A Brief Introduction to Large Multilingual Language Models

Tianjian Li*

Johns Hopkins University

July 26, 2022

1 Pretrained Language Models

1.1 BERT

In 2018, Jacob Devlin and his team at Google AI published a paper named "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" [4]. BERT has beaten the current best performance on the GLUE[24] benchmark by a very large margin, and has become the backbone structure of language models nowadays.[19][11][20] There are a few reasons that account for the universal effectiveness of BERT, but the two main reasons are:

1. BERT stacks multiple Transformer[22] encoder blocks, which adopts the attention mechanism to capture long range dependencies, allowing each word to dynamically extract syntactic and semantic meaning from every other word. Building on the Transformers, BERT also made a few engineering choices that serves its pretrain objective, including using the sum of the token embedding, segment embedding and positional embedding as the final input representation of a token[1].

2. BERT proposes a pretrain-finetune paradigm, which first trains the model with a unsupervised objective, giving the model parameters better initialization to be finetuned with a supervised objective on downstream tasks with labelled data. Moreover, BERT proposes a **Masked Language Modeling(MLM)** self-supervised(or unsupervised) objective which is effective to teach the model to encode deep bidirectional dependencies.

Note that BERT also proposes a **Next Sentence Prediction(NSP)** unsupervised task, in which the model is given a pair of sentences to detect whether or not the second one is the next sentence of the first one. However subsequent studies[11][20] have confirmed that NSP is not particularly useful, if not detrimental to the performance of downstream tasks, therefore I choose to omit the NSP task when accounting for BERT's effectiveness.

1.2 Masked Language Modeling

Language Modeling aims to calculate the probability density function of a word conditioning on its context. For example, if we are trying to calculate which words is most likely to appear after the context "I ate an sweet apple, it was", a pretrained language model would give us the probability of different words appearing after "I ate an sweet apple, it was". Mathematically speaking, given a context C, a language model aims to compute the probability distribution of the next word X.

$$P(X_i = x|C = (x_1, x_2, ...x_{i-1}))$$

*tli104@jhu.edu

[1] A token is similar to a word, only the longer words are divided into sub-words to reduce the amount of vocabulary and learn the meaning of prefixes and suffixes

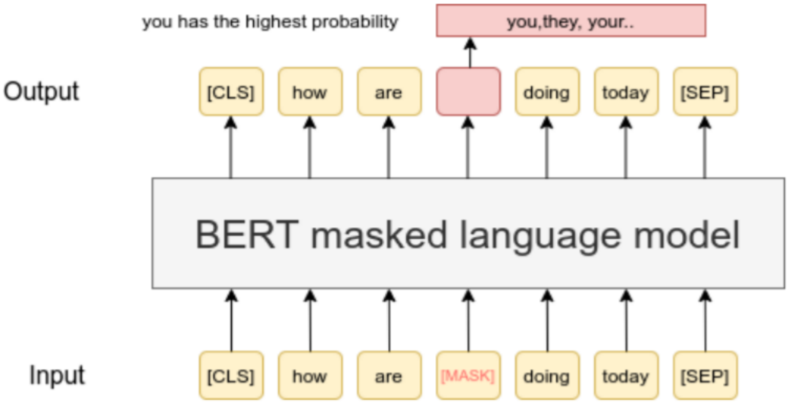


Figure 1: MLM illustration

You might notice that here the next word is only allowed to be conditioned on previous words, but what if the word we are trying to model heavily depends on the following words. For example, if we are trying to model the word that goes after ”The following story I am about to tell you is”, we cannot give an accurate prediction of the word if we remain agnostic about the following story. **Masked Language Modeling** aims to solve this issue by replacing the original word with a [MASK] token. While we are modeling the probability distribution of the [MASK] token, we are allowed to condition on bidirectional context:

$$P(X_i = x|C = (x_1, x_2, ...x_{i-1}, x_{i+1}, x_{i+2}, ...x_N))$$

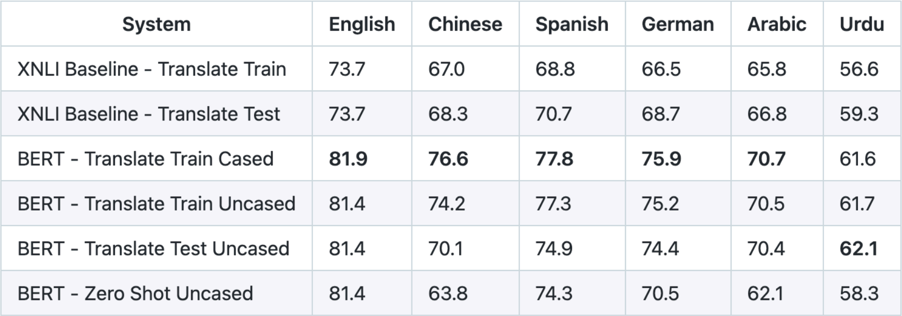
By allowing our language model to learn bidirectionally, BERT has quickly revolutionized the realm of pretraining in NLP, and has cultivated many subsequent work. Models that are built on BERT usually include BERT in its model name. RoBERTa[11] removes the NSP objective and uses dynamic masking, ALBERT[10] aims the reduce the number of parameters, and DeBERTa[7] aims to disentangle the attention for positional embeddings and for word embeddings. However, all of these models are trained on English only data, which limits its usefulness since the majority of person on earth does not speak English. In the next section I will show that BERT not only works on English, but also can be trained on multilingual corpora and possesses an amazing transfer learning ability.

2 The Curse of Multilinguality

2.1 mBERT and Zero-Shot Transfer

The English-only BERT have beaten the state-of-the-art on the GLUE benchmark by a very large margin, so researchers extended by pretraining BERT on large multilingual corpora, resulting in a model named mBERT(multilingual BERT). Superizingly, language models trained on multilingual corpora have this amazing ability called zero-shot transfer: pretrained models are capable of performing natural language inference(NLI) and natural language understanding(NLU) tasks in different languages by only being fine-tuned on English. Moreover, the zero-shot performance are often comparable to fully supervised methods, albeit still behind a few points. From figure 2 we can see that the performance of mBERT zero-shot(the last row) is comparable to the best Translate-Train(Translating the training data to different languages) performance, which is a supervised method that requires manually or machine translating data. The zero-shot setting is desirable because acquiring high quality data in different languages is often expensive. Therefore we want our model can be taught to generalize from English to other languages and still achieve a high performance. After all, humans excel in generalizing knowledge from one language to another, and we wish to let computers acquire that ability.

Wu and Dredze[26] closely examines the multilinguality of mBERT[4] by extending the downstream task from



System	English	Chinese	Spanish	German	Arabic	Urdu
XNLI Baseline - Translate Train	73.7	67.0	68.8	66.5	65.8	56.6
XNLI Baseline - Translate Test	73.7	68.3	70.7	68.7	66.8	59.3
BERT - Translate Train Cased	**81.9**	**76.6**	**77.8**	**75.9**	**70.7**	61.6
BERT - Translate Train Uncased	81.4	74.2	77.3	75.2	70.5	61.7
BERT - Translate Test Uncased	81.4	70.1	74.9	74.4	70.4	**62.1**
BERT - Zero Shot Uncased	81.4	63.8	74.3	70.5	62.1	58.3

Figure 2: mBERT performance on XNLI, pasted from mBERT Github repo

only XNLI[2] to document classification, named entity recognition, part-of-speech tagging and dependency parsing. The main findings of their paper are:

1. The zero-shot transfer learning ability of mBERT is universal across all tasks, and outperforms strong zero-shot transfer learning baselines based on LSTM.

2. The knowledge mBERT acquired via pretraining varies across different layers. More importantly, freezing early layers of mBERT leads to better performance to the chosen dowstream tasks.

3. mBERT retains monolingual knowledge despite having strong cross-lingual generalizability.

4. The cross-lingual transfer ability strives under the condition when the source language and the target language shares sub-words. Further studies[6] also confirmed that other multilingual models also benefit from sharing sub-words or even scripts(the same alphabets).

We highlight the zero-shot transfer ability is at its best on NLI and NLU tasks, in which the model is asked to classify sentences, tagging part-of-speech, and extract meaningful sentences from teh original document, as opposed to natural language generation tasks where models are asked to generate words and sentences that fulfill a given requirement. Admittedly, recent studies have been working on enhancing the performance of zero-shot transfer on generation tasks[13][23], it is much harder to let a pretrained multilingual model achieve comparable performance with fully supervised training on zero-shot cross-lingual transfer on generation tasks.

2.2 XLM and XLM-R

language models trained monolingual corpora of different corpora through a self-supervised perspective have shown extraordinary results in providing a better initialization to natural language inference tasks. People began to investigate the potential of pretrained multilingual language models on the canonical area of neural machine translation. At year 2019, researchers at Facebook proposed their novel self-supervised pre-train objective that utilizes parallel corpora in different languages[9].

2.2.1 XLM

XLM[9] proposes a new language model pre-train objective using parallel corpora named **Translation Language Modeling**(TLM). TLM is an extension of MLM, where instead of using monolingual text streams, TLM concatenates parallel sentences and randomly masks in both the source and target sentences. To predict a masked out word in English, the model can attended to the context in English, or the word that have the same semantic meaning in the parallel language, or even the context in the parallel language, encouraging the model to align the representations between the source language and the target language.

Experiment results show that the TLM objective yields major improvements(3.6%) over standard MLM with parallel corpora added, advancing the state of the art of the XNLI[2] benchmark to 75.1%. Moreover, obtaining a better alignment not only improves on cross-lingual transfer in classification tasks, but also

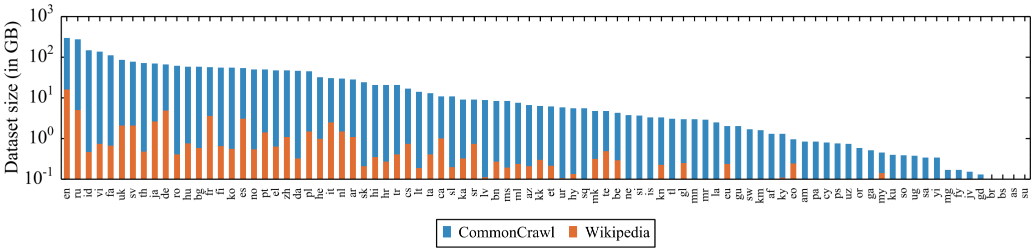


Figure 3: languages and sizes of the language data in the pre-train datasets of XLM-R, pasted from [3]

improves the quality of supervised and unsupervised machine translation.

XLM is trained on data from Wikipedia in the 15 XNLI languages using a MLM objective, and TLM on additional parallel corpora. We want to highlight that high quality parallel corpora is hard to acquire and even more so for low resource languages.

2.2.2 XLM-R

One of the most recent models release by Facebook(or Meta) is XLM-R[3], which is a multilingual variant of RoBERTa[11]. RoBERTa made a few modifications to BERT, the four main ones are eliminating the need for the NSP task; using dynamic masks that varies across epochs; using longer batches and longer training time; using Byte-Pair Encoding[21] for text tokenizing. The backbone structure of XLM-R is a RoBERTa model. Counter-intuitively, XLM-R eliminates the need for parallel data and only uses unlabeled monolingual corpora extracted from the CommonCrawl dataset containing data in 100 languages.

There are a few key findings of XLM-R, and some of them leads to engineering choices that are crucial to its extraordinary performance. Until now, XLM-R still retains the best performance on the XTREME benchmark(a benchmark evaluating the cross-lingual transferability of language models) compared to other models that has comparable parameters(550 million) to XLM-R.

The first finding of XLM-R is no mystery to language models: **scale matters**. The author finds out that using a larger scale of training corpora leads to a better downstream performance. RoBERTa[11] proposed to train language models for more iterations, which lead to better performance. The XLM-R authors points out that using validation loss(perplexity) as a stopping criterion results in an under-tuned language model. The model's performance on downstream tasks continue to improve even after the validation loss have stopped decreasing. Therefore, researchers that trained XLM-R used a relatively larger model size(L = 24, H = 1024, A = 16, 550M params) and an astonishing amount of 2.5 TB training corpora.

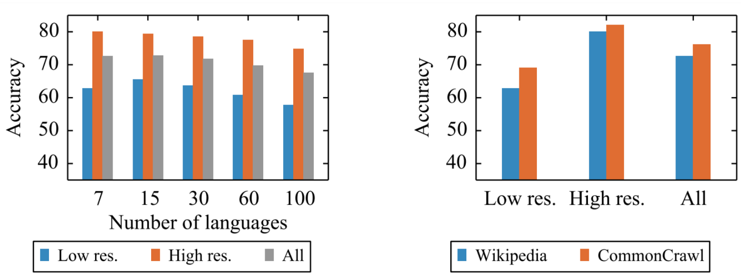


Figure 4: Trade-off between high and low resource language performance. In the right chart, the blue one is trained on wikipedia data containing 100 languages, and the orange one is trained on CC data with only 7 languages. figure pasted from [3]

The second finding of XLM-R is what the authors refer to **the curse of multilinguality**[3], which describes the zero-sum game between the number of languages and the performance of the model on low-resourced languages. It is intuitive to assume that given a fixed sized model, more languages means less capacity dedicated to each language. Since the amount of data for high resource languages far exceeds the amount of data for low resource languages, the performance of low resource languages is severely harmed. Latest research have shown that the curse of multilinguality not only harms the performance of low resource languages, but also negatively affects the performance of high resource languages. From figure 4 we can see that the more the number of pre-training languages, the lower the performance of low resource languages.

We want to highlight that from the left chart in figure 4, the performance of low resource languages increased as we increase the number of pretrained languages increased from 7 to 15. Subsequent studies have found that positive transfer also occurs in that low resource languages are able to transfer knowledge from high resource languages, especially when the two languages share the same script[26][18][6].

The third finding is somewhat a combination of the first two: by scaling the number of parameters and longer training time, we can retain comparable performance of a a multilingual language model to monolingual BERT on the GLUE[24] benchmark, overcoming the curse of multilinguality. We want to highlight that the author uses XLM-7(Model only trained on only 7 languages from CommonCrawl) instead of XLM-R and the baseline is BERT instead of the state-of-the-art RoBERTa, indicating that we can only alleviate the curse of multilinguality but not completely overcome it. Latest studies have also proposed to use language specific modules to over come the curse of multilinguality[15]. Nevertheless, using limited capacity to address unlimited multilingual language semantics will always be a uphill battle.

3 State of the Art

3.1 T5

MT5[27] is the state-of-the-art large multilingual LLM trained and produced by Google Research. It is the multilingual variant of T5[20], which conducted exhaustive experiments in search of the best pretrain objective and key hyper-parameters(corrupt rate, span length...). Here I will list the key engineering choices and experiment results in the T5 paper[20].

3.1.1 Engineering Solutions of T5

- Layer Normalization applied to the **input** of each sub-component, before the residual connection
- No Layer Normalization bias parameter
- Position embedding altered from sinusoidal function[22] to scalar
- Input corpora text cleaning
- Casting each downstream task into a text-to-text format
- 12 block Encoder + 12 block Decoder structure yields good results
- Encoder + Decoder structure Pretrain steps $2^{19} = 524,888$, max sequence length = 512, batch size = 128, a total of 34B tokens pretrained[2]
- **Sequence Packing**: Pack multiple sequences into entry of the batch, each batch contains $2^{16} = 65536$ tokens.
- **Inverse Square Root Learning Rate** $lr = \frac{1}{\sqrt{\max(n,k)}}$, although triangular learning rate works better, it requires us to know the total number of steps.

[2] RoBERTa[11] 2.2T tokens, BERT[4] 137B tokens

- **Multi-Task Sampling** There are two methods of sampling, one is use example proportional sampling, one is use temperature scaled sampling. In the former method, the sampling probability for task i is

$$r_i = \frac{\min(e_i, K)}{\sum_{n=1}^{N} \min(e_n, K)}$$

Where e_i denotes the number of samples of task i, and K is a hyper-parameter. In the latter method:

$$r_i' = (r_i)^{\frac{1}{T}}$$

3.1.2 Key Experiment Results of T5

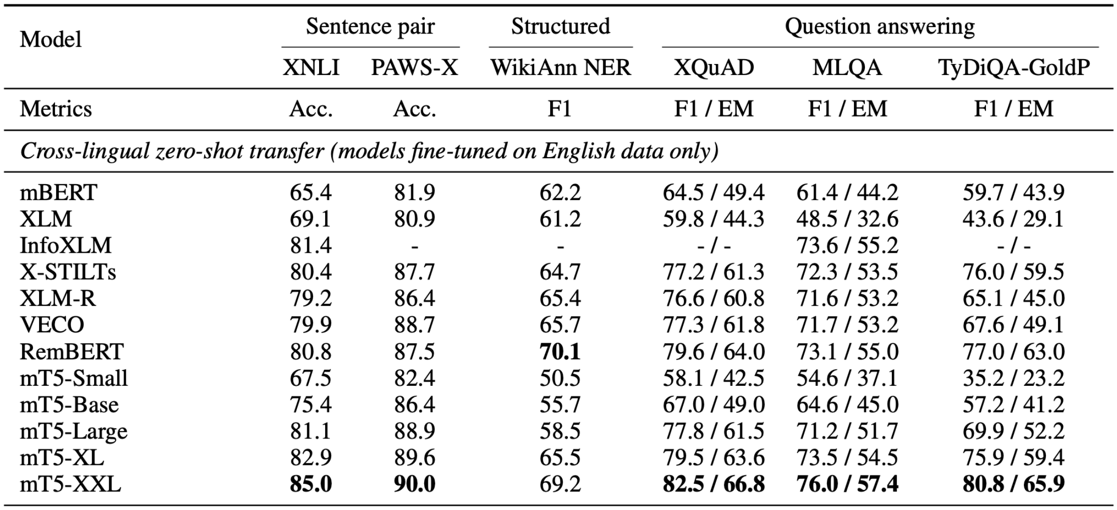
- **Denoising pretrain objective**: Denoising + Encoder-Decoder structure performs the best, as opposed to language modeling objective(left to right generation).
- BERT-Style[4] denoising objective performs better than Prefix language modeling and deshuffling.
- Replacing corrupted **spans** performs better than vanilla BERT[4], MASS(BERT only masking, without randomly replacement), and dropping corrupted tokens.
- Corruption rate 15% performs similarly to 25%. Higher or lower corruption rate performs worse, but not significantly.
- Corrupting span with an average length of 3 performs the best.
- Full dataset with no repeat performs better than a fraction of dataset repeated.
- Finetuning all parameters yield better results than using Adapters[8] and Gradual Unfreezing. Although the results of other parameter efficient finetuning methods need to be examined.
- Low resource tasks(SQuAD) works well with smaller value of adapter dimension d, and high resource tasks works well with larger d.
- In general, multi-task training underperforms pre-training followed by fine-tuning on most tasks. The "equal" mixing strategy in particular results in dramatically degraded performance.
- Finetuning after multi-task pretraining results in comparable performance to standard pretrain and finetuning.
- Performance of omitting the finetune task during multitask pretraining was only slightly worse, suggesting that a model that was trained on a variety of tasks can still adapt to new tasks (i.e. multi-task pre- training might not result in a dramatic task interference).
- Scaling is very crucial to the models performance, while scale up dimensions is silightly better than scaline up training data(or training steps)

3.2 MT5

As the multilingual variant of T5, MT5 is based on the same Transformer encoder-decoder structure but in different sizes. In the following chart is a comparison between different sizes of T5 and MT5. The reason why MT5 is larger is that it uses a much larger vocabulary, thus needing more word/token embedding parameters.

MT5 trains a large multilingual language model across different sizes on a Multilingual Cleaned Common-Crawl Corpora, which they named it by its initials(MC4). MT5 have achieved the state of the art on the XTREME benchmark, which reinforces the findings in XLM-R that larger multilingual model exhibits better transfer learning ability.

Moreover, MT5 closely examines a specific type of bad case that often occurs during zero-shot generation tasks, which the author refer to as **accidental translation**. Taking the example from the paper, if the



Model	Sentence pair		Structured	Question answering		
	XNLI	PAWS-X	WikiAnn NER	XQuAD	MLQA	TyDiQA-GoldP
Metrics	Acc.	Acc.	F1	F1 / EM	F1 / EM	F1 / EM
Cross-lingual zero-shot transfer (models fine-tuned on English data only)						
mBERT	65.4	81.9	62.2	64.5 / 49.4	61.4 / 44.2	59.7 / 43.9
XLM	69.1	80.9	61.2	59.8 / 44.3	48.5 / 32.6	43.6 / 29.1
InfoXLM	81.4	-	-	- / -	73.6 / 55.2	- / -
X-STILTs	80.4	87.7	64.7	77.2 / 61.3	72.3 / 53.5	76.0 / 59.5
XLM-R	79.2	86.4	65.4	76.6 / 60.8	71.6 / 53.2	65.1 / 45.0
VECO	79.9	88.7	65.7	77.3 / 61.8	71.7 / 53.2	67.6 / 49.1
RemBERT	80.8	87.5	**70.1**	79.6 / 64.0	73.1 / 55.0	77.0 / 63.0
mT5-Small	67.5	82.4	50.5	58.1 / 42.5	54.6 / 37.1	35.2 / 23.2
mT5-Base	75.4	86.4	55.7	67.0 / 49.0	64.6 / 45.0	57.2 / 41.2
mT5-Large	81.1	88.9	58.5	77.8 / 61.5	71.2 / 51.7	69.9 / 52.2
mT5-XL	82.9	89.6	65.5	79.5 / 63.6	73.5 / 54.5	75.9 / 59.4
mT5-XXL	**85.0**	**90.0**	69.2	**82.5 / 66.8**	**76.0 / 57.4**	**80.8 / 65.9**

Figure 5: MT5 experiment results, table copied from the MT5 paper[27].

groundtruth span is “新英格兰爱国者队” , the model occasionally predicts the span to be “New 英格兰爱国者队” , accidentally translating the Chinese character “新” to the English word “New” . MT5 believes that the reason why the model outputs English when given a non-English test input is that it has never observed a non-English target during fine-tuning. MT5 proposes two way to mitigate accidental translations.

1. Mixing in the multilingual pretrain objective to the downstream task so that the model will remember how to generate texts in different languages.
2. Further down-weighting high resource corpora during the mixin of the pretrain objective by reducing the langauge sampling coefficient from $\alpha = 0.3$ to $\alpha = 0.1$.

Experiment results show that such a way to mitigate accidental translation is effective. This way of mixing in the pretrain objective into finetuning can be viewed as a way of **Multi-Task Learning**(MTL). In the later sections, I will introduce other MTL strategies that are useful under low resource settings.

	T5	MT5
Small	60M	300M
Base	220M	580M
Large	770M	1.2B
XL	3B	3.7B
XXL	11B	13B

Table 1: Comparison of number of parameters of T5[20] and MT5[27]

3.3 MGLM

In this section, I will briefly introduce our work: Multilingual GLM. I am working on a paper to be submitted to a conference so if you really want to delve deeper into MGLM, I appreciate your patience to wait for our official release of the paper and code. MGLM is the multilingual variant of GLM[5], which is an autoregressive language model trained by with a blank infilling objective. GLM is significant in its autoregressive denoising pretrain method. Autoencoding pretrain models(BERT) relies on the false assumption that masked tokens are uncorrelated, moreover, it suffers from a discrepancy between pretraining and finetuning because it does not observe the [MASK] tokens during the latter period. However, auto regressive models are often unidirectional, incapable of learning deep bidirectional dependencies. We refer our readers

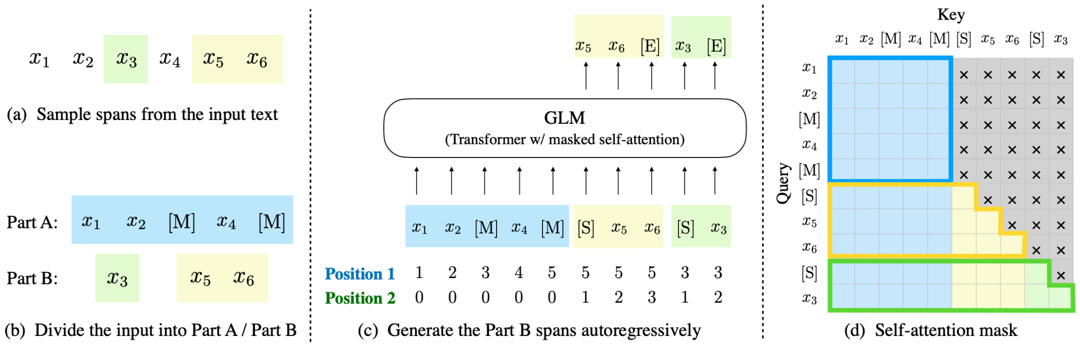

x_1 x_2 x_3 x_4 x_5 x_6
(a) Sample spans from the input text
Part A: x_1 x_2 [M] x_4 [M]
Part B: x_3 x_5 x_6
(b) Divide the input into Part A / Part B
x_5 x_6 [E] x_3 [E]
GLM
(Transformer w/ masked self-attention)
x_1 x_2 [M] x_4 [M] [S] x_5 x_6 [S] x_3
Position 1 1 2 3 4 5 5 5 5 3 3
Position 2 0 0 0 0 0 1 2 3 1 2
(c) Generate the Part B spans autoregressively
Key
x_1 x_2 [M] x_4 [M] [S] x_5 x_6 [S] x_3
Query
(d) Self-attention mask

Figure 6: Illustration of GLM pretrain objective, copied from[5]

to Table 2 for a comparision between autoregressive and autoencoding language models.

GLM combines the advantages of autoregressive and autoencoding language models. By masking a few text spans and autoregressively filling them. Furthermore, GLM incooperates the crux of XLNet[28]: autogregressively fit into all permutations of the source text, which solves the inability to capture bidirectional context. GLM randomly permutes the text spans, rather than the texts inside each span to fully capture interdependencies between different spans.

Mulitilingual GLM follows the model archetecture of GLM[5]. Given an input $X = [x_1, x_2, ..., x_n]$, where n is the length of the input, we sample multiple text spans that contains consecutive tokens $\{s_1, s_2, ..., s_m\}$, where m is the number of spans. Each span $s_i = [x_j, x_{j+1}, x_{j+k}]$ contains k consecutive words(or tokens depending on your source language). During pretraining, each span is substituted with a single [MASK] token. The model predicts the words/tokens inside each span by autoregressively filling the [MASK] tokens.

4 Going Further

4.1 Intermediate Fine-Tuning

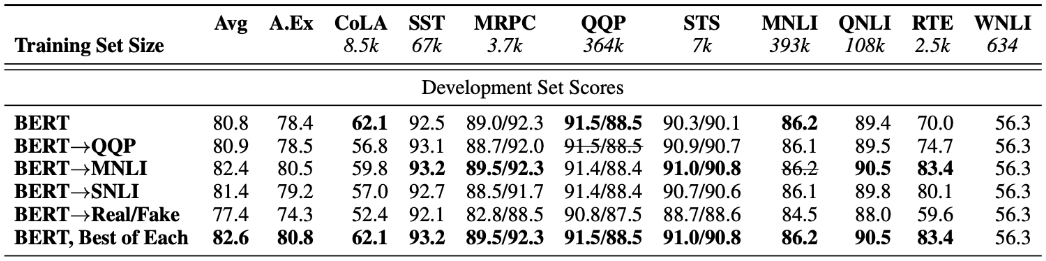
4.1.1 STILT

STILT stands for Supplementary Training on Intermediate Labeled data Tasks. It aims to add an intermediate phase between the standard pre-train and fine-tune paradigm. Experiments are usually conducted in the following form: First a language model is pre-trained with an unsupervised objective like masked language modeling[4]; Then the model is further trained(or intermediate trained) on resourceful labelled data; Finally, the model is fine-tuned further on the target task and evaluated.

The author evaluated this new intermediate-training paradigm on BERT[4], GPT[19] and a variant of

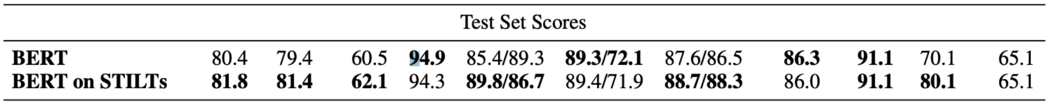
Type	Autoregressive	Autoencoding
Example	GPT-2[19]	BERT[4]
Independence Assumption	No assumption	Masked tokens are independent
Input Noise	No input noise	Task discrepancy between pretrain and finetune
Context Dependency	Unidirectional	Bidirectional

Table 2: Comparison between autoregressive and autoencoding language models



Training Set Size	Avg	A.Ex	CoLA *8.5k*	SST *67k*	MRPC *3.7k*	QQP *364k*	STS *7k*	MNLI *393k*	QNLI *108k*	RTE *2.5k*	WNLI *634*
Development Set Scores											
BERT	80.8	78.4	**62.1**	92.5	89.0/92.3	**91.5/88.5**	90.3/90.1	**86.2**	89.4	70.0	56.3
BERT→QQP	80.9	78.5	56.8	93.1	88.7/92.0	~~91.5/88.5~~	90.9/90.7	86.1	89.5	74.7	56.3
BERT→MNLI	82.4	80.5	59.8	**93.2**	**89.5/92.3**	91.4/88.4	**91.0/90.8**	~~86.2~~	**90.5**	**83.4**	56.3
BERT→SNLI	81.4	79.2	57.0	92.7	88.5/91.7	91.4/88.4	90.7/90.6	86.1	89.8	80.1	56.3
BERT→Real/Fake	77.4	74.3	52.4	92.1	82.8/88.5	90.8/87.5	88.7/88.6	84.5	88.0	59.6	56.3
BERT, Best of Each	**82.6**	**80.8**	**62.1**	**93.2**	**89.5/92.3**	**91.5/88.5**	**91.0/90.8**	**86.2**	**90.5**	**83.4**	56.3

Figure 7: Results of STILT on validation set, figure pasted from [16]



Test Set Scores											
BERT	80.4	79.4	60.5	**94.9**	85.4/89.3	**89.3/72.1**	87.6/86.5	**86.3**	**91.1**	70.1	65.1
BERT on STILTs	**81.8**	**81.4**	**62.1**	94.3	**89.8/86.7**	89.4/71.9	**88.7/88.3**	86.0	**91.1**	**80.1**	65.1

Figure 8: Results of STILT, figure pasted from [16]

ELMo[14] and evaluated on the GLUE benchmark. Since GPT and ELMo are a little bit outdated, I will only show the results of BERT with STILT in figure 7 and 8, representing the validation set adn test set, respectively. We can see that largest gains are usually on the MNLI dataset. However, even after manually selecting the best intermediate task according to the dev set still only results in better performance in 4 tasks out of 9 total downstream tasks. Either indicating that this method either is not effective on smaller are more naive models, or only works on specific datasets. However, this method works better when target data is limited, rendering intermediate training more effective as a data augmentation technique rather than a general improvement of parameter initialization.

The author also conducted experiments of different multitask learning techniques and find out that sequential learning yields the best result when the model backbone is GPT. The authors conclude that ”naive multitask learning appears to yield worse performance than STILTs, at potentially greater computational cost.”

4.1.2 Multilingual STILT

The authors also conducted STILT under a multilingual transfer learning setting[17], where the model is intermediate-trained and fine-tuned on English data only, but evaluated on different languages. The authors state that English STILT yields ”Moderate improvements on question-answering target tasks. MNLI, SQuAD and HellaSwag achieve the best overall results as intermediate tasks, while multi-task intermediate offers small additional improvements.”

Specifically speaking, SQuADv2(with MLM) yields largest gain(+1.9/+2.1) on PAWS-X. Multitask without MLM yields largest gain on NER. SQuADv1.1 yields largest gain on XQuAD and MLQA, note that here SQuADv1.1 serves as both the intermediate task and training set of XQuAD and MLQA. The authors hypothesize that the reason why SQuADv1.1 makes such a good intermediate task is due to the baseline XQuAD and MLQA is undertrained.

Two things that did not work out are 1) Mixing the Pre-train multilingual MLM objective, and 2) Translating the intermediate task dataset to other languages.

4.2 Multi-Task Learning

Another line of research is Multi-Task Learning, which jointly trains the desired task and one or a few auxiliary tasks to boost the performance. Usually, MTL is explored for a specific task(e.g. summarization,

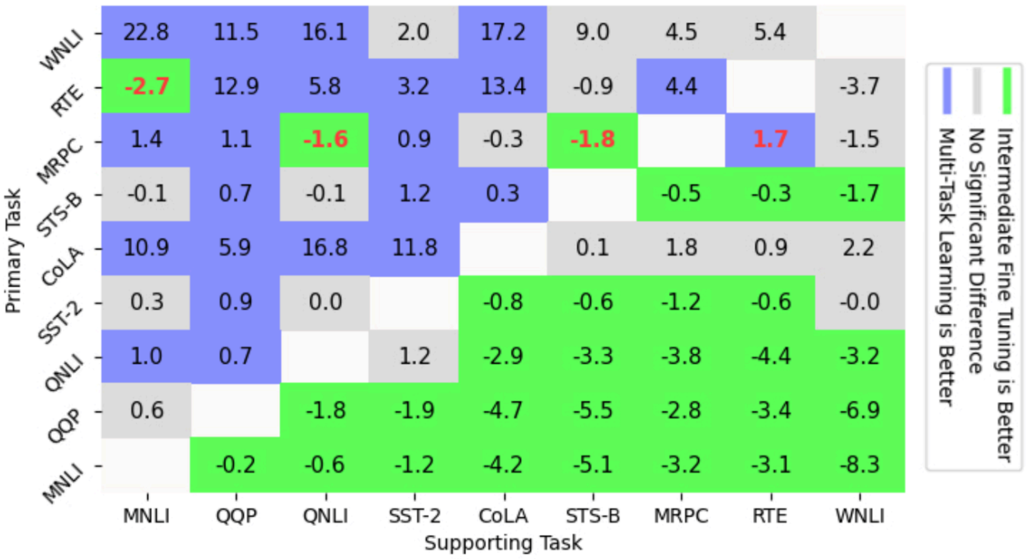
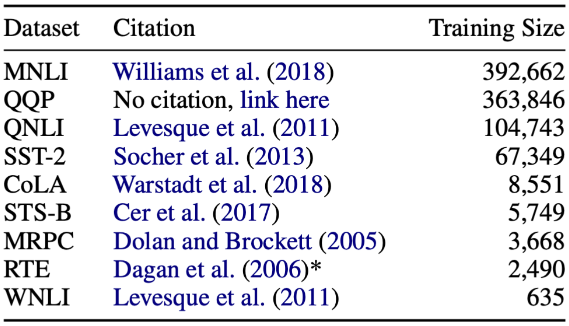


Figure 9: MTL VS Intermediate FT, figure pasted from [25]



Dataset	Citation	Training Size
MNLI	Williams et al. (2018)	392,662
QQP	No citation, link here	363,846
QNLI	Levesque et al. (2011)	104,743
SST-2	Socher et al. (2013)	67,349
CoLA	Warstadt et al. (2018)	8,551
STS-B	Cer et al. (2017)	5,749
MRPC	Dolan and Brockett (2005)	3,668
RTE	Dagan et al. (2006)*	2,490
WNLI	Levesque et al. (2011)	635

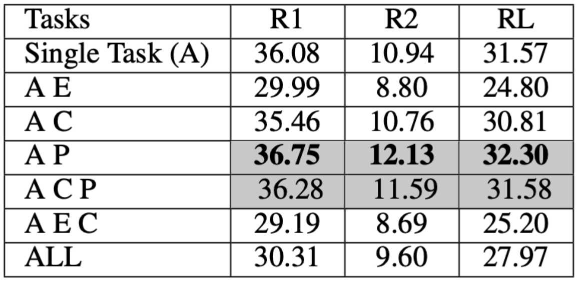
Figure 10: Detailed sizes of GLUE benchmark datasets, figure pasted from [25]

classification) and few studies have investigated which auxiliary task is useful for which type of tasks. Here I will introduce two papers, one is [25], which studies the pros and cons of MTL opposed to intermediate fine-tuning from a downstream task performance perspective; The other one is [12], which performs MTL for a specific task: abstractive summarization.

4.2.1 Multi-Task Learning vs Intermediate Fine-Tuning

This paper compares three different methods: Intermediate Fine-Tuning, pairwise MTL and full MTL. The backbone model is a pretrained DistilRoBERTa.

Experiment results(figure 9) clearly shows that MTL holds its superiority when the data of the downstream task is insufficient because the tasks are sorted from the smallest(WNLI) to the largest(MNLI). Another findings of the authors is that using full MTL, which jointly trains the target dataset along with every other dataset is usually worse than the average performance of using only intermediate FT and the average performance of using only MTL. Therefore even if we did not carefully select between Intermediate FT and MTL and carefully select which auxiliary task to use, full MTL is still worse than the average performance.



Tasks	R1	R2	RL
Single Task (A)	36.08	10.94	31.57
A E	29.99	8.80	24.80
A C	35.46	10.76	30.81
A P	**36.75**	**12.13**	**32.30**
A C P	36.28	11.59	31.58
A E C	29.19	8.69	25.20
ALL	30.31	9.60	27.97

Figure 11: T5 with MTL: C = Concept Labels, P = Paraphrase Detection, E = Extractive Summary, figure pasted from [12]

4.2.2 MTL for Abstractive Summarization

There are two different flavors of summarization: extractive and abstractive. The former aims to extract key spans and sentences from a document to reconstruct a summary. The latter aims to generate a summary from scratch while keeping the semantic meaning of the original document. Therefore abstractive summarization can be decomposed into extractive summarization and paraphrasing. This paper[12] leverages this and and jointly trains an abstractive summarization objective along with extractive summarization, paraphrase detection and also language modeling.

To verify their proposed simple-yet-effective MTL framework, the authors experimented on T5[20] and BERT[4]. Here I will only paste the results of T5. Clearly, Paraphrase detection as the auxiliary task yields the largest gains on T5(also on BERT) when jointly trained for a abstractive summarization objective. Moreover, using auxiliary tasks more or less improve upon the naive single objective training when it comes to the realm of summarization, probably because that, as I have stated before, summarization is a composition of extract and paraphrase. However, when training data is scares, MTL does not yield performance gains.

5 Epilogue

As I have mentioned before, overcoming the curse of multilinguality will always be a uphill battle. In my opinion, the most important task for researchers is not scaling up the model size to achieve better performance. How to attain comparable performance with much less parameters is also crucial. Another line of work is how to make transfer learning work when it comes to abstractive generation. Admitted that multilingual LLMs can be good in extractive generation[1]. Nevertheless our experiments show that it can be hard for a model to generalize under an abstractive summarization setting.

References

[1] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. “On the Cross-lingual Transferability of Monolingual Representations”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 4623–4637. DOI: 10.18653/v1/2020.acl-main.421. URL: https://aclanthology.org/2020.acl-main.421.

[2] Alexis Conneau et al. “XNLI: Evaluating Cross-lingual Sentence Representations”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 2475–2485. DOI: 10.18653/v1/D18-1269. URL: https://aclanthology.org/D18-1269.

[3] Alexis Conneau et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747.

[4] Jacob Devlin et al. "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[5] Zhengxiao Du et al. "GLM: General Language Model Pretraining with Autoregressive Blank Infilling". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 320–335. DOI: 10.18653/v1/2022.acl-long.26. URL: https://aclanthology.org/2022.acl-long.26.

[6] Yoshinari Fujinuma, Jordan Boyd-Graber, and Katharina Kann. "Match the Script, Adapt if Multilingual: Analyzing the Effect of Multilingual Pretraining on Cross-lingual Transferability". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1500–1512. DOI: 10.18653/v1/2022.acl-long.106. URL: https://aclanthology.org/2022.acl-long.106.

[7] Pengcheng He et al. "{DEBERTA}: {DECODING}-{ENHANCED} {BERT} {WITH} {DISENTANGLED} {ATTENTION}". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=XPZIaotutsD.

[8] Neil Houlsby et al. "Parameter-Efficient Transfer Learning for NLP". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 2790–2799. URL: https://proceedings.mlr.press/v97/houlsby19a.html.

[9] Guillaume Lample and Alexis Conneau. "Cross-lingual Language Model Pretraining". In: *CoRR* abs/1901.07291 (2019). arXiv: 1901.07291. URL: http://arxiv.org/abs/1901.07291.

[10] Zhenzhong Lan et al. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=H1eA7AEtvS.

[11] Yinhan Liu et al. *Ro{BERT}a: A Robustly Optimized {BERT} Pretraining Approach*. 2020. URL: https://openreview.net/forum?id=SyxS0T4tvS.

[12] Ahmed Magooda, Diane Litman, and Mohamed Elaraby. "Exploring Multitask Learning for Low-Resource Abstractive Summarization". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1652–1661. DOI: 10.18653/v1/2021.findings-emnlp.142. URL: https://aclanthology.org/2021.findings-emnlp.142.

[13] Kaushal Maurya and Maunendra Desarkar. "Meta-X$_{NLG}$: A Meta-Learning Approach Based on Language Clustering for Zero-Shot Cross-Lingual Transfer and Generation". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 269–284. DOI: 10.18653/v1/2022.findings-acl.24. URL: https://aclanthology.org/2022.findings-acl.24.

[14] Matthew E. Peters et al. "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://aclanthology.org/N18-1202.

[15] Jonas Pfeiffer et al. "Lifting the Curse of Multilinguality by Pre-training Modular Transformers". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 3479–3495. URL: https://aclanthology.org/2022.naacl-main.255.

[16] Jason Phang, Thibault Févry, and Samuel R. Bowman. "Sentence Encoders on STILTs: Supplementary Training on Intermediate Labeled-data Tasks". In: *CoRR* abs/1811.01088 (2018). arXiv: `1811.01088`. URL: `http://arxiv.org/abs/1811.01088`.

[17] Jason Phang et al. "English Intermediate-Task Training Improves Zero-Shot Cross-Lingual Transfer Too". In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing.* Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 557–575. URL: `https://aclanthology.org/2020.aacl-main.56`.

[18] Telmo Pires, Eva Schlinger, and Dan Garrette. "How Multilingual is Multilingual BERT?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4996–5001. DOI: `10.18653/v1/P19-1493`. URL: `https://aclanthology.org/P19-1493`.

[19] Alec Radford et al. "Language Models are Unsupervised Multitask Learners". In: (2019).

[20] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL: `http://jmlr.org/papers/v21/20-074.html`.

[21] Rico Sennrich, Barry Haddow, and Alexandra Birch. "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. DOI: `10.18653/v1/P16-1162`. URL: `https://aclanthology.org/P16-1162`.

[22] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems.* Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

[23] Tu Vu et al. *Overcoming Catastrophic Forgetting in Zero-Shot Cross-Lingual Generation.* 2022. DOI: `10.48550/ARXIV.2205.12647`. URL: `https://arxiv.org/abs/2205.12647`.

[24] Alex Wang et al. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *International Conference on Learning Representations.* 2019. URL: `https://openreview.net/forum?id=rJ4km2R5t7`.

[25] Orion Weller, Kevin Seppi, and Matt Gardner. "When to Use Multi-Task Learning vs Intermediate Fine-Tuning for Pre-Trained Encoder Transfer Learning". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).* Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 272–282. DOI: `10.18653/v1/2022.acl-short.30`. URL: `https://aclanthology.org/2022.acl-short.30`.

[26] Shijie Wu and Mark Dredze. "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).* Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 833–844. DOI: `10.18653/v1/D19-1077`. URL: `https://aclanthology.org/D19-1077`.

[27] Linting Xue et al. "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.* Online: Association for Computational Linguistics, June 2021, pp. 483–498. DOI: `10.18653/v1/2021.naacl-main.41`. URL: `https://aclanthology.org/2021.naacl-main.41`.

[28] Zhilin Yang et al. "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems.* Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: `https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf`.