Improved Multilingual Language Model Pretraining for Social Media Text via Translation Pair Prediction

Shubhanshu Mishra
Twitter, Inc.
smishra@twitter.com

Aria Haghighi
Twitter, Inc.
ahagighi@twitter.com

Abstract

We evaluate a simple approach to improving zero-shot multilingual transfer of mBERT on social media corpus by adding a pretraining task called translation pair prediction (TPP), which predicts whether a pair of cross-lingual texts are a valid translation. Our approach assumes access to translations (exact or approximate) between source-target language pairs, where we fine-tune a model on source language task data and evaluate the model in the target language. In particular, we focus on language pairs where transfer learning is difficult for mBERT: those where source and target languages are different in script, vocabulary, and linguistic typology. We show improvements from TPP pretraining over mBERT alone in zero-shot transfer from English to Hindi, Arabic, and Japanese on two social media tasks: NER (a 37% average relative improvement in F_1 across target languages) and sentiment classification (12% relative improvement in F_1) on social media text, while also benchmarking on a non-social media task of Universal Dependency POS tagging (6.7% relative improvement in accuracy). Our results are promising given the lack of social media bitext corpus. Our code can be found at: https://github.com/twitter-research/multilingual-alignment-tpp.

1 Introduction

Multilingual BERT (mBERT; Devlin et al. 2019), and other multilingual pretrained language models have been shown to learn surprisingly good cross-lingual representations for a range of NLP tasks (Pires et al., 2019; Mishra et al., 2021, 2020; Mishra and Mishra, 2019), despite not having explicit cross-lingual links between their monolingual training corpora (Artetxe and Schwenk, 2018; Yang et al., 2019). Analysis in Pires et al. (2019) shows that cross-lingual transfer with mBERT works best when transferring representations between source
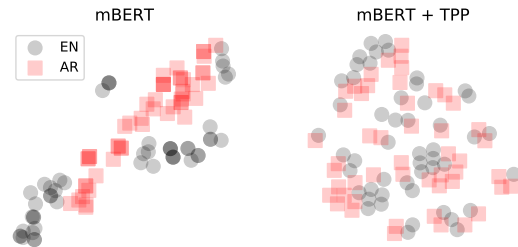

Figure 1: **Comparing Multilingual Representations** Left figure shows mBERT embeddings (T-SNE projection) for parallel sentences from the Tatoeba corpus (Section 3.1) in English (EN) and Arabic (AR). These embeddings exhibit distinct language regions, making cross-lingual transfer challenging. The right figure shows the same data after further pretraining using translation pair prediction (Figure 2), which learns representations that are near cross-lingual translations (change in distance showed in Figure 3 of appendix). In Section 3, we show this improves cross-lingual transfer performance on several tasks (see Table 2).

and target languages that share lexical structure (i.e, overlapping word-piece vocabulary) and structural linguistic typology. In some ways, this result is not surprising since mBERT does not see explicit cross-lingual examples and cannot learn how to transform representations to a target language and relies on similar linguistic structure to transfer well. Furthermore, limited work exists on multilingual transfer for social media corpus which tend to be more noisy and shorter compared to traditional NLP corpora. This issue is further exacerbated for transfer between language pairs which use different scripts.

In this work, we explore whether we can improve multilingual transfer by providing cross-lingual examples and encouraging mBERT to learn aligned representations between source and target texts before task fine-tuning. Concretely, we perform a "language pretraining" task called *translation pair prediction* (TPP) where we assume access to a corpus of source-target text pairs and predict whether the pair is a translation of the same content (see Section 2). This pretraining phase is intended to

occur after standard mBERT pretraining (Devlin et al., 2019), but before task fine-tuning (see Figure 2). The intent behind this task is to leverage translation data that exists between a source language with abundant task training data and a target language with little or no task training data in order to improve transfer learning. As with standard mBERT pretraining, one can pretrain mBERT using TPP once and fine-tune for multiple tasks (as long as transferring to the same target language set). The translation pair data doesn't need to be related to the downstream task and we can leverage multiple sources of source-target translations of varying degrees of translation quality (see Section 3.3).

We demonstrate the benefits of adding TPP pretraining in experiments on social media corpus for NER, and sentiment detection, along with a non-social media corpus of universal POS tagging (for comparison) on fine-tuned models from English to Hindi, Japanese, and Arabic (see Section 3 for evaluation). We show gains from TPP on all tasks (averaged across target languages) using translation data of varying quality (see Section 3.3).

Related works Gururangan et al. (2020) argues for the benefits of continued transformer pretraining for better performance on in-domain tasks. Our work closely follows this approach however instead of keeping the Masked Language Modelling (MLM) task on in-domain data, we introduce a new pretraining task (TPP). In Artetxe and Schwenk (2019) the authors train an BiLSTM encoder-decoder model on a large parallel corpora of multiple languages and demonstrate zero-shot transfer. In Eisenschlos et al. (2019) the authors use a domain/task specific fine-tuning to improve zero shot cross lingual performance by inducing labels from the model of Artetxe and Schwenk (2019). Our work is closely related to the work of Wieting et al. (2019); Guo et al. (2018) which use a subword averaging and a LSTM encoder. Our approach uses transformers and also focuses evaluation on individual token level task like NER as well as sentence level task. Another closely related work is of Cao et al. (2020) which uses word pair alignment with mBERT and evaluates on XNLI task. However, their work does not consider typologically different languages like this work. In Lample and Conneau (2019) the authors use Translation Language Modeling task which concatenate the translation pairs in the BERT model and also add a language identifier to each token. Huang et al.
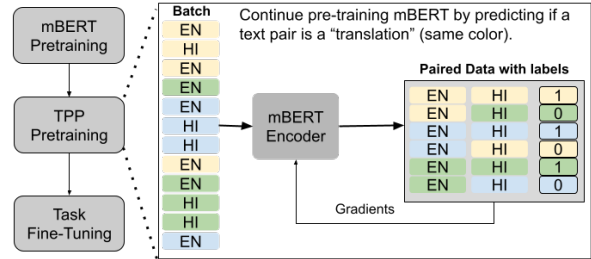

Figure 2: **Translation Pair Prediction (TPP)** We further pre-train an mBERT model on the TPP task (see Section 2). For each batch of aligned source-target text translation pairs, we create a balanced binary dataset consisting of the aligned pairs (positive examples) and a random source-text within the batch (negative examples). We update the mBERT model based on optimizing for binary cross-entropy on this task. In Section 3, we show this added TPP phase improves transfer learning to the target language on downstream tasks.

(2019) include a similar approach as discussed before and get improvement from jointly fine-tuning model with translations of downstream task data. Our approach is comparatively simpler to implement as we don't make any assumption about the encoding model or assumptions about word-level alignments; our approach works independently of the choice of encoder (mBERT here) or the quality of word-alignment models.

2 Model

A description of our approach is provided in Figure 2. The primary exploration in this work is the impact of adding a translation pair prediction (TPP) pretraining phase between traditional mBERT pretraining and downstream task fine-tuning (Devlin et al., 2019). The TPP pretraining task assumes a collection of aligned text pairs (s, t) which are "translation" pairs between source and target language text pairs s and t respectively. It is assumed that both languages have word-piece coverage in the pretrained mBERT model, but that we primarily have task training data for the source language. After TPP pretraining, we can fine-tune the resulting model for multiple tasks as with standard mBERT for transfer to the target language(s).

The TPP task is a binary classification problem where given an (s, t) pair we predict whether it is a valid "translation" pair or a random source-target pair using the dot product of mBERT encodings for s and t.[1] More formally, if $f(\cdot)$ represents the encoder mapping a document to the mBERT

---

[1]Note that in contrast to some related work (Huang et al., 2019; Lample and Conneau, 2019) we explore a symmetric task formulation instead of an asymmetric formulation as in the Next Sentence Prediction (NSP) pretraining task where

[CLS] token embedding, we predict if the pair is a translation pair with probability:

$$\texttt{is-translation}(s,t) = \sigma\left(f(s)^T f(t)\right)$$

where $\sigma(\cdot)$ is the sigmoid activation function. We create a binary dataset from aligned pairs by sampling a random unaligned s and t pair as a negative example for each aligned pair, creating a balanced dataset. The TPP task is trained using binary cross-entropy against this label and the loss gradient back-propogates through the embedding function $f(\cdot)$ which include mBERT layer parameters.

After TPP pretraining, we fine-tune on task data in the source language (English in our experiments) and evaluate transfer performance on target languages (Hindi, Japanese, and Arabic in our experiments).

3 Evaluation

3.1 Experimental setup

In order to evaluate our model we focus on three languages which are dissimilar in vocabulary and syntax from English. Furthermore, **our goal is to improve performance on social media data which is noisier as well as low resource for the above languages and our benchmarked tasks compared to newswire corpus**. Based on these criteria we utilize three languages apart from English which meet the following requirements: a) availability of parallel translation corpora, and b) availability of task data, particularly in the social media domain (NER and sentiment detection here). Based on the above criteria we identify English as our source language and Hindi, Japanese, and Arabic as target languages. We also include POS tagging on Universal Dependencies, which is a non social media benchmark corpus to evaluate our models. More details can be found at: `https://github.com/twitter-research/multilingual-alignment-tpp`.

Lang pair	Tatoeba	Wikimatrix	Wikidata
en-ar	28K	773K	1.6M
en-ja	220K	480K	509K
en-hi	11K	134K	77K

Table 1: Translation pair corpus data sizes used for translation pair prediction pretraining. See Section 3.1 for details.

---

source and target representations are concatenated before prediction.

Translation pair data For translation pairs we utilize datasets from the Tatoeba (TT) [2] collection as well as the Wikimatrix (WM) collection (Schwenk et al., 2019). While Tatoeba consists of human generated short sentences, Wikimatrix contains aligned sentences mined using a neural network. We assume that Tatoeba has higher quality across languages compared to Wikimatrix, where many of the translations are either incorrect or not exact translations. We also introduce a new dataset called Wikidata aligned pairs (WD) which can be automatically generated by using pair of language labels and descriptions from Wikidata which are written by human annotators in different scripts. Table 1 has details on the dataset size.

Wikidata aligned pairs (WD) We collect dataset from Wikidata (`https://wikidata.org`). For a given language pair, we consider wikidata items which have wikidata item labels and descriptions in the given pair of languages. We concatenate the label and the description using a space to make a sentence and use the sentences for each language as a translation pair.

NER data For our downstream task we use Tweet datasets. For NER we generate a dataset of 100k English Tweets, which is used for training. It consists of Tweets annotated for PERSON, LOCATION, PRODUCT, ORGANIZATION, and OTHER. We also generate test datasets of 2.3k Japanese and 10k Arabic Tweets. All datasets are generated using crowd sourced annotations. For Hindi, we use the Hindi subset of 2008 SSEA shared task (Rajeev Sangal and Singh, 2008). We report micro avg-F1 score across all classes. We use the standard NER task formulation for BERT style models which use softmax on the first sub-word for each token to make the prediction (Devlin et al., 2019). The NER models can be further improved by using Conditional Random Fields (CRF) (Lafferty et al., 2001) layer on the output which often gives good performance even without neural features across all languages (Mishra, 2020).

Sentiment Detection data We utilize the SemEval Twitter Sentiment dataset (Rosenthal et al., 2017) with the data split as used in Mishra and Diesner (2018). For Japanese, we utilize the 500k Tweet dataset from Suzuki (2019). For Arabic we use the data from Abdulla et al. (2013). For Hindi,

---

[2]`https://tatoeba.org` data is under CC-BY 2.0 FR.

we use the dataset from SAIL 2015 shared task for Indian languages (Patra et al., 2015). We report macro F_1 across classes as in Rosenthal et al. (2017).

Universal Dependencies POS data We use the English, Hindi, Japanese, and Arabic subsets from the Universal dependencies data (Nivre et al., 2020). We train using the GUM + PUD subset for English and evaluate on PUD subset for other languages. We do note a shortcoming of our PUD based datasets, i.e. they were created by translating the original text from English, German, French, Italian, or Spanish.[3] This can introduce translation artifacts and can favor translation based pretraining approaches.

Training All models are trained using the mBERT model available from the HuggingFace library (Wolf et al., 2019). For each translation pair data the model is trained for 3 epochs (hyperparameter details in appendix section A). For fine-tuning on downstream task we initialize our model with weights from the pretrained models and train it on the English task dataset for 3 epochs.

TPP using all target languages We also experiment with a TPP model which was pretrained using all the target language pair data from Tatoeba which were equally sampled. We denote this setting by ALL in Table 2 for our results. Even though this model may not perform as well as pretraining with TPP for a single target language at a time, it allows us to reduce pretraining and inference time, as well as reduce the number of models to manage.

Change in embedding distance after alignment In Figure 3 we show how the normalized (by max) embedding distances change beetween sentence pairs shown in Figure 1 after applying TPP.We find that alignment pairs are closer than their distance in mBERT.

3.2 Zero-shot improvement via TPP

As can be seen in Table 2, adding TPP pretraining improves performance for both NER and sentiment detection across all languages. The universal POS gains are consistent, but more modest; we hypothesize this is because POS-tagging reilies more heavily on the language-neutral components of mBERT representations (Libovický et al., 2019). The greatest improvement of our models is for Japanese NER

---

[3]Details on UD PUD datasets - `https://github.com/UniversalDependencies/UD_English-PUD`

	Hindi		Japanese		Arabic	
NER	F_1	$\Delta\%$	F_1	$\Delta\%$	F_1	$\Delta\%$
mBERT	21.1	0.0	16.5	0.0	32.1	0.0
+TPP (ONE)	**24.3**	15.2	**29.9**	81.4	**39.4**	22.8
+TPP (ALL)	23.2	10.3	27.4	66.4	38.5	19.9
Sentiment	F_1	$\Delta\%$	F_1	$\Delta\%$	F_1	$\Delta\%$
mBERT	31.7	0.0	55.0	0.0	51.5	0.0
+TPP (ONE)	**32.7**	3.0	66.4	20.6	58.3	13.2
+TPP (ALL)	32.4	2.3	**67.7**	23.1	**58.5**	13.7
UD POS	acc.	$\Delta\%$	acc.	$\Delta\%$	acc.	$\Delta\%$
mBERT	67.4	0.0	52.7	0.0	64.0	0.0
+TPP (ONE)	**71.5**	6.0	**57.6**	9.2	**67.1**	4.8
+TPP (ALL)	66.4	-1.5	52.7	0.1	65.0	1.5

Table 2: Results on zero-shot transfer learning for NER (F_1), sentiment detection (F_1), and UD POS tagging (Accuracy) from English to Hindi, Japanese, and Arabic. We compare fine-tuning standard mBERT to fine-tuning mBERT further pretrained on the TPP task (Section 2). The ONE variant pretrains a separate model for each target language; the ALL variant is a single pretraiend model for all target languages. We use $\Delta\%$ to display relative improvement.
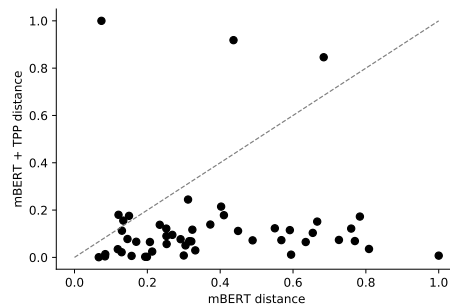

Figure 3: Change in embedding distance of EN-AR translation pairs shown in Figure 1

for which the model improves by 81% and an absolute improvement of around 15% in F_1 score. We hypothesize that NER benefits the most from some our TPP sources (Wikimatrix and Wikidata) which contain many paired entity descriptions. Furthermore, the Tatoeba dataset comprises of short sentence which are not likely to contain sentiment information, especially sentiment bearing content likely to be shared on social media platforms.

3.3 Improvement using combinations of alignment pairs and quality of alignments

In this section we discuss the impact of translation datasets available for our languages as well as how combining them for the TPP task leads to improved performance. We focus on NER F_1 and relative

	Hindi		Japanese		Arabic	
NER	F_1	$\Delta\%$	F_1	$\Delta\%$	F_1	$\Delta\%$
mBERT	21.1	0.0	16.5	0.0	32.1	0.0
+TPP (TT)	23.1	9.6	27.8	68.6	36.3	13.2
+TPP (WD)	22.4	6.3	26.5	60.8	36.9	15.0
+TPP (WM)	21.6	2.6	27.7	68.3	38.3	19.3
+TPP (BP)	24.3	15.2	29.9	81.4	39.4	22.8
+TPP (ALL)	23.2	10.3	27.4	66.4	38.5	19.9
Sentiment	F_1	$\Delta\%$	F_1	$\Delta\%$	F_1	$\Delta\%$
mBERT	31.7	0.0	55.0	0.0	51.5	0.0
+TPP (TT)	31.8	0.3	62.4	13.5	58.3	13.2
+TPP (WD)	30.8	-2.9	50.2	-8.7	53.0	3.0
+TPP (WM)	32.7	3.0	63.2	14.8	54.7	6.4
+TPP (BP)	32.0	0.9	66.4	20.6	55.3	7.5
+TPP (ALL)	32.4	2.3	67.7	23.1	58.5	13.7
UD POS	acc.	$\Delta\%$	acc.	$\Delta\%$	acc.	$\Delta\%$
mBERT	67.4	0.0	52.7	0.0	64.0	0.0
+TPP (TT)	65.1	-3.5	54.0	2.4	66.7	4.1
+TPP (WD)	70.5	4.5	53.0	0.5	66.4	3.7
+TPP (WM)	70.4	4.3	54.4	3.1	65.4	2.2
+TPP (BP)	71.5	6.0	57.6	9.2	67.1	4.8
+TPP (ALL)	66.4	-1.5	52.7	0.1	65.0	1.5

Table 3: Impact of translation quality and using combination of translations in NER, Sentiment, and UD POS examples task performance after TPP (F_1 = micro F_1 score for NER, macro F_1 for sentiment, acc. = accuracy for UD POS). $\Delta\%$ is the % change in evaluation score compared to mBERT. Translation pairs used for each TPP task are shown in parenthesis, TT=Tatoeba, WD=Wikidata, WM=WikiMatrix, BP=best pairing of TT, WD, WM, and ALL=TT pairs from all languages, equally sampled.

improvements (denoted $\Delta\%$), but other tasks also see similar variation for each language (see Table 3 in appendix). We find that increasing the alignment pair data or using higher quality alignment data yield improvements. Using single-corpus translation pairs alone leads to significant improvements for all languages ($\Delta\%$ HI=9.6, JA=68.6, AR=19.3). Tatoeba results in most prominent improvements (HI=9.6, JA=68.6), while Wikimatrix has the best performance for AR (19.3). We assume this is related to larger size of the data, e.g. JA has 10 times more data in Tatoeba as compared to AR and HI.

Next, we move to sequencing pairs of TPP tasks from each translation dataset for a given language pair. We find that as we pair the TPPdatasets the performance of the models continues to improve (Table 3) but it depends on the previous checkpoint we start from as well as the quality of the new TPP data. Our experiments revealed that using the best performing single pairs in their order of performance individually improves performance significantly ($\Delta\%$ HI=15.2, JA=81.4, AR=22.8). For Sentiment and UD POS we observe similar gains, except in few case there is slight degradation.

Note on translation quality In terms of translation quality, Tatoeba is likely to be the most accurate as it is manually curated. Next, Wikidata is likely to be higher quality for HI compared to Wikimatrix as Wikimatrix is auto generated using a model and hence likely to perform worse on low resource languages. For AR and JA we can expect Wikimatrix to be higher quality as these languages have larger Wikipedia size and hence likely to have better quality representation in pretrained LMs.

4 Conclusion

We evaluated the use of translation pair prediction (TPP) as a pretraining task which can improve the performance of multilingual language models for cross-lingual zero-shot transfer from English. This pretraining task is performed after standard mBERT pretraining and the resulting model can be task fine-tuned for cross-lingual transfer similar to mBERT. We show significant improvement in zero-shot transfer by adding TPP pretraining on NER and sentiment detection on social media tasks for three languages (Hindi, Japanese, and Arabic) which are dissimilar in vocabulary and syntax from English.

References

N. A. Abdulla, N. A. Ahmed, M. A. Shehab, and M. Al-Ayyoub. 2013. Arabic sentiment analysis: Lexicon-based and corpus-based. In *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–6.

Mikel Artetxe and Holger Schwenk. 2018. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *CoRR*, abs/1812.10464.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transac-*

tions of the Association for Computational Linguistics, 7:597–610.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *International Conference on Learning Representations*.

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

Julian Eisenschlos, Sebastian Ruder, Piotr Czapla, Marcin Kadras, Sylvain Gugger, and Jeremy Howard. 2019. MultiFiT: Efficient multi-lingual language model fine-tuning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5702–5707, Hong Kong, China. Association for Computational Linguistics.

Mandy Guo, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Hernandez Abrego, Keith Stevens, Noah Constant, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Effective parallel corpus mining using bilingual sentence embeddings. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 165–176, Brussels, Belgium. Association for Computational Linguistics.

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

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pretraining with multiple cross-lingual tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2485–2494, Hong Kong, China. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Jindrich Libovický, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *CoRR*, abs/1911.03310.

Shubhanshu Mishra. 2020. Non-neural Structured Prediction for Event Detection from News in Indian Languages. In *Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation*, Hyderabad, India. CEUR Workshop Proceedings, CEUR-WS.org.

Shubhanshu Mishra and Jana Diesner. 2018. Detecting the correlation between sentiment and user-level as well as text-level meta-data from benchmark corpora. In *Proceedings of the 29th on Hypertext and Social Media*, HT '18, page 2–10, New York, NY, USA. Association for Computing Machinery.

Shubhanshu Mishra and Sudhanshu Mishra. 2019. 3Idiots at HASOC 2019: Fine-tuning Transformer Neural Networks for Hate Speech Identification in Indo-European Languages. In *Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation*, pages 208–213, Kolkata, India.

Sudhanshu Mishra, Shivangi Prasad, and Shubhanshu Mishra. 2020. Multilingual joint fine-tuning of transformer models for identifying trolling, aggression and cyberbullying at TRAC 2020. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 120–125, Marseille, France. European Language Resources Association (ELRA).

Sudhanshu Mishra, Shivangi Prasad, and Shubhanshu Mishra. 2021. Exploring multi-task multi-lingual learning of transformer models for hate speech and offensive speech identification in social media. *SN Computer Science*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. 2015. Shared task on sentiment analysis in indian languages (sail) tweets - an overview. In *Mining Intelligence and Knowledge Exploration*, pages 650–655, Cham. Springer International Publishing.

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

Dipti Misra Sharma Rajeev Sangal and Anil Kumar Singh, editors. 2008. *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*. Asian Federation of Natural Language Processing, Hyderabad, India.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *ArXiv*, abs/1907.05791.

Yu Suzuki. 2019. Filtering method for twitter streaming data using human-in-the-loop machine learning. *Journal of Information Processing*, 27:404–410.

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Simple and effective paraphrastic similarity from parallel translations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4602–4608, Florence, Italy. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

A Experiment

Code availability Our model training code and evaluation code will be released at: https://github.com/twitter-research/multilingual-alignment-tpp.

Training setup Our default parameters are as follows: We use Adam optimizer with 500 warmup steps=500 and weight decay of 0.01. Our batch size is 16. Models were trained using 2 NVIDIA v100 GPUs. Training time for TPP ranged from 30 mins to 32hrs for different settings. NER training time was around 1.5 hrs. Sentiment training time was 1 minute, and UD POS training time was also 1 minute.

Evaluation For NER tasks we used micro-averaged F1 score as implemented in the seqeval [4] library. For sentiment tasks we use document level micro-averaged F1 score implemented in scikit-learn library [5]. For UD-POS tasks we used token level accuracy as implemented in scikit-learn library.

B Data

B.1 Data sources

NER

1. SSEA Hindi - From the Workshop on NER for South and South East Asian Languages http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=5

2. Tweets annotated for in English, Japanese, and Arabic with entities from the following types: PERSON, LOCATION, PRODUCT, ORGANIZATION, and OTHER.

Sentiment For all sentiment datasets we only consider the positive and negative examples and exclude the neutral class if present.

1. Japanese - Available at: http://www.db.info.gifu-u.ac.jp/data/Data_5d832973308d57446583ed9f

2. Arabic - Available at: https://archive.ics.uci.edu/ml/datasets/Twitter+Data+set+for+Arabic+Sentiment+Analysis

3. Hindi - Available at: http://amitavadas.com/SAIL/data.html

---

[4] https://github.com/chakki-works/seqeval
[5] https://scikit-learn.org/