Small and Practical BERT Models for Sequence Labeling

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li and Amelia Archer

Google Research
San Francisco, CA, USA
{henrytsai,riesa,melvinp,navari,xinl,ameliaarcher}@google.com

Abstract

We propose a practical scheme to train a single multilingual sequence labeling model that yields state of the art results and is small and fast enough to run on a single CPU. Starting from a public multilingual BERT checkpoint, our final model is 6x smaller and 27x faster, and has higher accuracy than a state-of-the-art multilingual baseline. We show that our model especially outperforms on low-resource languages, and works on codemixed input text without being explicitly trained on codemixed examples. We showcase the effectiveness of our method by reporting on part-of-speech tagging and morphological prediction on 70 treebanks and 48 languages.

1 Introduction

There have been many recent modeling improvements (Smith et al., 2018; Bohnet et al., 2018) on morphosyntactic tagging tasks. However, these models have largely focused on building separate models for each language or for a small group of related languages. In this paper, we consider the implications of training, evaluating, and deploying a single multilingual model for a diverse set of almost 50 languages, evaluating on both part-of-speech tagging and morphological attribute prediction data from the Universal Dependencies repository (Nivre et al., 2018).

There are several benefits of using one multilingual model over several language specific models.

- Parameter sharing among languages reduces model size and enables cross-lingual transfer learning. We show that this improves accuracy, especially for low-resource languages.
- No language identification model needed to decide which language-specific model to query. Critically, this reduces system complexity and prevents prediction errors from language identification from propagating into the downstream system.
- Multilingual models can be applied to multilingual or codemixed inputs without explicitly being trained on codemixed labeled examples. Otherwise, given, e.g. a mixed Hindi/English input, one must decide to query either the Hindi model or the English model, both of which are sub-optimal.

In this paper, we show that by finetuning a pretrained BERT (Devlin et al., 2019) model we can build a multilingual model that has comparable or better accuracy to state-of-the-art language-specific models, and outperforms the state-of-the-art on low-resource languages. Our model outperforms other multilingual model baselines by a large margin. We evaluate on both part-of-speech tagging and morphological attribute prediction tasks with data from the Universal Dependencies repository (Nivre et al., 2018). However, this model is slow, very large, and difficult to deploy in practice.

We describe our solution for making this model small and practical enough to use in practice on a single CPU, while preserving quality. The final model is 27x faster than a BERT-based baseline model and 7x faster than a state-of-the-art LSTM-based model on CPU. It is 6 times smaller than the BERT-based model. Furthermore, most of the quality gains are preserved in the small model.

2 Multilingual Models for Sequence Labeling

We discuss two core models for addressing sequence labeling problems and describe, for each, training them in a single-model multilingual setting: (1) the Meta-LSTM (Bohnet et al., 2018), an extremely strong baseline for our tasks, and (2)

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3632–3636,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

Model	Multilingual?	Part-of-Speech F1	Morphology F1
Meta-LSTM	No	94.5	92.5
BERT	No	**95.1**	**93.0**
Meta-LSTM	Yes	91.1	82.9
BERT	Yes	**94.5**	**91.0**

Table 1: Macro-averaged F1 comparison of per-language models and multilingual models over 48 languages. For non-multilingual models, F1 is the average over each per-language model trained.

a multilingual BERT-based model (Devlin et al., 2019).

2.1 Meta-LSTM

The Meta-LSTM is the best-performing model of the CoNLL 2018 Shared Task (Smith et al., 2018) for universal part-of-speech tagging and morphological features. The model is composed of 3 LSTMs: a character-BiLSTM, a word-BiLSTM and a single joint BiLSTM which takes the output of the character and word-BiLSTMs as input. The entire model structure is referred to as Meta-LSTM.

To set up multilingual Meta-LSTM training, we take the union of all the word embeddings from the Bojanowski et al. (2017) embeddings model on Wikipedia[1] in all languages. For out-of-vocabulary words, a special unknown token is used in place of the word.

The model is then trained as usual with cross-entropy loss. The char-BiLSTM and word-biLSTM are first trained independently. And finally we train the entire Meta-LSTM.

2.2 Multilingual BERT

BERT is a transformer-based model (Vaswani et al., 2017) pretrained with a masked-LM task on millions of words of text. In this paper our BERT-based experiments make use of the cased multilingual BERT model available on GitHub[2] and pretrained on 104 languages.

Models fine-tuned on top of BERT models achieve state-of-the-art results on a variety of benchmark and real-world tasks.

To train a multilingual BERT model for our sequence prediction tasks, we add a softmax layer on top of the the first wordpiece (Schuster and Nakajima, 2012) of each token[3] and finetune on data from all languages combined. During training, we concatenate examples from all treebanks and randomly shuffle the examples.

Model	Embedding Size	Tokens	Hidden Units	Layers
Meta-LSTM	300	1.2M	8M	3
BERT	768	120k	87M	12
MiniBERT	256	120k	2M	3

Table 2: The number of parameters of each model. *Tokens* refers to the number of tokens of the embedding rows. For the Meta-LSTM, a word-based model, this is the number of words in training. For BERT, this means the size of the Wordpiece vocabulary. And *Hidden Units* refers to all units that are not among the embedding layer or and output layer.

Input	32 words	128 words
Relative Speedup on GPU		
Meta-LSTM	0.8x	0.2x
MiniBERT	**4.3x**	**2.6x**
Relative Speedup on CPU		
Meta-LSTM	6.8x	2.3x
MiniBERT	**27.7x**	**14.0x**

Table 3: Relative inference speedup over BERT. We see MiniBERT is the fastest on both CPU and GPU. CPU is an Intel Xeon CPU E5-1650 v3 @3.50GHz. GPU is an Nvidia Titan V.

3 Small and Practical Models

The results in Table 1 make it clear that the BERT-based model for each task is a solid win over a Meta-LSTM model in both the per-language and multilingual settings. However, the number of parameters of the BERT model is very large (179M parameters), making deploying memory intensive and inference slow: 230ms on an Intel Xeon CPU. Our goal is to produce a model fast enough to run on a single CPU while maintaining the modeling capability of the large model on our tasks.

Size and speed

We choose a three-layer BERT, we call *MiniBERT*, that has the same number of layers as the Meta-

[1] https://fasttext.cc/docs/en/pretrained-vectors.html

[2] https://github.com/google-research/bert/blob/master/multilingual.md

[3] We experimented with wordpiece-pooling (Lee et al., 2017) which we found to marginally improve accuracy but at a cost of increasing implementation complexity to maintain.

LSTM and has fewer embedding parameters and hidden units than both models. Table 2 shows the parameters of each model. The Meta-LSTM has the largest number of parameters dominated by the large embeddings. BERT's parameters are mostly in the hidden units. The MiniBERT has the fewest total parameters.

The inference-speed bottleneck for Meta-LSTM is the sequential character-LSTM-unrolling and for BERT is the large feedforward layers and attention computation that has time complexity quadratic to the sequence length. Table 3 compares the model speeds.

BERT is much slower than both MetaLSTM and MiniBERT on CPU. However, it is faster than Meta-LSTM on GPU due to the parallel computation of the transformer. The MiniBERT is significantly faster than the other models on both GPU and CPU.

Distillation

For model distillation (Hinton et al., 2015), we extract sentences from Wikipedia in languages for which public multilingual is pretrained. For each sentence, we use the open-source BERT wordpiece tokenizer (Schuster and Nakajima, 2012; Devlin et al., 2019) and compute cross-entropy loss for each wordpiece:

$$L(\mathbf{t}, \mathbf{s}) = H(\sigma(\mathbf{t}/T), \sigma(\mathbf{s}/T))$$

where H is the cross-entropy function, σ is the softmax function, $\mathbf{t}$ is the BERT model's logit of the current wordpiece, $\mathbf{s}$ is the small BERT model's logits and T is a temperature hyperparameter, explained in Section 4.2.

To train the distilled multilingual model *m*MiniBERT, we first use the distillation loss above to train the student from scratch using the teacher's logits on unlabeled data. Afterwards, we finetune the student model on the labeled data the teacher is trained on.

4 Experiments

4.1 Data

We use universal part-of-speech tagging and morphology data from the The CoNLL 2018 Shared Task (Nivre et al., 2018; Zeman and Hajič, 2018). For comparison simplicity, we remove the languages that the multilingual BERT public checkpoint is not pretrained on.

Model	Part-of-Speech F1	Morphology F1
*m*Meta-LSTM	91.1	82.9
*m*MiniBERT	93.7	88.6
*m*BERT	**94.5**	**91.0**

Table 4: Macro-averaged F1 comparison of multilingual models. Multilingual models are prefixed with '*m*'.

For segmentation, we use a baseline segmenter (UDPipe v2.2)[4] provided by the shared task organizer to segment raw text. We train and tune the models on gold-segmented data and apply the segmenter on the raw test of test data before applying our models.

The part-of-speech tagging task has 17 labels for all languages. For morphology, we treat each morphological group as a class and union all classes as a output of 18334 labels.

4.2 Tuning

For Meta-LSTM, we use the public repository's hyperparameters[5].

Following Devlin et al. (2019), we use a smaller learning rate of 3e-5 for fine-tuning and a larger learning rate of 1e-4 when training from scratch and during distillation. Training batch size is set to 16 for finetuning and 256 for distillation.

For distillation, we try temperatures $T = 1, 2, 3$ and use the teacher-student accuracy for evaluation. We observe BERT is very confident on its predictions, and using a large temperature $T = 3$ to soften the distribution consistently yields the best result.

4.3 Multilingual Models

Multilingual Modeling Results We compare per-language models trained on single language treebanks with multilingual models in Table 1 and Table 4. In the experimental results we use a prefix m to denote the model is a single multilingual model. We compare Meta-LSTM, BERT, and MiniBERT.

Multilingual Models Comparison *m*BERT performs the best among all multilingual models. The smallest and fastest model, *m*MiniBERT, performs comparably to *m*BERT, and outperforms *m*Meta-LSTM, a state-of-the-art model for this task.

[4] https://ufal.mff.cuni.cz/udpipe/models

[5] https://github.com/google/meta_tagger

	POS Tagging						Morphology					
Languages	kk	hy	lt	be	mr	ta	kk	hy	lt	be	mr	ta
Train Size	31	50	153	260	373	400	31	50	153	260	373	400
Meta-LSTM	61.7	75.4	81.4	91.1	72.1	72.7	48.5	54.5	69.7	74.0	59.1	71.0
BERT	75.9	84.4	88.9	94.8	**77.5**	**75.7**	47.8	44.8	**75.2**	82.8	64.0	72.9
*m*BERT	**81.4**	**86.6**	**90.0**	**95.0**	75.9	74.3	**64.6**	**51.1**	73.6	**87.5**	**64.2**	**73.8**
*m*Meta-LSTM	52.9	63.8	65.6	87.6	65.5	61.5	25.6	36.6	42.5	59.2	33.6	46.9
*m*MiniBERT	**76.6**	**86.0**	**86.9**	**95.0**	**75.4**	**74.6**	**59.7**	**47.6**	**64.8**	**81.6**	**59.4**	**71.7**

Table 5: POS tagging and Morphology F1 for all models on low-resource languages. Multilingual models are prefixed with '*m*'.

Multilingual Models vs Per-Language Models When comparing with per-language models, the multilingual models have lower F1. Kondratyuk (2019) shows similar results. Meta-LSTM, when trained in a multilingual fashion, has bigger drops than BERT in general. Most of the Meta-LSTM drop is due to the character-LSTM, which drops by more than 4 points F1.

4.4 Low Resource Languages

We pick languages with fewer than 500 training examples to investigate the performance of low-resource languages: Tamil (ta), Marathi (mr), Belarusian (be), Lithuanian (lt), Armenian (hy), Kazakh (kk)[6]. Table 5 shows the performance of the models.

BERT Cross-Lingual Transfer While Wu and Dredze (2019) shows effective zero-shot crosslingual transfer from English to other high-resource languages, we show that cross-lingual transfer is even effective on low-resource languages when we train on all languages as *m*BERT is significantly better than BERT when we have fewer than 50 examples. In these cases, the *m*MiniBERT distilled from the multilingual *m*BERT yields results better than training individual BERT models. The gains becomes less significant when we have more training data.

***m*MiniBERT Effectiveness** The multilingual baseline *m*Meta-LSTM does not do well on low-resource languages. On the contrary, *m*MiniBERT performs well and outperforms the state-of-the-art Meta-LSTM on the POS tagging task and on four out of size languages of the Morphology task.

Model	F1
BERT Supervised	90.6
Meta-LSTM English-Only	47.7
Meta-LSTM Hindi-Only	53.8
*m*Meta-LSTM	**83.4**
*m*BERT	82.9
*m*MiniBERT	79.5

Table 6: F1 score on Hindi-English codemixed POS tagging task. Each multilingual model is within 10 points of the supervised BERT model without having explicitly seen code-mixed data.

4.5 Codemixed Input

We use the Universal Dependencies' Hindi-English codemixed data set (Bhat et al., 2017) to test the model's ability to label code-mixed data. This dataset is based on code-switching tweets of Hindi and English multilingual speakers. We use the Devanagari script provided by the data set as input tokens.

In the Universal Dependency labeling guidelines, code-switched or foreign-word tokens are labeled as X along with other tokens that cannot be labeled[7]. The trained model learns to partition the languages in a codemixed input by labeling tokens in one language with X, and tokens in the other language with any of the other POS tags. It turns out that the 2nd-most likely label is usually the correct label in this case; we evaluate on this label when the 1-best is X.

Table 6 shows that all multilingual models handle codemixed data reasonably well without supervised codemixed traininig data.

5 Conclusion

We have described the benefits of multilingual models over models trained on a single language for a single task, and have shown that it is possible to resolve a major concern of deploying large

[6] The Universal Dependencies data does not have explicit tuning data for hy and kk.

[7] https://universaldependencies.org/u/pos/X.html

BERT-based models by distilling our multilingual model into one that maintains the quality wins with performance fast enough to run on a single CPU. Our distilled model outperforms a multilingual version of a very strong baseline model, and for most languages yields comparable or better performance to a large BERT model.

References

Irshad Bhat, Riyaz A Bhat, Manish Shrivastava, and Dipti Sharma. 2017. Joining hands: Exploiting monolingual treebanks for parsing of code-mixing data. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 324–330.

Bernd Bohnet, Ryan McDonald, Gonçalo Simões, Daniel Andor, Emily Pitler, and Joshua Maynez. 2018. Morphosyntactic tagging with a meta-BiLSTM model over context sensitive token encodings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2642–2652, Melbourne, Australia. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Daniel Kondratyuk. 2019. 75 languages, 1 model: Parsing universal dependencies universally. *CoRR*, abs/1904.02099.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, and Lene Antonsen et al. 2018. Universal dependencies 2.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018. 82 treebanks, 34 models: Universal dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. *CoRR*, abs/1904.09077.

Daniel Zeman and Jan Hajič, editors. 2018. *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, Brussels, Belgium.