*, *Italian* and *Portuguese*. BLEU-4 score is used as the metric.

News Title Generation (NTG) This task aims to generate a proper title for a given news body. We collect <news body, news title> pairs from Microsoft News (MSN). It covers 5 languages, including *German*, *English*, *French*, *Spanish* and *Russian*. BLEU-4 score is used as the metric.

3 Pre-train Unicoder for Cross-lingual Understanding Tasks

We select Unicoder (Huang et al., 2019) as the backbone model. Section 3 introduces a simplified version of Unicoder using two pre-training tasks (MLN and TLM) for cross-lingual understanding tasks. Section 4 describes how to extend Unicoder to cover cross-lingual generation tasks.

The original Unicoder (Huang et al., 2019) includes more pre-training tasks besides MLM and TLM. But to keep the baseline pre-trained model simple and to reduce the experimental cost, we just use MLM and TLM in this paper. It means for understanding tasks, Unicoder is almost equal to XLM, except some hyper-parameter differences.

3.1 Masked Language Model (MLM)

Following Devlin et al. (2019), this task extends the masked language model task to multiple languages. At each iteration, a batch is composed of sentences sampled from different languages. The sampling probability of a language l_i is defined as $\lambda_{l_i} = p_{l_i}^{\alpha} / \sum_{l_i} p_{l_i}^{\alpha}$, where p_{l_i} is the percentage of the language l_i in the entire corpus, the smoothing factor α is set to 0.3. For each batch, we randomly sample 15% of the words and replace them with (i) a special symbol [MASK], (ii) a random token or (iii) keep them unchanged with probability 80%, 10% and 10%, respectively. For each token, we only use its token embedding and position embedding, and discard segment embedding and language embedding.

3.2 Translation Language Model (TLM)

Following Conneau and Lample (2019), this task extends the MLM task to bilingual corpus. Given a bilingual sentence pair, TLM first concatenates them into a single sentence, and then masks words using the same strategy of MLM. The pre-trained model learns to recover each masked word based on the bilingual context. We follow MLM to sample language pairs in each batch with $\alpha = 0.3$.

4 Pre-train Unicoder for Cross-lingual Generation Tasks

The encoder-decoder architecture is employed to extend Unicoder to generation tasks, where the BPE embeddings are shared between encoder and decoder. Two separate generative tasks are proposed for Unicoder pre-training: *Multilingual Denoising Auto-Encoding (xDAE)* and *Multilingual Future N-gram Prediction (xFNP)*.

4.1 Multilingual Denoising Auto-Encoding (xDAE)

Motivated by BART (Lewis et al., 2019a), xDAE aims to predict the original text $X = (x_1, x_2, ..., x_{|X|}) \in l_i$ from a language l_i based on its corrupted form $c(X)$, where $c(X)$ is a noising function that corrupts an input text X as its output.

Four different text noising strategies for $c(\cdot)$ are explored in this paper. (1) Shuffle the input text X by adding a noise $\alpha \sim \mathrm{U}(0,3)$ to the input indices and then re-ordering X based on the rank of the noised indices. (2) Drop words with a probability of 0.1. (3) Replace 10% of the input words in X with the [MASK] symbol. (4) Sample a number of token spans from X with span lengths drawn from a Poisson distribution ($\lambda = 3$), and then replace each token span with a single [MASK] token. Here, 0-length spans correspond to the insertion of [MASK] tokens. Based on the performance of different noising strategies (Table 10), we select (4) and use it in pre-training. We leave finding better text noising strategies for future work.

We train Unicoder using this task by maximizing the following loss function $\mathcal{L}_{xDAE}$:

$$\mathcal{L}_{xDAE} = \sum_{l_i \in L} \sum_{X \in l_i} \sum_{t=1}^{|X|} \log p(x_t | x_{<t}, c(X))$$

where $L = l_1, ..., l_N$ denotes N languages, X is an instance in the i^{th} language l_i, $p(x_t|x_{<t}, c(X))$ denotes the probability of generating a single token x_t at time step t given $c(X)$ and $x_{<t}$.

4.2 Multilingual Future N-gram Prediction (xFNP)

Motivated by ProphetNet (Yan et al., 2020), xFNP introduces a future n-gram prediction mechanism to natural language generation. It encourages the model to plan for the future tokens explicitly and prevents over-fitting on strong local correlations.

Given an input text $X = (x_1, x_2, ..., x_{|X|}) \in l_i$ from a language l_i, we randomly mask k token spans of X to generate the masked text $X^{'}$ as the input, and concatenate all masked token spans into Y as the output. Details of this mask strategy are described in Section 6.1. After this, xFNP first encodes $X^{'}$ to H_{enc} with the encoder:

$$H_{enc} = \text{Encoder}(X^{'})$$

Then, instead of predicting the next token only at each time step, xFNP generates n future tokens simultaneously at time step t with the decoder:

$$\begin{aligned} p(y_t|y_{<t}, X^{'}), ..., p(y_{t+n-1}|y_{<t}, X^{'}) \\ = \text{Decoder}(y_{<t}, H_{enc}) \end{aligned}$$

Following Yan et al. (2020), we set $n = 2$.

We train Unicoder using this task by maximizing the following loss function $\mathcal{L}_{xFNP}$:

$$\begin{aligned} \mathcal{L}_{xFNP} = \sum_{l_i \in L} \sum_{X \in l_i} \{\alpha_0 \cdot \sum_{t=1}^{|Y|} \log p(y_t|y_{<t}, X^{'}) \\ + \alpha_1 \cdot \sum_{t=1}^{|Y|-1} \log p(y_{t+1}|y_{<t}, X^{'})\} \end{aligned}$$

where $X^{'}$ and Y are generated from X based on the method mentioned above. Following Yan et al. (2020), we set $\alpha_0 = \alpha_1 = 1$.

5 Experiments

5.1 Data Labeling

For tasks QADSM, WPR, QAM and QG, we label the data on an Microsoft internal crowdsourcing platform. Each labeler must learn the guideline and pass the labeling test. Each sample is labeled by three labeler. We only keep the samples with two or three labeler have same label.

For tasks NC and NTG, we directly use the category label on MSN website. All the category label on MSN is review by human.

5.2 Experimental Settings

Understanding Tasks The hyper-parameters are set as follows: 768 hidden units, 12 heads, GELU activation, a dropout rate of 0.1, 512 max input length, 12 layers in encoder.

In the pre-training stage, we first initialize Unicoder$_{LC}$ with XLM-R$_{base}$ (Conneau et al., 2019), and then run continue pre-training with the accumulated 8,192 batch size with gradients accumulation. We use Adam Optimizer with a linear warm-up and set the learning rate to 3e-5. We select different understanding tasks randomly in different batches. This costed 12 days on 16 V100.

In the fine-tuning stage, the batch size is set to 32. We use Adam Optimizer (Kingma and Ba, 2014) with warm-up and set the learning rate to 5e-6. For all sentence classification tasks, we fine-tune 10 epochs. For POS Tagging and NER, we fine-tune 20 epochs. And for POS Tagging, we set the learning rate to 2e-5. For MLQA, we set the learning rate to 3e-5, batch size to 12 and train 2 epochs following BERT for SQuAD. After each epoch, we test the fine-tuned model on the dev sets of all languages. We select the model with the best average result on the dev sets of all languages.

Generation Tasks We evaluate Unicoder$_{SC}^{xDAE}$ and Unicoder$_{SC}^{xFNP}$ as two separate models.

For Unicoder$_{SC}^{xDAE}$, the hyper-parameters are set as follows: 768 hidden units, 12 heads, GELU activation, a dropout rate of 0.1, 512 max input length, 12 layers in encoder, 12 layers in decoder.

In the pre-training stage, we first initialize encoder and decoder with XLM-R, and then run continue pre-training with 1,024 batch size. We use Adam optimizer with warm-up and set the learning rate to 2e-4. This costed 10 days on 16 V100.

In the fine-tuning stage, the batch size is 1024. We use Adam Optimizer with learning rate 1e-5 and warm-up steps 2000.

For Unicoder$_{SC}^{xFNP}$, the hyper-parameters are set as follows: 1,024 hidden size, 12 layers in encoder, 12 layers in decoder, 512 max input length.

In the pre-training stage, we pre-train the model from scratch and follow ProphetNet (Yan et al., 2020) to randomly mask a continuous span (with a fixed length 9) in every 64 tokens. About 15% of the tokens in original sequence are masked in this step. We use a special symbol [MASK] to replace 80% of the masked tokens, keep 10% unchanged, and random replace 10% of the masked tokens. We set the batch size to 1,024, training steps to 350,000. The learning rate is set to 1e-4. We set the number of future tokens n to 2.

In the fine-tuning stage, we use Adam Optimizer and set the learning rate to 1e-4. We set the batch size to 64 and the warm-up steps to 1,000.

5.3 Main Result

7 cross-lingual pre-trained models are evaluated on XGLUE and compared in Table 4: 12-layer M-BERT (Devlin et al., 2019) trained on Wikipedia corpus for 102 languages, 12-layer XLM (Conneau and Lample, 2019) trained on Wikipedia and bilingual corpora for 15 languages, 12-layer XLM-R$_{base}$ (Conneau et al., 2019) trained on Common Crawl corpus for 100 languages, 12-layer Unicoder$_{SC}$ trained on small corpus for 100 languages, 12-layer Unicoder$_{LC}$ trained on large corpus for 100 languages, 12-layer Unicoder$_{SC}^{xDAE}$ and

Task	Model	ar	bg	de	el	en	es	fr	hi	it	nl	pl	pt	ru	sw	th	tr	ur	vi	zh	AVG
NER	M-BERT	-	-	69.2	-	90.6	75.4	-	-	-	77.9	-	-	-	-	-	-	-	-	-	78.2
	XLM-R$_{base}$	-	-	70.4	-	90.9	75.2	-	-	-	79.5	-	-	-	-	-	-	-	-	-	79.0
	Unicoder$_{LC}$	-	-	71.8	-	91.1	74.4	-	-	-	81.6	-	-	-	-	-	-	-	-	-	**79.7**
POS	M-BERT	52.4	85.0	88.7	81.5	95.6	86.8	87.6	58.4	91.3	88.0	81.8	88.3	78.8	-	43.3	69.2	53.8	54.3	58.3	74.7
	XLM-R$_{base}$	67.3	88.8	92.2	88.2	96.2	89.0	89.9	74.5	92.6	88.5	85.4	89.7	86.9	-	57.9	72.7	62.1	55.2	60.4	**79.8**
	Unicoder$_{LC}$	68.6	88.5	92.0	88.3	96.1	89.1	89.4	69.9	92.5	88.9	83.6	89.8	86.7	-	57.6	75.0	59.8	56.3	60.2	79.6
NC	M-BERT	-	-	82.6	-	92.2	81.6	78.0	-	-	-	-	-	79.0	-	-	-	-	-	-	82.7
	XLM-R$_{base}$	-	-	84.5	-	91.8	83.2	78.2	-	-	-	-	-	79.4	-	-	-	-	-	-	83.4
	Unicoder$_{LC}$	-	-	84.2	-	91.7	83.5	78.5	-	-	-	-	-	79.7	-	-	-	-	-	-	**83.5**
MLQA	M-BERT	50.9	-	63.8	-	80.5	67.1	-	47.9	-	-	-	-	-	-	-	-	-	59.5	55.4	60.7
	XLM-R$_{base}$	56.4	-	62.1	-	80.1	67.9	-	60.5	-	-	-	-	-	-	-	-	-	67.1	61.4	65.1
	Unicoder$_{LC}$	57.8	-	62.7	-	80.6	68.6	-	62.7	-	-	-	-	-	-	-	-	-	67.5	62.1	**66.0**
XNLI	M-BERT	64.9	68.9	71.1	66.4	82.1	74.3	73.8	60.0	-	-	-	-	69.0	50.4	55.8	61.6	58.0	69.5	69.3	66.3
	XLM†	73.1	77.4	77.8	76.6	85.0	78.9	78.7	69.6	-	-	-	-	75.3	68.4	73.2	72.5	67.3	76.1	76.5	75.1
	XLM-R$_{base}$	72.1	77.5	77.0	75.9	84.6	79.2	78.2	69.8	-	-	-	-	75.5	64.7	71.6	72.9	65.1	74.8	73.7	74.2
	Unicoder$_{SC}$	68.5	73.2	71.6	71.6	82.9	75.0	74.7	66.0	-	-	-	-	70.6	64.1	67.0	68.7	62.5	71.2	69.7	70.5
	Unicoder$_{LC}$	73.9	78.5	78.2	77.3	85.4	79.8	79.2	70.1	-	-	-	-	76.7	67.4	71.8	73.8	66.3	75.9	74.7	**75.3**
PAWS-X	M-BERT	-	-	82.9	-	94.0	85.9	86.0	-	-	-	-	-	-	-	-	-	-	-	-	87.2
	XLM-R$_{base}$	-	-	86.9	-	94.4	88.0	88.7	-	-	-	-	-	-	-	-	-	-	-	-	89.5
	Unicoder$_{LC}$	-	-	87.4	-	94.9	88.8	89.3	-	-	-	-	-	-	-	-	-	-	-	-	**90.1**
QADSM	M-BERT	-	-	60.3	-	68.3	-	64.1	-	-	-	-	-	-	-	-	-	-	-	-	64.2
	XLM-R$_{base}$	-	-	65.8	-	71.7	-	68.3	-	-	-	-	-	-	-	-	-	-	-	-	**68.6**
	Unicoder$_{LC}$	-	-	64.6	-	71.8	-	68.7	-	-	-	-	-	-	-	-	-	-	-	-	68.4
WPR	M-BERT	-	-	76.6	-	78.1	75.3	74.2	-	70.1	-	-	76.6	-	-	-	-	-	-	64.5	73.5
	XLM-R$_{base}$	-	-	77.6	-	78.2	76.0	74.4	-	70.7	-	-	77.3	-	-	-	-	-	-	63.9	73.8
	Unicoder$_{LC}$	-	-	77.2	-	78.4	75.7	74.9	-	70.3	-	-	77.4	-	-	-	-	-	-	64.4	**73.9**
QAM	M-BERT	-	-	64.7	-	67.5	-	66.0	-	-	-	-	-	-	-	-	-	-	-	-	66.1
	XLM-R$_{base}$	-	-	68.1	-	69.3	-	67.8	-	-	-	-	-	-	-	-	-	-	-	-	68.4
	Unicoder$_{LC}$	-	-	68.4	-	69.9	-	68.4	-	-	-	-	-	-	-	-	-	-	-	-	**68.9**
AVG^2_U	M-BERT																				72.6
	XLM-R$_{base}$																				75.8
	Unicoder$_{LC}$																				**76.2**
QG	M-BERT	-	-	0.1	-	7.8	0.1	0.1	-	0.2	-	-	0.1	-	-	-	-	-	-	-	1.4
	XLM-R$_{base}$	-	-	0.1	-	6.0	0.0	0.0	-	0.1	-	-	0.0	-	-	-	-	-	-	-	1.0
	Unicoder$^{xDAE}_{SC}$	-	-	3.0	-	14.0	12.4	4.2	-	15.8	-	-	8.3	-	-	-	-	-	-	-	9.6
	Unicoder$^{xFNP}_{SC}$	-	-	3.7	-	13.9	14.8	4.9	-	17.0	-	-	9.5	-	-	-	-	-	-	-	**10.6**
NTG	M-BERT	-	-	0.7	-	9.0	0.4	0.4	-	-	-	-	-	0.0	-	-	-	-	-	-	2.1
	XLM-R$_{base}$	-	-	0.6	-	8.1	0.4	0.3	-	-	-	-	-	0.0	-	-	-	-	-	-	1.9
	Unicoder$^{xDAE}_{SC}$	-	-	6.8	-	15.6	9.0	8.7	-	-	-	-	-	7.7	-	-	-	-	-	-	9.6
	Unicoder$^{xFNP}_{SC}$	-	-	7.5	-	15.8	11.9	9.9	-	-	-	-	-	8.4	-	-	-	-	-	-	**10.7**
AVG^2_G	M-BERT																				1.8
	XLM-R$_{base}$																				1.5
	Unicoder$^{xDAE}_{SC}$																				9.6
	Unicoder$^{xFNP}_{SC}$																				**10.7**

Table 4: The overall evaluation results on XGLUE. We use M-BERT (Devlin et al., 2019), XLM (Conneau and Lample, 2019) and XLM-R$_{base}$ (Conneau et al., 2019) as baselines. Unicoder$_{SC}$ and Unicoder$_{LC}$ are pre-trained using small corpus and large corpus, respectively. Unicoder$^{xDAE}_{SC}$ and Unicoder$^{xFNP}_{SC}$ are pre-trained by xDAE and xFNP for 100 languages, respectively. For the results of M-BERT/XLM-R on generation tasks, we initialize the encoder-decoder model with M-BERT/XLM-R and fine-tune it on each downstream task without pre-training. **All models are (12-layer) based ones.** Given a task, each pre-trained model is fine-tuned using its English training set only, and then applied to all test sets in different languages. AVG^2_U and AVG^2_G denote the average score of the average scores on 9 understanding tasks and 2 generation tasks, respectively.

Pivot	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	AVG
en	**85.4**	79.2	79.8	78.2	77.3	78.5	76.7	73.8	73.9	75.9	71.8	74.7	70.1	67.4	66.3	75.3
fr	84.0	79.9	80.3	78.8	77.4	79.2	77.0	73.6	73.7	76.7	72.7	75.3	73.0	67.4	68.3	75.8
es	84.5	**80.2**	**81.2**	79.7	78.2	79.2	77.6	74.5	74.8	77.0	72.8	76.2	73.2	67.7	**69.6**	**76.4**
de	83.5	79.1	80.1	**80.2**	77.9	78.6	77.0	74.9	74.6	76.1	73.3	76.2	73.1	67.7	68.9	76.1
el	83.8	80.1	81.0	78.6	**79.6**	79.3	77.0	74.2	74.9	77.1	73.5	75.9	72.7	69.1	69.1	**76.4**
bg	83.5	79.6	80.4	79.1	77.9	**80.5**	77.9	74.9	73.9	76.5	73.9	75.6	72.8	68.6	68.9	76.3
ru	84.1	79.9	79.9	78.8	77.5	79.9	**78.1**	73.9	74.5	77.1	73.8	75.7	73.1	68.5	69.0	76.2
tr	83.3	78.4	79.6	78.4	77.5	79.2	77.5	**77.1**	74.2	77.1	74.5	76.5	73.7	69.3	70.3	**76.4**
ar	83.2	78.9	79.5	77.6	77.4	78.6	77.0	75.4	**76.8**	76.8	74.0	76.0	73.0	69.5	69.3	76.2
vi	83.2	78.6	79.1	77.7	76.6	78.9	77.5	75.3	74.7	**78.5**	73.5	**76.8**	73.1	67.8	69.0	76.0
th	82.5	78.5	79.1	77.8	77.1	78.3	76.7	75.0	74.3	76.9	**76.4**	76.2	72.9	68.4	69.7	76.0
zh	81.6	78.2	77.9	77.1	76.0	77.9	76.2	73.7	73.7	75.8	73.6	76.6	71.7	67.4	68.3	75.1
hi	81.8	78.5	79.2	76.7	77.2	78.2	76.2	74.5	73.9	76.4	71.7	75.2	**73.8**	68.2	68.5	75.3
sw	82.0	77.6	78.8	77.2	76.5	77.7	76.2	74.4	74.3	76.3	74.0	75.2	72.2	**71.4**	69.5	75.6
ur	76.7	72.5	74.1	72.6	72.1	73.9	72.7	69.7	69.7	72.8	70.1	72.4	69.0	66.0	67.5	71.5

Table 5: Impacts of different pivot languages on XNLI. Given each pivot language, the corresponding fine-tuned XNLI results on all languages are listed in the same row. Each **bolded number** is the best result in that column.

Pivot	en	es	fr	de	ru	AVG
en	**15.6/15.8**	9.0/11.9	8.7/9.9	6.8/7.5	7.7/8.4	9.6/10.7
es	7.8/8.8	**17.1/17.1**	10.6/10.9	7.6/8.0	8.0/8.6	10.2/10.7
fr	8.2/8.7	11.4/12.5	**19.4/20.9**	8.3/8.2	7.6/7.8	**11.0/11.6**
de	8.2/8.6	9.9/11.2	9.5/10.2	**14.1/13.7**	8.4/8.0	10.0/10.3
ru	6.9/7.4	9.3/10.8	8.8/9.9	6.9/7.0	**16.6/16.7**	9.7/10.4

Table 6: Impacts of different pivot languages on NTG. Unicoder$_{SC}^{xDAE}$/Unicoder$_{SC}^{xFNP}$ evaluated by BLEU-4.

12-layer Unicoder$_{SC}^{xFNP}$ trained on Wikipedia corpus for 100 languages. Given a downstream task, each pre-trained model is fine-tuned using its English training set and then applied to all test sets in different languages. Note that, all results are reproduced by this paper, except the XLM† results on XNLI are from Conneau and Lample (2019).

We find (1) Unicoder$_{LC}$ performs slightly better than M-BERT and XLM-R$_{base}$ on the 9 understanding tasks, as it is pre-trained based on multilingual and bilingual corpora at the same time and uses TLM; (2) Unicoder$_{LC}$ performs better than Unicoder$_{SC}$, as it is pre-trained based on the larger corpus; (3) Unicoder$_{SC}^{xDAE}$ and Unicoder$_{SC}^{xFNP}$ show good cross-lingual transfer capabilities and perform significantly better than M-BERT and XLM-R$_{base}$ on the 2 generation tasks. It proves the importance of introducing generation tasks into pre-training for cross-lingual text generation; (4) Unicoder$_{SC}^{xFNP}$ performs slightly better than Unicoder$_{SC}^{xDAE}$. But it is not a fair comparison, because they use different text denoising tasks (sentence prediction vs. span prediction) and different generation mechanisms (single-token prediction vs. multi-token prediction). We leave combining these two tasks for future work.

5.4 Ablation Study

5.4.1 Pivot-language Fine-tuning

We define pivot-language (*pl*) fine-tuning as fine-tune a pre-trained model for a downstream task using its labeled data in a pivot language (e.g. English) and then apply the fine-tuned model to all languages. Table 4 chooses English as the pivot language, as all tasks in XGLUE have labeled data in English. But is English always the optimal choice? Will the results become better, if we do fine-tuning using other pivot languages?

To answer these questions, we evaluate Unicoder on XNLI and NTG using different pivot languages in fine-tuning and list comparison results in Table 5 and Table 6, respectively. (1) For each test set in language l_i in Table 5 and Table 6, its best result is often achieved when the model is fine-tuned using l_i as the pivot language; (2) For XNLI in Table 5, the best pivot languages are Spanish (es), Greek (el) and Turkish (tr), rather than English (en). For NTG in Table 6, the best pivot language is French (fr) for both Unicoder$_{SC}^{xDAE}$ and Unicoder$_{SC}^{xFNP}$. It means the average quality of a cross-lingual pre-trained model could be further improved on a downstream task, by selecting a specific pivot language in fine-tuning.

5.4.2 Multi-language Fine-tuning

We define multi-language (*ml*) fine-tuning as fine-tune a pre-trained model for a downstream task using all its available labeled data in different languages. We evaluate Unicoder on XNLI and NTG using this fine-tuning method and list evaluation results in Table 7 and Table 8, respectively.

We find multi-language fine-tuning can achieve better results than pivot-language fine-tuning on both XNLI and NTG. It means the average quality of a cross-lingual pre-trained model could be significantly improved on a downstream task, by using combined labeled data in multiple languages.

5.4.3 Multi-task Fine-tuning

We define multi-task (*mt*) fine-tuning as fine-tune a pre-trained model for multiple downstream tasks using their combined labeled data. To reduce the experimental cost, we evaluate Unicoder on 5 understanding tasks: XNLI, PAWS-X, NC, QAM and QADSM, using their merged English labeled data in fine-tuning. Results are listed in Table 9.

We find PAWS-X and QADSM can benefit from the joint fine-tuning strategy, but XNLI, NC and QAM cannot. We leave discovering relationships between different tasks for better downstream task fine-tuning for future work.

5.4.4 Impacts of Text Noising Strategies

We investigate the impacts of different text noising strategies (Section 4.1) in Unicoder$_{SC}^{xDAE}$, and list comparison results in Table 10, where (1)+(2)+(3) denotes the result of using the first three strategies in pre-training, (4) denotes the result of using the last strategy in pre-training, (1)+(2)+(3)+(4) denotes the result of using all strategies in pre-training. To reduce experiment cost, we set max sequence length to 256 and only train 60K steps. We find that (4) can achieve the best average result on NTG. So all results of Unicoder$_{SC}^{xDAE}$ reported in this paper is pre-trained using (4) only.

	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	AVG
XLM-R$_{base}$ (pl)	84.6	78.2	79.2	77.0	75.9	77.5	75.5	72.9	72.1	74.8	71.6	73.7	69.8	64.7	65.1	74.2
XLM-R$_{base}$ (ml)	85.7	81.5	**82.5**	81.2	79.7	81.7	**80.0**	**79.0**	77.1	80.1	77.9	79.2	**76.5**	73.0	71.3	79.1
Unicoder$_{LC}$ (pl)	85.4	79.2	79.8	78.2	77.3	78.5	76.7	73.8	73.9	75.9	71.8	74.7	70.1	67.4	66.3	75.3
Unicoder$_{LC}$ (ml)	**85.8**	**81.9**	82.3	**81.5**	**80.8**	**82.0**	79.9	78.7	**78.1**	**80.2**	**78.4**	**79.3**	76.2	**73.2**	**72.4**	**79.4**

Table 7: Impact of multi-language fine-tuning on XNLI. pl and ml denote pivot-language fine-tuning (English as pivot) and multi-language fine-tuning, respectively.

Model	en	es	fr	de	ru	AVG
Unicoder$_{SC}^{xDAE}$ (pl)	15.6	9.0	8.7	6.8	7.7	9.6
Unicoder$_{SC}^{xDAE}$ (ml)	**18.5**	**18.3**	**28.2**	**15.5**	**33.4**	**22.8**
Unicoder$_{SC}^{xFNP}$ (pl)	**15.8**	11.9	9.9	7.5	8.4	10.7
Unicoder$_{SC}^{xFNP}$ (ml)	15.6	**17.1**	**19.1**	**13.9**	**15.8**	**16.3**

Table 8: Impact of multi-language fine-tuning on NTG. pl and ml denote pivot-language fine-tuning (English as pivot) and multi-language fine-tuning, respectively. BLUE-4 is the metric.

Model	XNLI	PAWS-X	NC	QAM	QADSM	AVG
Unicoder$_{LC}$ (pl)	**75.3**	90.1	**83.5**	**68.9**	68.4	**77.2**
Unicoder$_{LC}$ (mt)	74.4	**90.2**	83.4	68.7	**69.0**	77.1

Table 9: Impacts of multi-task fine-tuning on XNLI, PAWS-X, NC, QAM and QADSM. pl and mt denote pivot-language fine-tuning (English as pivot) on each task and multi-task fine-tuning, respectively.

Text Noising Strategy	en	es	fr	de	ru	AVG
(1)+(2)+(3)	14.6	8.5	7.4	6.0	7.4	8.8
(4)	14.8	**8.7**	**7.5**	**6.7**	**8.2**	**9.2**
(1)+(2)+(3)+(4)	**15.2**	7.9	7.3	6.2	7.7	8.9

Table 10: Impact of different text noising strategies on NTG using pivot-language fine-tuning (English as pivot). BLUE-4 is the metric.

Model	fr	zh	AVG
XNLG (Chi et al., 2019)	36.3	38.9	37.6
Unicoder$_{SC}^{xDAE}$	**37.9**	**42.2**	**40.1**

Table 11: The zero-shot results on Abstractive Summarization. Unicoder$_{SC}^{xDAE}$ and XNLG are fine-tuned using English labeled data. ROUGE-L is the metric.

We also compare Unicoder$_{SC}^{xDAE}$ with XNLG (Chi et al., 2019) on the Abstractive Summarization task. For fairly comparison, we implement xDAE in same code base and use same pre-training languages as XNLG. The zero-shot comparison results are listed in Table 11. We can see that by using xDAE only in pre-training, Unicoder$_{SC}^{xDAE}$ can outperform XNLG significantly, which is pre-trained using 4 tasks including MLM, DAE, XMLM and XAE. It verifies the effectiveness of the fourth text noising strategy described in Section 4.1 for generation tasks.

6 Related Work

Dataset GLUE (Wang et al., 2019) includes 9 natural language understanding tasks that are labeled in English only. Comparing to GLUE, XGLUE not only expands task annotations to multiple languages, but also includes natural language generation tasks. XNLI (Conneau et al., 2018), NER (Sang, 2002; Sang and De Meulder, 2003), POS Tagging (Kim et al., 2017), MLQA (Lewis et al., 2019b) and PAWS-X (Yang et al., 2019a) are 5 multilingual datasets built for specific tasks. XGLUE not only includes these 5 existing tasks, but also introduces 6 new tasks selected from real-world scenarios (i.e., Search, Ads and News). This makes XGLUE have more practical values. XTREME (Hu et al., 2020) is a concurrent work of XGLUE. Comparing to it, XGLUE includes both understanding and generation tasks, which, to the best of our knowledge, is the first attempt in the cross-lingual dataset construction efforts.

Cross-lingual Pre-trained Model Multilingual BERT (M-BERT) (Devlin et al., 2019) performs pre-training based on the multilingual corpus with the masked language model task. By sharing the model parameters and the vocabulary for all languages, M-BERT can obtain the cross-lingual capability over 102 languages. XLM (Conneau and Lample, 2019) performs cross-lingual pre-training based on multilingual corpus and bilingual corpus, by introducing the translation language model task into pre-training. Based on XLM, Unicoder (Huang et al., 2019) uses more cross-lingual pre-training tasks and achieves better results on XNLI. XLM-R (Conneau et al., 2019) is a RoBERTa (Liu et al., 2019)-version XLM without using translation language model in pre-training. It is trained based on a much larger multilingual corpus (i.e. Common Crawl) and become the new state-of-the-art on XNLI. In this paper, we use both the Common Crawl corpus and the bilingual corpus, aiming to build a stronger baseline model on XGLUE. BART (Lewis et al., 2019a) and ProphetNet (Yan et al., 2020) are two latest generative pre-trained models. We borrow ideas from these two works and extend Unicoder to cross-lingual generation tasks, which goes a step further to verify and explore different text generation approaches in the cross-lingual scenario.

7 Conclusion

We present XGLUE as a new cross-lingual benchmark and conduct comprehensive evaluations with interesting findings observed. We thank STC-A NLP, Bing Answers, Bing Ads, Bing Relevance and Microsoft News for providing the datasets.

References

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Uniter: Learning universal image-text representations. *arXiv*.

Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. 2019. Cross-lingual natural language generation via pre-training. In *AAAI*.

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

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *NeurIPS*.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *NeurIPS*.

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 generalization. *arXiv*.

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

Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. 2017. Cross-lingual transfer learning for POS tagging without cross-lingual resources. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2832–2838, Copenhagen, Denmark. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019a. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019b. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Zhou Zhou. 2020. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *AAAI*.

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. *arXiv preprint arXiv:1907.11692*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NeurIPS*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *arXiv*.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Tjong Kim Sang. 2002. Ef: Introduction to the conll-2002 shared task. In *Proceedings of the 6th Conference on Natural Language Learning*.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *ICLR*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzman, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019a. Paws-x: A cross-lingual adversarial dataset for paraphrase identification. *arXiv preprint arXiv:1908.11828*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. *NeurIPS*.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Noëmi Aepli, Željko Agić, Lars Ahrenberg, Gabrielė Aleksandravičiūtė, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Johannes Heinecke, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Takumi Ikeda, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Kamil Kopacewicz, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Maria Liovina, Yuan Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Tomohiko Morioka, Shinsuke Mori, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Luong Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Daria Petrova, Slav Petrov, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Takaaki Tanaka, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Manying Zhang, and Hanzhi Zhu. 2019. Universal dependencies 2.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. *AAAI*.

A The fine-tune parameters of Unicoder on XGLUE.

Task	batch size	epoch number	learning rate
NER	32	20	5e-6
POS	32	20	5e-6
NC	32	10	5e-6
MLQA	12	2	3e-5
XNLI	32	10	5e-6
PAWS-X	32	10	5e-6
QADSM	32	10	5e-6
WPR	32	10	5e-6
QAM	32	10	5e-6

Table 12: The fine-tune parameters of understanding tasks.

Task	Model	batch size	learning rate	warm up steps
QG	Unicoder$_{SC}^{xDAE}$	64	1e-4	1000
NTG	Unicoder$_{SC}^{xDAE}$	64	1e-4	1000
QG	Unicoder$_{SC}^{xFNP}$	1024	1e-5	2000
NTG	Unicoder$_{SC}^{xFNP}$	1024	1e-5	2000

Table 13: The fine-tune parameters of generation tasks.