Causal and Masked Language Modeling of Javanese Language using Transformer-based Architectures

Wilson Wongso
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
wilson.wongso001@binus.ac.id

David Samuel Setiawan
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
david.setiawan002@binus.ac.id

Derwin Suhartono
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
dsuhartono@binus.edu

2021 International Conference on Advanced Computer Science and Information Systems (ICACSIS) | 978-1-6654-4264-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICACSIS53237.2021.9631331

***Abstract*—Most natural language understanding breakthroughs occur in popularly spoken languages, while low-resource languages are rarely examined. We pre-trained as well as compared different Transformer-based architectures on the Javanese language. They were trained on causal and masked language modeling tasks, with Javanese Wikipedia documents as corpus, and could then be fine-tuned to downstream natural language understanding tasks. To speed up pre-training, we transferred English word-embeddings, utilized gradual unfreezing of layers, and applied discriminative fine-tuning. We further fine-tuned our models to classify binary movie reviews and find that they were on par with multilingual/cross-lingual Transformers. We release our pre-trained models for others to use, in hopes of encouraging other researchers to work on low-resource languages like Javanese.**

***Index Terms*—Javanese Language Modeling, Low-resource Languages, Natural Language Understanding, Transformers, Deep Learning**

I. Introduction

Natural language understanding is a particular field that attained major groundbreaking results due to the recent advancement of deep learning [1]. However, most of these milestones took place in high-resource languages that are widely spoken such as English, Mandarin Chinese, and Hindi.

Languages like Javanese, Sundanese, Cebuano, and various other regional languages have barely received any attention nor practical benefits from deep learning researchers [2], mainly due to data scarcity. The Javanese language, more specifically, could greatly benefit from such works given that the language is the 26th most spoken language in the world, with over 68 million speakers worldwide [3].

Multiple business processes could be aided with the presence of language models for uses like sentiment analysis, neural machine translation, text generation, etc. This is very much evident in generic, pre-trained language models like the OpenAI GPT-3 [4] where various business use cases can be implemented using the language model.

Catering to this issue, we aim to provide a baseline benchmark to the field of language modeling of the Javanese language. And at the same time, hope to encourage other researchers to work on natural language understanding of other less popular languages.

To do so, we compared and pre-trained different Transformer-based [5] architectures such as the OpenAI GPT-2 [6], BERT [7], RoBERTa [8], and DistilBERT [9] models on causal and masked language modeling of the Javanese language. The corpus used for pre-training consists of over 80 thousand Javanese Wikipedia articles on different topics.

To test out the resultant models, a two-step fine-tuning to the task of text classification was performed, specifically on Javanese movie reviews. Our single models were mostly able to outperform a multilingual BERT model and traditional machine learning algorithms in this task, while performing similarly to an XLM-RoBERTa model. An ensemble of the trained models was able to outperform larger multilingual models. After training these different models, they are then deployed via the Hugging Face Transformers Model Hub[1], where other users could freely utilize our models' pre-trained weights.

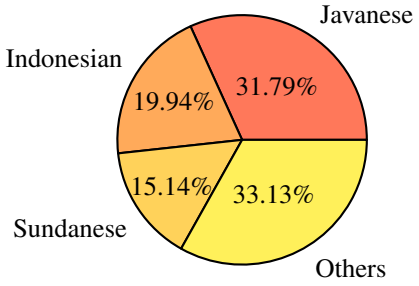

Fig. 1. Percentage of native speakers of languages in Indonesia [10].

II. Related Works

A. Deep Learning Language Models

Modeling a natural human language has proven to be one of the hardest tasks to be performed by a computer, mainly due to its complex structure and virtually limitless grammatical rules. A method called deep learning [1] has shown promising results as the algorithm figures out the underlying structure of

[1] https://hf.co/w11wo

data without needing to be programmed explicitly – translating to the recent successes in the field of natural language understanding.

The development of deep learning language model architectures commenced with the introduction of Recurrent Neural Networks (RNN) [11]. Unlike ordinary neural networks, RNN introduced the concept of time-dependency, where new data are fed into the neural network in sequence over time. This is parallel with the structure of languages, where the order of words and composition of words are critical.

A multitude of RNN-like architectures built to cater to issues faced by RNNs has been proposed previously. For instance, the Long short-term memory (LSTM) network [12] was introduced to deal with the problem of vanishing gradients and exhibiting long-term dependency. LSTM differs from plain RNN as it contains different gates: an input gate, an output gate, and a forget gate. These gates help to control which sections of information should be carried over by the network in the long run.

B. Transformers

A major breakthrough in language modeling happened after the introduction of the attention mechanism [13]. It is especially useful in the case where an encoder-decoder approach is necessary like in neural machine translation, for example. The attention mechanism works by training the model to find parts of a source sentence that are relevant to a specific part of the target sentence. This method is therefore optimal in remembering long-term dependencies as reflected in most natural languages.

The Transformer architecture takes the idea of attention and takes it to the extreme of removing the usage of recurrence and convolutions entirely [5]. Instead, it claims that the attention mechanism is all that is needed to train an encoder-decoder architecture.

Generally, the attention function is described as the following equation:

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

where Q represents the attention vector, K being the keys of the query and V corresponding to its value, normalized by d_k, the dimension of either of the three vectors.

Transformers have become one of the most widely used architectures for natural language processing due to their superiority over other neural network architectures especially for the case of sequence-to-sequence modeling. Furthermore, it works well with a large amount of data, leveraging the model's large number of parameters, and is parallelizable in a multi-accelerator setup.

C. Transfer-Learning

The Transformer architecture enables us to pre-train models on large datasets, which can then be fine-tuned to downstream tasks like text classification, summarization, translation, and many others. The convention of transferring a general language model to a downstream language model only started in recent years. Such practice became popular after its initial successes in the realm of Computer Vision.

Most image-related tasks nowadays transfer learn pre-trained neural network weights trained on the ImageNet dataset with over 14 million images and 20,000 categories [14]. Rather than training the model to learn from randomly initialized weights, the practice of transfer-learning is proven to be more effective in terms of saving up training time as well as achieving a more accurate result [15].

Unsurprisingly, transfer-learning was successfully implemented on language models and produced state-of-the-art results. One of the earliest natural language transfer-learning techniques called Universal Language Model Fine-tuning (ULMFiT) [16], did not only reduce error rates on text classification, but also significantly reduced the amount of data required for training. Transfer-learning is also shown to benefit greatly from discriminative fine-tuning, whereby different layers of the neural network are trained with different learning rates.

Unfortunately, it's quite difficult and tedious to transfer learn and fine-tune these Transformer models and apply them to specific tasks. Catering to this issue, Hugging Face's *Transformers* library enables us to fine-tune different pre-trained Transformers and easily apply them to various downstream tasks [17].

This open-source library stores a wide variety of Transformer architectures in a centralized hub under a unified API, enabling us to play around with different models, compare them, and apply them to a variety of different tasks with relative ease. Given that the framework is easily extensible and fast, it is, therefore, the framework of choice to training our Javanese language models.

D. Javanese Language Modeling

Before our work, other related researches similarly attempt to perform language modeling on the Javanese language. Most of them, however, immediately jumps to downstream tasks of POS tagging [18] and text classification [19], without providing a general language model. Those models are thus unable to be fine-tuned to other downstream natural language understanding tasks.

More versatile approaches include the training of a multilingual Transformer-based language model using a multilingual corpus that includes the Javanese language in it. For instance, a BERT model [7] could be trained on Wikipedia texts consisting of multiple languages at once. Alternatively, a cross-lingual language model like XLM [20] could do similarly given a large, multilingual corpus.

E. Transformer Architectures

Since the publication of the Transformer paper [5], there have been multiple renditions to the original architecture, each with its own capabilities and limitations.

1) OpenAI GPT-2: The OpenAI GPT-2 [6] is a huge Transformer-based language model with approximately 1.5 billion parameters in its largest rendition, trained on a dataset of 8 million web pages. It is the successor of the first GPT architecture [21] and is similarly trained based on a causal language modeling task and achieved 7 out of 8 state-of-the-art results in a zero-shot setting. This research proves that the language model can learn to do tasks like reading comprehension, summarization, translation, etc., without any explicit supervision, and in turn, it studied the natural occurring demonstration.

2) BERT: Like the OpenAI GPT-2, BERT is a pre-trained language model [7] trained on English Wikipedia documents and BooksCorpus [22]. However, it is intrinsically different from ordinary sequence-to-sequence models as BERT is naturally bidirectional. Furthermore, it is trained on two tasks: masked language modeling and next-sentence prediction (NSP) tasks. BERT achieved state-of-the-art results on GLUE [23], MultiNLI [24], and SQuAD [25] benchmarks.

There have been further modifications to the original BERT model, some of which aim to reduce the training time of BERT. As an example, the ALBERT model [26] applies parameter-reduction techniques which shrink the model's parameters and, in turn, reduces memory consumption while speeding up its training time. Similarly, the DistilBERT model [9] implements knowledge-distillation [27], [28] and thus proposes a lighter, yet equally as effective, modification to the original BERT. DistilBERT restored 97% of BERT's performance while being 60% faster and 40% smaller in size.

3) RoBERTa: While BERT has proven to be effective in most natural language understanding tasks, authors of RoBERTa have shown that the BERT model was in fact undertrained [8]. Unlike BERT, RoBERTa removes the next-sentence prediction task and increases the number of data for pre-training. As a result, they outperformed BERT's results in SQuAD [25], MNLI [24], and RACE [29] benchmarks.

III. Methodology

We propose the following research pipeline reflected in Figure 2. It follows the regular transfer-learning pipeline that has been used in various other NLU training schemes such as BERT [7] and ULMFiT [16], to name a few.

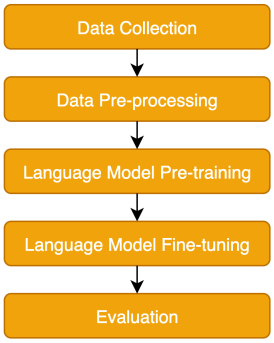

Fig. 2. Research Methodology Pipeline

A. Dataset

The dataset used to train the language model is a collection of the latest 80,000 Javanese Wikipedia articles retrieved in December 2020. Such a collection is also known as Wikitext and is often used to pre-train language models like BERT [7]. Advantages of using Wikitext as our pre-training corpus include its rich vocabulary and diverse topics to better generalize our language model. However, this does mean that the quality of our pre-training dataset, and hence the output of our language models, depends wholly on the quality of text written by different Javanese Wikipedia contributors with various levels of writing ability. Nonetheless, the main purpose of using Wikipedia text for pre-training is to simply teach linguistic context to the language model and thus may carry over the fluency level and biases of the authors of the Wikipedia texts. In total, the documents accumulate to a total size of 319MB.

Training language models with a task of causal language modeling and masked language modeling do not require us to provide labels, i.e. they are self-supervised. In other words, the training method relies merely on the corpus itself. As for causal language modeling, the labels are simply the next word in the sequence. While for masked language modeling, the masked-out words are the labels to be predicted by the model.

B. Pre-processing

There are several possible ways to encode text data into numeric representations interpretable by a neural network. Different language modeling tasks require different labeling methods of their own. Nevertheless, the general pre-processing pipeline of all four of our models is as depicted in Figure 3. Like the main pipeline we proposed in Figure 2, this pre-processing scheme follows the regular practices of handling text data for language models.

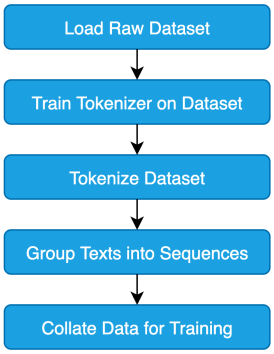

Fig. 3. Data Pre-processing Pipeline

1) Tokenization: To begin with the pre-processing pipeline, a corpus of words is required to lay out all the possible vocabularies and list the respective token's encoded value. For our GPT-2-based and RoBERTa-based models, Hugging Face's implementation of Byte-level Byte-Pair Encoding Tokenizer [30] is used to tokenize our dataset. The tokenizer starts by building byte base tokens of all possible characters

and later learns to merge them into subwords according to their frequency occurrence.

The GPT-2-based model's tokenizer is set to have a vocabulary size of 50,257, while the RoBERTa-based model's tokenizer has a vocabulary size of 50,265. Their vocabulary sizes accord to their respective English models' tokenizers and the same special tokens used in the English models were added to the tokenizer as well.

On the other hand, our BERT-based and DistilBERT-based model used Hugging Face's implementation of BERT's WordPiece Tokenizer [31]. This tokenizer is very similar to the Byte-level Byte-Pair Encoding Tokenizer, but is different in terms of selecting which subwords were to be formed. Instead of choosing the most frequent byte-pairs to be merged, it forms a new subword that maximizes the likelihood of the subword being used as training data once added to the vocabulary. Unlike the two previous models, the BERT-based and DistilBERT-based model has a significantly smaller vocabulary size of 30,522.

2) Text-Grouping: Once these tokenizers have been successfully trained, the corpus is encoded into their corresponding numerical-token representations and their designated special tokens along with attention masks are added to facilitate the Transformer input pipeline.

Moreover, these encoded texts are grouped into different sequences to limit the sequence length per input. Specifically, they are grouped into blocks/sequences of size 128, whereby longer sequences are thus split into shorter subsequences, and shorter sequences are concatenated together.

3) Data Collation: Finally, the grouped texts are collated according to the different language modeling tasks. Causal language modeling doesn't require this specific step since no special operations are required to facilitate training. Masked language modeling, however, relies on this step to apply random masking. The masking probability of each token, such that it may become a label for training, is set to 0.15.

C. Language Model Pre-training

1) Transfer-Learning: Once the pre-processing of data is finished, the pre-training of each Transformer-based language model then proceeds. The base models pre-trained include the OpenAI GPT-2 [6], BERT [7], DistilBERT [9] and RoBERTa [8], all of which utilize Hugging Face's open-source implementation with a PyTorch [32] backend. 20% of the pre-processed dataset was also left for validation purposes.

Rather than initializing our models' weights from scratch, the respective English models were loaded. Namely, the pre-trained weights of the OpenAI GPT-2 model, the BERT base model (uncased), the RoBERTa base model, and the DistilBERT base model (uncased), were initialized to become the bases of our Javanese language models. All of these pre-trained weights were retrieved from the Hugging Face Transformers Model Hub.

Since the English models were trained in a completely different corpus, their embedding representations are thus irrelevant to our language model. However, there may be identical tokens in both corpora. Therefore, if an identical token was to be found in the English corpus, its corresponding embedding representation is brought to our Javanese language model. Otherwise, if the token is non-existent in the English corpus, the mean of all English word embeddings weights were used for that particular token.

This approach is very much like the fine-tuning process proposed in ULMFiT [16] and is effective in the training of GPorTuguese-2 [33], a Portuguese GPT-2-based causal language model. This way, the word-embeddings of our Javanese language model are semi-pre-trained and already contain several representations of tokens in the Javanese corpus.

2) Gradual Unfreezing and Discriminative Fine-tuning: During training, gradual unfreezing and discriminative fine-tuning are applied as suggested in the ULMFiT paper [16]. Instead of training all of our models' layers at one go, only the last few layers are first trained for a few epochs, while the remaining layers are kept frozen.

In the proceeding epochs, the preceding last few layers were gradually unfrozen and are trained in a lower learning rate. This process is repeated until all the layers in the model are unfrozen. Table I reflects our setup in more detail.

TABLE I
GRADUAL UNFREEZING AND DISCRIMINATIVE FINE-TUNING SETUP IN PRE-TRAINING.

Epoch	Block to Unfreeze	Learning Rate
1	Embeddings & Head	2×10^{-3}
2	Blocks 9-12	1×10^{-3}
3	Blocks 5-8	5×10^{-4}
4-5	Blocks 1-4	5×10^{-5}

Further, all phases of training are facilitated by the AdamW optimizer [34], coupled with linear annealing of the set learning rate. As for the remaining hyper-parameters, the default values in Hugging Face's `TrainingArguments` are used.

D. Language Model Fine-tuning

1) Javanese IMDb Movie Reviews Dataset: As of the time of writing, there are no open-source Javanese classification datasets. To test out our pre-trained models, we prepared a Javanese version[2] of the IMDb Movie Reviews dataset [35] by translating the original English dataset to Javanese. Specifically, the translation process was handled by a multilingual MarianMT Transformer, `opus-mt-en-mul` [36].

Although the translation model managed to achieve a BLEU score of 7.8 on the official Tatoeba benchmark dataset, its translation output may not always be perfect. Regardless, it remains as one of the best English-to-Javanese translation models and is hence the translator of choice when it comes to creating this dataset. It does mean that the quality of the translated dataset depends fully on the quality of the translation model, but nevertheless provides a trivial way of creating a quick benchmark dataset on a low-resource language like Javanese.

[2]https://hf.co/datasets/w11wo/imdb-javanese

Like the original version, the Javanese IMDb dataset comprises 25,000 movie reviews for training and another 25,000 for testing. Additionally, there are 50,000 unlabeled movie reviews for the first half of the fine-tuning process. There are only two possible categories in the dataset: either a positive or a negative review.

2) Two-step Fine-tuning: To then apply our pre-trained language model on a downstream task like text classification, a two-step fine-tuning is conducted to our models.

Firstly, our Javanese language models were fine-tuned, with the same language modeling task, on the IMDb Movie Review corpus, such that it would gain an understanding of movie review-related texts. Just as its pre-training process, this step is self-supervised and could thus use the entire 100,000 movie reviews available. Those movie reviews undergo the same pre-processing pipeline as the data used during the pre-training process. In the first fine-tuning step, all the models used a learning rate of 2×10^{-5} and are trained for a total of 5 epochs.

Afterward, to perform classification, the models were fine-tuned by replacing their language model head with a fully connected layer. The final layer outputs the same number of neurons as the number of possible target classes, i.e. 2, for 5 epochs with a learning rate of 2×10^{-5} as well. Like the pre-training stage, the two-step fine-tuning process used the AdamW optimizer [34] and linear annealing of the set learning rate.

IV. Result and Discussion

Different experiments were conducted according to the setup described in Section III. It began with the pre-training stage where the respective English language models are first transferred to the Javanese Wikipedia corpus. After pre-training, the language models are then fine-tuned with the same respective tasks on the Javanese IMDb movie review dataset. Table II describes the perplexity achieved by the different models on the validation subsets of the two corpora.

TABLE II
Perplexity of the trained Javanese language models.

Model	#params	LM Task	Wikipedia	IMDb
Javanese BERT	110M	Masked	**22.00**	**19.87**
Javanese RoBERTa	124M	Masked	33.30	20.83
Javanese DistilBERT	66M	Masked	23.54	21.01
Javanese GPT-2	124M	Causal	25.39	60.54

Finally, at the last step of fine-tuning, the language models learned to classify Javanese movie reviews with the replacement of their language model head with a fully connected layer that outputs the two possible target classes.

To compare our results, we also performed the same classification task using a multilingual mBERT model [7], a cross-lingual XLM-RoBERTa model [37], and scikit-learn's [38] traditional machine learning algorithms such as Logistic Regression and Multinomial Naive Bayes classifier. We also performed a simple mean-probability ensembling on four of our monolingual models. Table III shows the accuracy of every model on the test subset.

TABLE III
Result of text-classification on Javanese IMDb movie review test set.

Model	#params	Accuracy
Javanese BERT	110M	76.37
Javanese RoBERTa	124M	77.70
Javanese DistilBERT	66M	76.04
Javanese GPT-2	124M	76.70
Ensembling	-	**78.92**
mBERT	167M	76.13
XLM-RoBERTa	278M	78.13
Logistic Regression with Count Vectorizer	91K	73.45
Logistic Regression with TF-IDF Vectorizer	91K	73.61
Naive Bayes with Count Vectorizer	-	70.84
Naive Bayes with TF-IDF Vectorizer	-	70.67

Out of all the models we have pre-trained, our Javanese RoBERTa model achieved the highest accuracy of 77.7% on the test subset of the Javanese IMDb movie review dataset, despite it having a higher perplexity compared to our Javanese BERT model. It should, however, be noted that the former has more parameters than the latter, which could explain the slight difference in classification results.

Compared to the benchmark models, most of our single models were able to outperform both the multilingual mBERT model as well as the baseline traditional models. However, they were all unsurprisingly outperformed by the XLM-RoBERTa model which has more than twice the number of parameters. Moreover, XLM-RoBERTa is trained on over 2.4TB of data, which allows the large model to generalize to multilingual tokens that may be present in the dataset, hence the slightly better classification result. Regardless, the ensembled model of all four of our models yielded the highest accuracy score of 78.92%.

Before XLM-RoBERTa, most low-resource downstream language tasks are done using the mBERT model since it provides a quick result compared to training an entirely new monolingual model. But as our results show, monolingual pre-trained models perform better than mBERT for low-resource languages like Javanese.

We suspect that if there are more Javanese datasets to pretrain on, i.e. Javanese being a high-resource language like English or Chinese, these monolingual models would easily outperform their multilingual counterparts. A trivial example of this case is clearly the English language where monolingual models could outperform large multilingual models, despite being smaller in size; mainly due to the discrepancy in the pre-training dataset size and chosen tokenizer [39].

As for the traditional machine learning algorithms, they displayed a decent result in classifying text data since the dataset is pre-processed using either Count Vectorizer or TF-IDF Vectorizer; producing high dimensional, sparse data. They thus serve as a good and quick baseline for text classification where these models are generally performant.

Overall, the results that our models attained are in line with what their respective papers have concluded. For instance,

the DistilBERT-based model was able to recover over 97% of the BERT-based model's result. Likewise, the RoBERTa-based model outperformed the BERT-based model like its paper suggested, while GPT-2-based performed similarly to the BERT-based model.

Ensembling of the four models led to an even greater accuracy compared to the single models and the larger multilingual models. Moreover, the usage of gradual unfreezing and discriminative fine-tuning helped us to train these models faster compared to training without the two regimes.

A. Impact of Transferring English Word-Embeddings

As outlined in Section III.C.1, our Javanese language models' word-embeddings were not initialized from scratch. Rather, they copy the embedding weights of tokens that exist in both English and Javanese corpora. This greatly sped up our model's learning process given that it has prior knowledge of a significant amount of tokens shared between the two vocabularies.

This method, however, came with a side-effect of creating a partially bilingual model, since some of the English tokens are carried over. In theory, our models should at least understand some context in the English language – though the Javanese language remains as their main language. To test this idea out, we evaluated our Javanese IMDb movie review classifier on the original English IMDb movie review [35] test set, without further fine-tuning. Table IV reflects the classification result of our models.

TABLE IV
EVALUATION RESULT OF TEXT-CLASSIFICATION ON ENGLISH IMDB MOVIE REVIEW TEST SET.

Model	#params	Accuracy
Javanese BERT	110M	69.93
Javanese RoBERTa	124M	**80.07**
Javanese DistilBERT	66M	65.12
Javanese GPT-2	124M	71.18

As shown in Table IV, our Javanese RoBERTa IMDb classifier achieved the highest accuracy in the task of classifying the test set of English IMDb movie reviews. Although these results could barely compete with the original English models, it is interesting to analyze why the RoBERTa-based model achieved a significantly higher accuracy out of all the four models trained.

The possible reason behind these results is likely because, during the first pre-training stage, the RoBERTa-based and the GPT-2-based models copied $11,959$ and $11,954$ embedding vectors from the English corpus respectively. On the other hand, the BERT-based and the DistilBERT-based models only copied $7,656$ and $7,655$ embedding vectors from the English corpus respectively.

Given the difference in vocabulary sizes and the number of English tokens copied, it is thus unsurprising that the RoBERTa-based model was able to achieve an evaluation accuracy of 80.07% without the need to be fine-tuned to the English IMDb dataset. It remains unclear why the GPT-2-based model wasn't able to emulate the same result as the RoBERTa-based model despite it having a similar vocabulary size and the number of English tokens carried over.

V. CONCLUSION

We pre-trained and compared different Transformer-based Javanese language models on two different tasks of causal and masked language modeling. In the process, we applied transfer learning, gradual unfreezing, and discriminative fine-tuning which greatly sped up the training process. Finally, we fine-tuned these language models on the task of text classification of the IMDb movie review dataset which has been translated into the Javanese language. Our ensembled model was able to attain an accuracy of 78.92% in classifying between positive and negative reviews.

With these pre-trained models, we could perform other downstream natural language understanding tasks in the Javanese language, such as extractive question answering, summarization, named entity recognition, and many others. These language models could attain better results if a greater and more diverse corpus was used during pre-training. It might also be helpful to establish a strong and robust benchmark dataset of the Javanese language, like that of GLUE [23], to standardize the field of Javanese language modeling and encourage other researchers to work on this very language.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] S. Ruder, "Why You Should Do NLP Beyond English," http://ruder.io/nlp-beyond-english, 2020.

[3] "What are the top 200 most spoken languages?" Dallas, TX, USA, Feb 2021. [Online]. Available: https://www.ethnologue.com/guides/ethnologue200

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[9] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[10] Badan Pusat Statistik, "Kewarganegaraan, suku bangsa, agama, dan bahasa sehari-hari penduduk indonesia: Hasil sensus penduduk 2010." http://www.bps.go.id/website/pdf_publikasi/watermark%20_Kewarganegaraan,%20Suku%20Bangsa,%20Agama%20dan%20Bahasa_281211.pdf.

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[15] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *arXiv preprint arXiv:1411.1792*, 2014.

[16] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[17] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.

[18] R. A. Pratama, A. A. Suryani, and W. Maharani, "Part of speech tagging for javanese language with hidden markov model," *Journal of Computer Science and Informatics Engineering (J-Cosine)*, vol. 4, no. 1, pp. 84–91, 2020.

[19] A. F. Hidayatullah, S. Cahyaningtyas, and R. D. Pamungkas, "Attention-based cnn-bilstm for dialect identification on javanese text," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, pp. 317–324, 2020.

[20] G. Lample and A. Conneau, "Cross-lingual language model pretraining," *arXiv preprint arXiv:1901.07291*, 2019.

[21] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[22] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 19–27.

[23] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.

[24] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," *arXiv preprint arXiv:1704.05426*, 2017.

[25] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[26] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[27] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 535–541.

[28] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[29] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "Race: Large-scale reading comprehension dataset from examinations," *arXiv preprint arXiv:1704.04683*, 2017.

[30] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[31] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 5149–5152.

[32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.

[33] P. Guillou, "Gportuguese-2 (portuguese gpt-2 small): a language model for portuguese text generation (and more nlp tasks...)," 2020.

[34] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[35] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[36] J. Tiedemann, "The tatoeba translation challenge – realistic data sets for low resource and multilingual mt," 2020.

[37] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," 2020.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[39] P. Rust, J. Pfeiffer, I. Vulić, S. Ruder, and I. Gurevych, "How good is your tokenizer? on the monolingual performance of multilingual language models," 2021.