COMP 550 Final Project
A Comparative Analysis Between mBERT and XLM-R

Xinyi Zhu, Jiaxuan Chen, Junyu Li

Abstract

In this project, we performed a comparative analysis between two cross-lingual language models, Multilingual BERT(mBERT) and XLM-RoBERTa (XLM-R), on an Amazon reviews dataset consisting of reviews in six different languages. The task was to predict the star ratings from 1 to 5, based on the review body. Our experiment aimed to train and finetune these two models on each language, conducting both supervised and zero-shot cross-lingual transfer for fine-grained classification. We used mean absolute error (MAE) and top-1 accuracy to evaluate the performance. Our experimental results affirmed that XLM-R significantly outperforms mBERT and verified our hypothesis that fully supervised classification has a better performance than the unsupervised one.

1 Introduction

BERT (Bidirectional Encoder Representations from Transformers) [5] was developed at Google to work specifically with natural language data in 2020, which can be used for various multilingual tasks. MBERT is based on BERT but was trained in 104 languages simultaneously. XLM-R is another recent state-of-the-art cross-lingual model developed by the Meta research team [6], dynamically changing the masking pattern, presenting the possibility of training one model for many languages while not sacrificing performance. Both models were pretrained with the masked language modeling (MLM) objective.

Phillips et al. [1] generated Multilingual Amazon Reviews Corpus and reported baseline results after fine-tuning the mBERT model on reviews data, but they did not mention other state-of-art models like XLM-R. Thus, We would like to reproduce their results on a smaller scale of data (due to the time and computational limitations) and also apply the XLM-R model to the data for comparison.

We applied supervised zero-shot transfer–representing the labels in the same semantic space as the documents to be classified, where at test time, a learner observes samples from classes that were not observed during training and needed to predict the class they belong to. [12]

Our hypothesis are that

1. XLM-R outperforms mBERT more or less;

2. fully supervised classification has a better performance than the unsupervised one.

2 Related work

2.1 Multilingual Amazon Reviews Corpus

Phillips et al. [1] generated Multilingual Amazon Reviews Corpus (MARC), by applying careful sampling, filtering, and text processing to the documents. This is the dataset we selected for our experiments. The authors reported baseline results for supervised text classification and zero-shot cross-lingual transfer learning by fine-tuning an mBERT model on reviews data.

2.2 XLM-R vs. mBERT

Previous work [4] has shown that XLM-R outperforms mBERT on various cross-lingual benchmarks, including +14.6% average accuracy on XNLI, +13% average F1 score on MLQA, and +2.4% F1 score on NER. XLM-R performed exceptionally well on low-resource languages, improving 15.7% in XNLI accuracy for Swahili and 11.4% for Urdu over previous XLM models. This paper was the first time to show the possibility of multilingual modeling without sacrificing per-language performance.

2.3 mBERT

Google AI Language Team first introduced the language representation model BERT[5], and they revealed that BERT is conceptually simple and empirically powerful. Other works [6] showed that mBERT representations could be split into language-specific and language-neutral components, and the language-neutral component is sufficiently general in modeling semantics to allow high-accuracy word alignment and sentence retrieval. Pires et al.[7] recently examined the cross-lingual properties of mBERT on zero-shot NER and part-of-speech (POS) tagging, but zero-shot transfer success strongly depends on how typologically similar the languages are.

2.4 Other multilingual tasks

Zein Shaheen et al.[10] trained a classifier using an English training set and test using French and German test sets. They found that Language model finetuning of the multilingual pre-trained model (mDistilBERT, mBERT) leads to 32.0-34.94%, 76.15- 87.54% relative improvement on French and German test sets correspondingly. Akiko Eriguchi et al.[11] demonstrated a multilingual Encoder-Classifier for cross-lingual transfer learning by reusing the encoder from a multilingual NMT system and sewing it with a task-specific classifier component. Their system can perform classification in a new language for which no classification data was seen during training, showing that zero-shot classification is possible and remarkably competitive.

3 Method

3.1 Dataset

Multilingual Amazon Reviews Corpus[1] is a large-scale Amazon review for multilingual text classification containing English, Japanese, German, French, Spanish, and Chinese reviews collected between 2015 and 2019. There are 200,000, 5,000, and 5,000 reviews of JSON format in the training, development, and test sets, respectively. Each record in the dataset contains a review ID, a reviewer ID, a product ID, the language, the star rating, a review title, a review text, and a product category. The maximum number of reviews per reviewer or per product is 20. All reviews are truncated after 2,000 characters, and are at least 20 characters long. The corpus is balanced across the five possible star ratings, so each rating constitutes 20 percent of the reviews in each language. We randomly trimmed the training samples to 25,000 for each language, with 150,000 in total due to limited time and resources, and kept all the development and testing samples. The trimmed datasets (and all the code) can be found here: `https://github.com/xyyzh/NLP_final_project`

3.2 Experiments

We did not manually preprocess the dataset. Since our objective was to compare different models, we considered that running the models on the same data would be a fair approach.

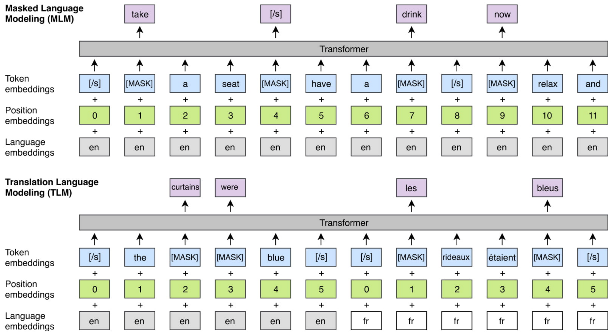

Masked Language Modeling (MLM)
take
[/s]
drink
now
Transformer
Token embeddings
[/s] [MASK] a seat [MASK] have a [MASK] [/s] [MASK] relax and
Position embeddings
0 1 2 3 4 5 6 7 8 9 10 11
Language embeddings
en en en en en en en en en en en en
Translation Language Modeling (TLM)
curtains
were
les
bleus
Transformer
Token embeddings
[/s] the [MASK] [MASK] blue [/s] [/s] [MASK] rideaux étaient [MASK] [/s]
Position embeddings
0 1 2 3 4 5 0 1 2 3 4 5
Language embeddings
en en en en en en fr fr fr fr fr fr

Figure 1: A structure overview of Cross-lingual Language Model Pretraining

3.2.1 XLM-RoBERTa (XLM-R)

For the experimental setup, we loaded *XLM-RobertaTokenizer* [2] and *XLMRobertaModel* [3] from the Hugging Face transformers library. An overview of the general cross-lingual language model structure is in Figure 1. [14] We first loaded the dataset containing JSON objects as dataframes, and passed the input (ie. the review body) to the *XLMRobertaTokenizer*. This tokenizer tokenizes all of the sentences and maps the tokens to their word IDs. It returns pytorch tensors that contain input_ids and attention_masks that indicates to the model which tokens should be attended to, and which should not. Since the length of the majority of the review body is under 100 (Figure 2), we decided to truncate and pad all the sentences to the length of 100. With the generated input_ids and attention_masks, we created the data loaders. The training data loader took training samples in random order by selecting batches randomly each time. We also passed the labels to the loaders. The validation and testing data loader just pulls out batches sequentially since the order does not matter.

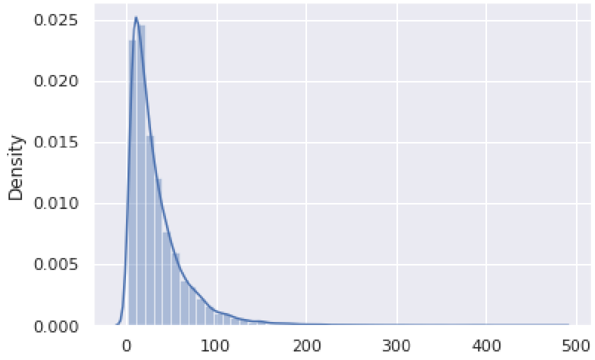


Figure 2: Distribution of the input length (Source language: English)

During training, we unpacked the training batches from our data loaders. In the forward pass, we got the loss and the ”logits”(ie.the model outputs). In the backward pass, the gradients got calculated. The parameters would be modified based on their gradients, learning rate, etc.

We started training the data with *XLMRobertaModel* for 30 epochs but noticed that the model quickly converged at the fourth epoch, after which the validation accuracy was not further increasing. Also, there is more overfitting as the epochs are increasing (Figure 3). Detailed training statistics can be found in the GitHub repository. Therefore, we ran all our following experiments with five epochs. we fine-tuned the model with the AdamW optimizer with a base learning rate of $8 * 10^{-7}$. A scheduler updated the learning rate for each epoch. We used mini-batches of 8, and each epoch took around 10 minutes with a single GPU on google colab.

There are six languages in total, so we trained the model on one source language and tested it on all languages (source or non-source). We reported the mean absolute error (MAE) as well as the accuracy in the end.

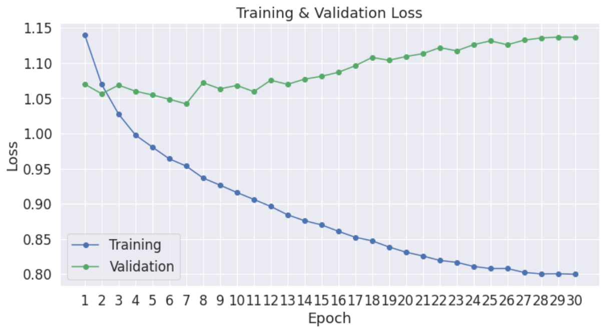


Figure 3: Training and validation loss vs epochs (Source language: English)

3.2.2 mBERT

The experiment procedures for mBERT were similar to XLM-R. First, we tokenized our dataset with the *BertTokenizer* [13] which converted the text to word IDs and attention masks. Then these data are wrapped into tensor objects. The process here followed the decision of the XLM-R model, where all texts were truncated and padded to length 100. With these tensor objects, the data loaders were created to separate the training process into batches to reduce the memory usage of the GPU.

After setting up the data loaders, we loaded the pretrained multilingual model from the *BertModel* of the *transformers* library [13]. We then defined our model structure by adding three sequential linear neural network layers to convert the 768 output channels of the pretrained model to five labels [9], corresponding to the five star ratings of our dataset. A new *forward* method was defined in this structure which takes in the word ids and the mask, giving the logits as output. The optimizer we chose here was the *AdamW* optimizer with a learning rate of $2{*}10^{-5}$, with a scheduler updating the learning rate at each epoch. The number of epochs and batches we chose are 10 and 16 respectively. In google colab with a ‘Tesla P100-PCIE-16GB GPU’, the time of training for each epoch was about 3 minutes. The model converged at about 4-5 epochs. This differed a little, depending on the language we trained the model on. Training the models based on six different languages took up to 4 hours in total. We also reported the mean absolute error (MAE) and accuracy in the end.

3.3 Evaluation metrics

We used two types of evaluation metrics. One is mean absolute error (MAE) as suggested in the previous work [1] to evaluate the performance.

Since the star rating for each review is ordinal, if the true label is one star, the predicted label of five stars should be penalized more heavily than three stars. The formula for MAE is the following:

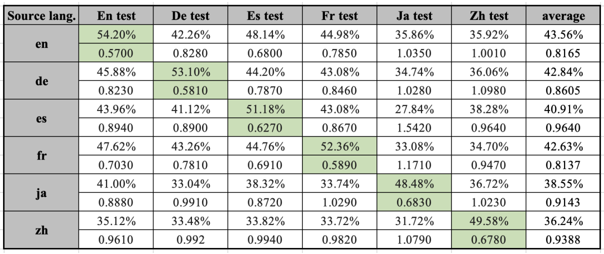
$$MAE(y, \hat{y}) = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

Where $y_i, \hat{y}_i \in \{0, 1, 2, 3, 4\}$ are the true label and the predicted label for the i-th review, respectively. Note that we subtracted the star ratings by 1 because the cross-entropy loss is zero-based, so 0 corresponds to 1 star, 1 corresponds to 2 stars, etc.

Another evaluation metric is the classic top-1 accuracy to observe the exact matches.

4 Results

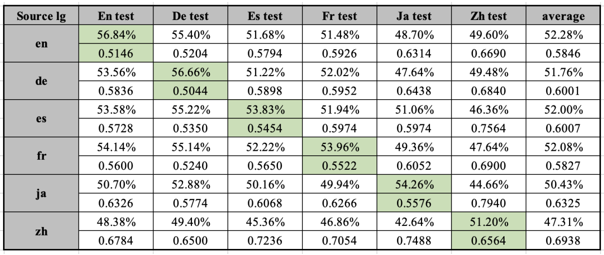
In Figures 4 and 5, we report experimental results for English (en), German (de), Spanish (es), French (fr), Japanese (ja), and Chinese (zh). The ones on the top are the test accuracies and the bottom ones are the MAE's. The diagonals show the results for the fully supervised classification, where the languages of training and testing are the same. The remaining data are for the zero-shot cross-lingual transfer, where we only fine-tune the models on the data from one source language and test on other languages.



Source lang.	En test	De test	Es test	Fr test	Ja test	Zh test	average
en	54.20%	42.26%	48.14%	44.98%	35.86%	35.92%	43.56%
	0.5700	0.8280	0.6800	0.7850	1.0350	1.0010	0.8165
de	45.88%	53.10%	44.20%	43.08%	34.74%	36.06%	42.84%
	0.8230	0.5810	0.7870	0.8460	1.0280	1.0980	0.8605
es	43.96%	41.12%	51.18%	43.08%	27.84%	38.28%	40.91%
	0.8940	0.8900	0.6270	0.8670	1.5420	0.9640	0.9640
fr	47.62%	43.26%	44.76%	52.36%	33.08%	34.70%	42.63%
	0.7030	0.7810	0.6910	0.5890	1.1710	0.9470	0.8137
ja	41.00%	33.04%	38.32%	33.74%	48.48%	36.72%	38.55%
	0.8880	0.9910	0.8720	1.0290	0.6830	1.0230	0.9143
zh	35.12%	33.48%	33.82%	33.72%	31.72%	49.58%	36.24%
	0.9610	0.992	0.9940	0.9820	1.0790	0.6780	0.9388

Figure 4: Experimental results for mBERT

In general, higher accuracy corresponds to a lower mean absolute error. For mBERT, the supervised classification has a better performance than the zero-shot cross-lingual transfer on all languages. The average accuracy for the model trained with English is higher than that trained with other languages.

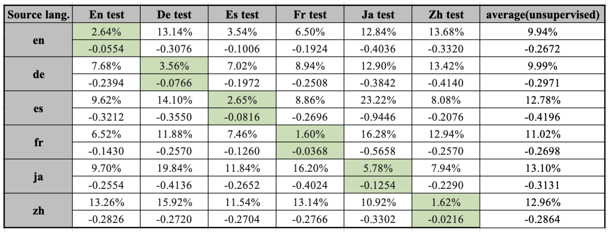


Source lg	En test	De test	Es test	Fr test	Ja test	Zh test	average
en	56.84%	55.40%	51.68%	51.48%	48.70%	49.60%	52.28%
	0.5146	0.5204	0.5794	0.5926	0.6314	0.6690	0.5846
de	53.56%	56.66%	51.22%	52.02%	47.64%	49.48%	51.76%
	0.5836	0.5044	0.5898	0.5952	0.6438	0.6840	0.6001
es	53.58%	55.22%	53.83%	51.94%	51.06%	46.36%	52.00%
	0.5728	0.5350	0.5454	0.5974	0.5974	0.7564	0.6007
fr	54.14%	55.14%	52.22%	53.96%	49.36%	47.64%	52.08%
	0.5600	0.5240	0.5650	0.5522	0.6052	0.6900	0.5827
ja	50.70%	52.88%	50.16%	49.94%	54.26%	44.66%	50.43%
	0.6326	0.5774	0.6068	0.6266	0.5576	0.7940	0.6325
zh	48.38%	49.40%	45.36%	46.86%	42.64%	51.20%	47.31%
	0.6784	0.6500	0.7236	0.7054	0.7488	0.6564	0.6938

Figure 5: Experimental results for XLM-Roberta

For XLM-R, the supervised classification does not necessarily have a better performance. For instances, when we trained in Spanish, the test MAE in German is lower than that on Spanish itself, and when we trained in french, testing in German has the best score. Same as mBERT, the model trained on English achieved the best performance.

In Figure 6, we subtracted all the results from XLM-R by the data in the corresponding cells from mBERT and calculate the average differences for the zero-shot transfer performance.



Source lang.	En test	De test	Es test	Fr test	Ja test	Zh test	average(unsupervised)
en	2.64%	13.14%	3.54%	6.50%	12.84%	13.68%	9.94%
	-0.0554	-0.3076	-0.1006	-0.1924	-0.4036	-0.3320	-0.2672
de	7.68%	3.56%	7.02%	8.94%	12.90%	13.42%	9.99%
	-0.2394	-0.0766	-0.1972	-0.2508	-0.3842	-0.4140	-0.2971
es	9.62%	14.10%	2.65%	8.86%	23.22%	8.08%	12.78%
	-0.3212	-0.3550	-0.0816	-0.2696	-0.9446	-0.2076	-0.4196
fr	6.52%	11.88%	7.46%	1.60%	16.28%	12.94%	11.02%
	-0.1430	-0.2570	-0.1260	-0.0368	-0.5658	-0.2570	-0.2698
ja	9.70%	19.84%	11.84%	16.20%	5.78%	7.94%	13.10%
	-0.2554	-0.4136	-0.2652	-0.4024	-0.1254	-0.2290	-0.3131
zh	13.26%	15.92%	11.54%	13.14%	10.92%	1.62%	12.96%
	-0.2826	-0.2720	-0.2704	-0.2766	-0.3302	-0.0216	-0.2864

Figure 6: Accuracy and MAE differences between XLM-Roberta and mBERT

XLM-R outperforms mBERT in all ways if we compare all the corresponding data. In particular, for all six languages, XLM-R has evident advantages in terms of the unsupervised classification task, with around 10% higher accuracy and 0.3 lower MAE.

5 Discussion

Our experimental results affirm the claim that XLM-R significantly outperforms mBERT, according to the previous research on other datasets such as XNLI, MLQA, and NER. We also verified our hypothesis that fully supervised classification generally has a better performance than the unsupervised one.

All the data were collected from the Amazon marketplace, we noticed that some review contents do not match the ratings, for example, a user wrote a negative review but gave five stars. Also, the differences between two, three, and four-star reviews are not obvious. These can be the factors of the low accuracies.

One interesting finding is that XLM-R has evident advantages in terms of the zero-shot transfer task, compared with mBERT, while for the fully supervised task, there's no significant difference. Also, as reported in the results section, in some cases, testing on other languages can even perform better than on source languages. It is justified since mBERT is just BERT trained on text from many languages, while XLM-R is a more complicated model pretrained on enormous data. XLM-R is wholly self-supervised, so it can generalize across languages better without supervision.

We observed that models trained on English have the best scores, which is not very surprising since these models were pretrained with a large amount of data and English is by far the most well-resourced language.

The drawback of XLM-R is that it took more than three times longer than mBERT to train. It may not be ideal if one has enormous data but only a limited time. For a simpler task, mBERT could be more than sufficient.

The limitations of our work would be that

- we did not include the review titles and the product categories as our input, and the training data samples were at a small scale.
- instead of fine-grained (multi-class) classification, one could conduct binary classification by grouping the labels of 1,2, and 3 and grouping labels of 4 and 5.

For future investigation, we would test XLM-R on languages with very few resources available, since the previous research [6] revealed that the exceptional performance on low-resource languages is one of its important features.

6 Conclusion

This paper presented a comparative analysis of the mBERT and XLM-R models. Our experiment aimed to predict the star ratings from 1 to 5 on an Amazon review dataset consisting of reviews in six different languages. We trained and finetuned these two models on each language. We conducted both supervised and zero-shot cross-lingual transfer for fine-grained classification and observed that fully supervised classification has a better performance than unsupervised one. Furthermore, the result indicated that XML-R consistently outperformed mBERT.

7 Statement of contributions

Xinyi: experiments design and setup, XLM-R model application, data visualization, report writeup.
Jiaxuan: mBERT model application, report writeup.
Junyu: report writeup.

8 References

[1] Keung, Phillip & Lu, Yichao & Szarvas, György & Smith, Noah. (2020). *The Multilingual Amazon Reviews Corpus.* https://arxiv.org/abs/2010.02573

[2] XLMRobertaTokenizer.*Hugging Face* https://huggingface.co/docs/transformers/model_doc/xlmroberta#transformers.XLMRobertaTokenizer

[3] XLMRobertaModel. *Hugging Face* https://huggingface.co/docs/transformers/model_doc/xlmroberta#transformers.XLMRobertaModel

[4] Alexis Conneau & Kartikay Khandelwal & Naman Goyal & Vishrav Chaudhary & Guillaume Wenzek & Francisco Guzman & Edouard Grave & Myle Ott & Luke Zettlemoyer & Veselin Stoyanov. (2019). *Unsupervised Cross-lingual Representation Learning at Scale.* https://arxiv.org/abs/1911.02116

[5] Jacob Devlin & Ming-Wei Chang &Kenton Lee & Kristina Toutanova. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* https://arxiv.org/abs/1810.04805

[6] Jindrich Libovicky & Rudolf Rosa & Alexander Fraser. (2019). *How Language-Neutral is Multilingual BERT?* https://arxiv.org/abs/1911.03310

[7] Telmo Pires & Eva Schlinger & Dan Garrette. (2019). How Multilingual is Multilingual BERT? https://aclanthology.org/P19-1493/

[8] Transfer learning NLP: Fine tune bert for text classification.(2020, July 26). *Analytics Vidhya.* https://www.analyticsvidhya.com/blog/2020/07/transfer-learning-for-nlp-fine-tuning-bert-for-text-classification/

[9] Chris McCormick. (2019, July 22). Bert fine-tuning tutorial with pytorch. https://mccormickml.com/2019/07/22/BERT-fine-tuning/

[10] Zein Shaheen & Gerhard Wohlgenannt & Dmitry Mouromtsev. (2021).Zero-Shot Cross-Lingual Transfer in Legal Domain Using Transformer Models. https://arxiv.org/abs/2111.14192

[11] Akiko Eriguchi & Melvin Johnson & Orhan Firat & Hideto Kazawa & Wolfgang Macherey. (2018). Zero-Shot Cross-lingual Classification Using Multilingual Neural Machine Translation. https://arxiv.org/abs/1809.04686

[12] Wikimedia Foundation. (2021, November 21). Zero-shot learning. Wikipedia. Retrieved December 19, 2021, from Jarble. https://en.wikipedia.org/wiki/Zero-shot_learning

[13] Bert. *Hugging Face*. Retrieved December 19, 2021, from https://huggingface.co/docs/transformers/model_doc/bert

[14] Facebookresearch. “Facebookresearch/XLM: Pytorch Original Implementation of Cross-Lingual Language Model Pretraining.” GitHub, https://

github.com/facebookresearch/XLM.