IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.301

Volume 3, Issue 2, September 2023

Comparative Analysis of Pre-Training Models for Low-Resource Languages

Rekhadevi Kanyaboina[1] and Dr. Ramesh Kumar[2]
Research Scholar, Department of Computer Science and Engineering[1]
Research Guide, Department of Computer Science and Engineering[2]
NIILM University, Kaithal, India

Abstract*: Unlike major languages low-resource languages can suffer from a lack of expert-annotated benchmark and corpora resources, which makes it difficult to correctly apply state-of-the-art techniques. In this study, we take two approaches to addressing this shortage problem for the low-resource Filipino language. First, we provide a brand-new benchmark language modeling dataset in Filipino, WikiText-TL-39. Second, we show that language model finetuning methods such as BERT and ULMFiT can be used to consistently train robust classifiers in low-resource environments, with a validation error increase of no more than 0.0782 when finetuning with a privately owned sentiment dataset and reducing the number of training examples from 10K to 1K.*

Keywords: Language Model, Finetuning Techniques, Evaluation, Low-Resource Languages.

I. INTRODUCTION

In several domains, such as multitask learning (Radford et al., 2019; McCann et al., 2018), machine translation, and language modeling, neural network applications in Natural Language Processing (NLP) have demonstrated impressive success.

While effective, neural network algorithms are data-hungry and perform badly when dealing with low-resource languages or other scenarios where there is a dearth of data (Zoph et al., 2016). Moreover, these languages may not have simple access to mainstream language resources like pretrained word embeddings and expert-annotated corpora (Adams et al., 2017).

Creating correctly annotated corpora for these types of tasks is the perfect answer to the data scarcity problem, but the annotation labor is costly and time-consuming (Cotterell and Duh, models to be trained despite data scarcity (Cotterell and Duh, 2017).

Transfer learning is one way to address the problem of data scarcity. It allows models to be pre-trained and then refined on a smaller dataset, which reduces the amount of resources, computation, and time needed to create a workable model (Howard and Ruder, 2018).

In this work, we provide two new features: Initially, we provide the first large-scale, publicly accessible preprocessed unlabeled text corpora in the low-resource Filipino language under the name "WikiText-TL-39." Lastly, we show that transfer learning methods such as BERT (Devlin et al., 2018) and ULMFiT (Howard and Ruder, 2018) can be used to train robust classifiers in low-resource environments, with at most a 0.0782 increase in error observed when the number of training examples is lowered from 10K to 1K.

The public may access all of the pretrained models and datasets via an open repository1.

II. METHODOLOGY

Our evaluation process is as follows: Initially, we construct a vast unlabeled text corpus to facilitate the creation of transportable pretrained language models. Second, we use a privately held sentiment dataset to evaluate the efficacy of transfer learning. Next, we will progressively lower the number of training instances while tracking any changes in the validation accuracy.

We use BERT (Devlin et al., 2018) and Howard and Ruder (2018) as our two transfer learning methodologies.

ULMFiT

ULMFiT (Howard and Ruder, 2018) was intro- duced as a transfer learning method for Natural An AWD-LSTM is pre-trained using a language modeling task (Merity et al., 2017). The architecture stays the same after that, and the weights are recycled. Language modeling is applied once again, this time to the text of the target dataset, to help fine-tune the model to its distinct lexicon and peculiarities. Next, a "classification layer" that is intended exclusively for text categorization is added to improve the model. With adaption by Howard and Ruder (2018).

Russakovsky et al. (2015) describe the use of ImageNet-like pretraining in Computer Vision for language processing. As a foundation model, it begins with an AWD-LSTM (Merity et al., 2017) that has been pre-trained on a language modeling task. In order to match a downstream aim, it then refines this underlying model in two stages. First, the language model is syntactically fitted to the text of the target task. Second, a classification layer that is precisely tailored to the classification task is added to the model. Various techniques are used during finetuning to prevent catastrophic forgetting, which happens when the model loses most of the connections and information it learned during pretraining. State-of-the-art text categorization is provided by ULMFiT. It is attractive for use in contexts with low resources because of its remarkable ability to create comparable scores with as few as 1000 data samples. An overview schematic of ULMFiT is shown in Figure 1.

BERT

BERT is a transformer-based language model (Vaswani et al., 2017) that has achieved state-of-the-art results in several benchmarks. Its goal is to pretrain "deep bidirectional representations" that can be adjusted to various tasks (Devlin et al., 2018).

BERT's strength comes from a technique called Attention, which allows a network to give certain tokens in a sequence greater weight—essentially, "paying more attention to important parts" (Vaswani et al., 2017). More specifically, we compute attention on a set of queries packed as a matrix Q as follows, given key and value matrices K and V, respectively:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

where dk stands for the dimensions of the key matrix K. BERT is able to represent not just sequences but also the relative importance and weight of each word in a sequence with respect to other sequences and to itself by using attention.

It makes use of the Transformer architecture (Vaswani et al., 2017) in addition to Attention, which gives BERT bidirectionality. Transformers are sequence models that never use recurrent layers; instead, they only use feed-forward layers and attention processes. Reminders may be avoided for two reasons: Transformers are not sequential in nature like LSTMs or GRUs, hence they may be parallelized in the first place. Second, they make it possible to see a lot of text at once since they are not sequential. As a result, the text may employ attention processes and be bidirectional.

One unique feature of BERT is that it pretrains with changed tasks. next a traditional language modeling assignment would enable the model to "peek" directly at the next words since its bidirectionality gives it access to left-context. The authors propose that "masked-language" be used to mitigate this.

The pretraining architecture is mainly maintained during finetuning with little structural modifications. We pretrained the masked-language model and next-sentence prediction before sending model weights to downstream tasks like question answering and entailment. Source: Devlin et al. (2018) modeling," where the model's task is to identify hidden words in a sentence (Devlin et al., 2018). "Next-sentence prediction" was the name of the second pre-training activity that was used to create stronger connections between two phrases. In this activity, a target sentence is identified if it is most likely to appear before a source phrase (Devlin et al., 2018).

In addition to these improvements, BERT benefits from its depth, which makes it possible for it to collect more information and context. The lowest BERT model, known as BERT-Base, has a total of 110M parameters and consists of 12 layers with 12 attention heads and 768 units in each hidden layer. The larger twin of BERT-Large, with 24 layers (1024 units in each buried layer) and 16 attention heads, contains 340M parameters altogether.

WikiText-TL

A difficulty in adapting pretrained methods for low-resource languages is the lack of processed datasets large enough to construct robust pretrained models. Our benchmark dataset, WikiText-TL-39, is modeled after the first WikiText Long Term Dependency Language Modeling Dataset (Merity et al., 2016). The training set of this dataset consists of 39 million tokens, represented by the letter "39," where "TL" stands for Tagalog. The corpus statistics for WikiText-TL-39 are shown in Table 1.

Construction and Pre-processing

Compared to its English equivalent, Tagalog Wikipedia contains considerably less content pages (5,800,000 in English vs. 75,000), but it does include a list of certified "good" entries. This is why we limited our search to articles with titles starting with the letters A–Z and chose to scrape the material from every page mentioned in the Tagalog Wikipedia table of contents2. The material was extracted using two open-source Python scripts, Requests3 and Beautiful-Soup4.
All characters were converted to Unicode, and no escape was applied to any HTML elements. Tokenization and normalization were accomplished using the Moses Tokenizer (Koehn et al., 2007). We separated the corpus into training, validation, and test sets at a ratio of 70%-15%-15%. In contrast to Chelba et al. (2013), we made the decision to include words with a vocabulary count of at least three in our vocabulary building process. This resulted in a vocabulary size of 279,153 tokens. For each token in the test set that wasn't in the training set, we replace it with a special token.

III. EXPERIMENTS

BERT Pretraining

On our pre-parsed corpus and SentencePiece vocabularies7, we pretrain BERT Base models with 12 layers, 768 neurons per hidden layer, and 12 attention heads (a total of around 110M parameters) using Google's pretraining scripts. A number of vocabulary configurations (full 290K vs. 30K), casing (cased and uncased models), and training and warmup step counts (one million steps with 10K warmups and five million steps with 5K warmups) are tested. We stick to the original settings for the masked language model pretraining goal and utilize a 0.15 likelihood of a word being masked. Furthermore, we set the maximum back to the initial 20 masked language model predictions. All models employ a batch size of 256 and a maximum sequence length of 128. We use a learning rate of 1e-4 for all models.

AWD-LSTM Pretraining

We train an AWD-LSTM language model for ULMFiT using our provided corpus. We train a three-layer model with an embedding size of 400 and a hidden size of 1150. The dropout levels for the embedding, RNN input, hidden-to-hidden transition, and RNN output are (0.1, 0.3, 0.3, 0.4), respectively. We use a weight dropout of 0.5 for the LSTM's recurrent weight matrices.
To train the model across 30 epochs, 128 batch sizes, a learning rate of 1e-3, and a weight decay of 0.1 were used. We use the Adam optimizer and slanted triangular learning rate schedules. We train the model on a machine with a single NVIDIA Tesla V100 GPU.

Sentiment Classification Task

We enhance our model using a privately held sentiment classification dataset of 10,000 positive and 10,000 negative electronics reviews. From the original dataset, we randomly picked a full 10K-10K split of positive and negative evaluations, a 5K-5K split, a 1K-1K split, and a 100-100 split to imitate low-resource conditions. We modify pretrained models to each split for BERT and ULMFiT to evaluate performance with limited data. To evaluate performance, we use 1500 positive and 1500 negative evaluations from the same source. The same validation split is used without reduction for every split. This ensures consistency in analyzing validation accuracy changes as training samples decrease. Moses tokenizer (Koehn et al., 2007) preprocesses the dataset by retaining case and inserting spaces around punctuation. Filipino contractions like "can't" aren't preprocessed or given specific tokens since apostrophe contractions are rare.

Finetuning

We optimise our best cased and uncased BERT models for each sentiment categorization split. We finetune each configuration for 3 epochs at 2e-5 learning rate. We limit sequence length to 128 and batch size to 32. We finetune our pretrained AWD-LSTM language on each sentiment classification split for ULMFiT. The sentiment classification dataset is used for 10 epochs of language model finetuning at 1e-2 learning rate. We utilize weight decay of 0.1 and a batch size of 80 for the initial 10k-10k split and 0.3 and 40 for all additional splits. This language model refines ULMFiT's sentiment classification model in the final step. The last ULMFiT step is finetuned by progressive unfreezing. We finetune for five epochs, unfreezing the final layer until all layers are unfrozen on the fourth. We use a learning rate of 1e-3 and set Adam's α and β parameters to 0.8 and 0.7. We then track model performance improvements on a given validation set.

IV. RESULTS AND DISCUSSION

Pretraining Results

Using varying vocabulary sizes, casings, and pretraining methods, we were able to train eight models for BERT pretraining. Our best uncased model (which finally obtained a loss of 0.0935) was trained for 500K steps and fine-tuned for 5K steps using the reduced 30K SentencePiece vocabulary. In contrast, using the same 30K SentencePiece vocabulary, the best-cased model (which finally obtained a loss of 0.0642) needed 1M pretraining steps in addition to 10K warmup steps. We hypothesize that this is because learning and adapting to the casing needs more steps for the model. All of the BERT pretraining results are shown in Table 2. We obtained a final validation loss of 4.4857, or 1.5009 perplexity, using our AWD-LSTM language model for ULMFiT. The model required around eleven hours to complete 30 training epochs.

Finetuning Results

With an accuracy improvement of 0.006, the uncased model outperformed the cased model for BERT during fine-tuning on the first 10K–10K split. It is clear that an error increase of no more than 0.0617 for cased models and 0.0954 for uncased models occurs when the number of training samples is reduced from 10K to 1K. The error increases considerably by 0.1484 in the cased model and by 0.2554 in the uncased model as the number of training examples reduces to the 100-100 split. Comparable results may be seen by analyzing the validation sets of each individual split in addition to the validation set of the first 10K-10K split. For the cased models, fine-tuning on the 1K-1K split results in just a 0.038 gain in accuracy, whereas fine-tuning on the 100-100 split results in a 0.23 increase. For the uncased models on the 1K-1K split and the 100-100 split, we get an increase in error of 0.0437 and 0.248, respectively. The complete set of BERT finetuning results is shown in Table 3.

V. DISCUSSION

The previously described research offers empirical proof that language model pretraining might be advantageous in settings with limited resources. The improved models showed consistent performance even with fewer training cases by evaluating on the same validation set. ULMFiT fared somewhat better than BERT when adjusted on (a difference of 0.0201).

VI. CONCLUSION

We show how language model finetuning methods may be useful in low-resource settings, especially when there aren't enough expert-annotated examples. Pretraining using language models has two advantages. First of all, it just requires unlabeled text corpora, which are almost always accessible, even in settings with limited resources. Second, after pre-training is finished, finetuning may be affordably performed on the same pretrained model several times. This allows researchers to use a small proportion of available resources to construct trustworthy baselines even in contexts with restricted resources. The choice of fine-tuning technique requires weighing consistency against cost. Since ULMFiT's pretraining phase is somewhat less expensive than BERT's, we propose it as a baseline for finetuning in most circumstances. Although powerful, BERT can only be used in scenarios where a pretrained model is available or when the resources permit it due to its high compute and memory requirements.

VII. ACKNOWLEDGMENTS

The authors applaud the TensorFlow Research Cloud (TFRC) project for making BERT model pretraining more widely available.

REFERENCES

[1]. Oliver Adams, Adam Makarucha, Graham Neubig, Steven Bird, and Trevor Cohn. 2017. Cross-lingual word embeddings for low-resource language model- ing. In Proceedings of the 15th Conference of the European Chapter of the Association for Computa- tional Linguistics: Volume 1, Long Papers, pages 937–947, Valencia, Spain. Association for Compu- tational Linguistics.

[2]. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Ben- gio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

[3]. Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. 2013. One bil- lion word benchmark for measuring progress in sta- tistical language modeling. CoRR, abs/1312.3005.

[4]. Ryan Cotterell and Kevin Duh. 2017. Low- resource named entity recognition with cross- lingual, character-level neural conditional random fields. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 91–96, Taipei, Tai- wan. Asian Federation of Natural Language Process- ing.

[5]. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understand- ing. arXiv preprint arXiv:1810.04805.

[6]. Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In Proceedings of the 56th Annual Meeting of the As- sociation for Computational Linguistics (Volume 1: Long Papers), pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

[7]. Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondˇrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

[8]. Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language de- cathlon: Multitask learning as question answering. arXiv preprint arXiv:1806.08730.

[9]. Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and Optimizing LSTM Language Models. arXiv preprint arXiv:1708.02182.

[10]. Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture mod- els. CoRR, abs/1609.07843.

[11]. Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI Blog, 1(8).

[12]. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, An- drej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. Ima- geNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252.

[13]. Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In Proceedings of the 54th An- nual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715– 1725, Berlin, Germany. Association for Computa- tional Linguistics.

[14]. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information pro- cessing systems, pages 5998–6008.

[15]. Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In Proceedings of the 2016 Conference on Empirical Methods in Natu- ral Language Processing, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.