Feasibility of Improving BERT for Linguistic Prediction on Ukrainian corpus

Hanna Livinska[1] [0000-0001-9676-7932] and Oleksandr Makarevych[2] [0000-0002-2084-5209]
[1, 2]Taras Shevchenko National Univeristy of Kyiv
Volodymyrska str. 64, Kyiv, 01601, Ukraine
[2]oleksandrmakarevych@knu.ua

Abstract. What makes BERT (Bidirectional Encoder Representations from Transformers) different from other recently published language models is the fact that it supports numerous languages, including Ukrainian. The first purpose of this research is to look at how well the BERT model is actually trained taking into account that Ukrainian language is a low-resource one. The second one is to create a hand-picked dataset to further train the published model and to compare the results of two models. Training the model in this research is based on texts written in Ukrainian, including fairytales, novels and stories for kids. This specific dataset is chosen mainly because of the fact that stories for kids have not so big of a vocabulary considering its audience and the fact that those stories usually follow similar paths in their narration. Our model is trained on two tasks as in the original paper: masked token prediction and next sentence classification. The model shows a clear improvement for Ukrainian language compared to the original version.

Keywords: NLP, BERT, transformer, attention, next sentence prediction, machine learning.

1 Introduction

In the recent years, Data Science has been a buzzword in the Computer Science industry. It has been attracting both theoretical scientists and practicing developers with different backgrounds and has been called one of the best jobs of the 21st century by the Harvard Business Review. This resurrected interest might be attributed to the fact that today we have more computational power and storage which opens a door for developing new, more powerful machine learning models.

Amongst other reasons, Data Science attract so many researchers in the field because it requires not only classical programming skills, but also strong knowledge of Calculus, Probability, Statistics, desire to experiment with data and creative thinking in order to push industry further and find insights in the data that has not been seen before.

Today, one of the most prominent fields in the area of Computer Science is Natural Language Processing (NLP) as it allows us to analyze natural languages to solve problems we could not find solutions to before. It was shown that language model pre-training is an effective tool for many natural language processing tasks ([1], [2],

[3], [4]). In particular, as attention mechanism and transformer architecture have been developed, it allowed new models to reach state-of-the-art performance in the area of text generation and understanding. This research focuses on pre-training BERT-model (Bidirectional Encoder Representations from Transformers) which utilizes both attention mechanism and transformer to see if it is possible to improve results when working with low-resource language. It was originally developed and published in 2018 ([5], [6]) and it is considered one of the best in the industry.

In 2012, the deep neural network submitted to ImageNet Large Scale Visual Recognition Challenge by Alex Krizhevsky and Ilya Sutskever [7] demonstrated that deep learning was a viable strategy for machine learning and thus led to increased interest in deep learning and machine learning research. The success of AlexNet model was caused by the fact that lower layers of the model learned low-level features such as edges, while higher layers focused on understanding higher level concepts like patterns and entire parts of objects. A key property of an Image-Net-like dataset is thus to encourage a model to learn features that will likely generalize to new tasks in the problem domain. Previous studies [7] revealed that state-of-the-art models for tasks such as reading comprehension and natural language inference did not in fact posses deep natural language understanding but rather picked up on cues to perform special pattern matching. Attention mechanism is used to solve this problem and give the model the ability to understand language contexts better [5].

2 Architecture

In a gist, BERT model is comprised of six encoders and decoders stacked together. Each decoder processes the input sequence consequently which results in outputting probabilities for a missing word from the model's vocabulary. The input for the encoder is comprised of three main parts:

- Token embeddings
- Segment embeddings
- Position embeddings

Token embeddings are actual embeddings of the words in the sequence. Each sequence starts with a special token *[CLS]* to denote the beginning of the sequence. The two sentences are separated by a special *[SEP]* token. Segment embeddings are used to denote whether the token belongs to the first or the second sentence in the sequence. Position embeddings represent the numerical place of the token in the overall sequence. All these three parts are combined together to form the input that is then processed by attention mechanism to produce state-of-the-art results.

To understand attention mechanism better, first, let us look at the problem with Sequence-to-Sequence model.

A Sequence-to-Sequence model is a model that takes a sequence of items and outputs another sequence of items. In this particular application we have a translation from English to Ukrainian. These models are composed of two parts: encoder and decoder. Encoder processes each item in the input sequence and produces a vector called a context. Decoder, on the other hand, takes context vector as an input and produces output sequence item by item.

It is important to mention that both encoder and decoder are usually Recurrent Neural Networks (RNN). Since the encoder and decoder are both RNNs, at each time step either encoder or decoder updates its hidden state based on its inputs and previous inputs it has seen. As a result, the last hidden state of the encoder becomes context vector that decoder uses.

The most obvious drawback of this method is that the longer the input sequence is, the more information can be vanished when the context vector is produced. While the encoder processes the input sequence word by word, at each time step it produces a hidden state that is the summarization of the previous hidden states.

Another problem with this approach is that while the decoder produces output sequence it cannot focus on parts of the input sequence that are relevant for producing current output.

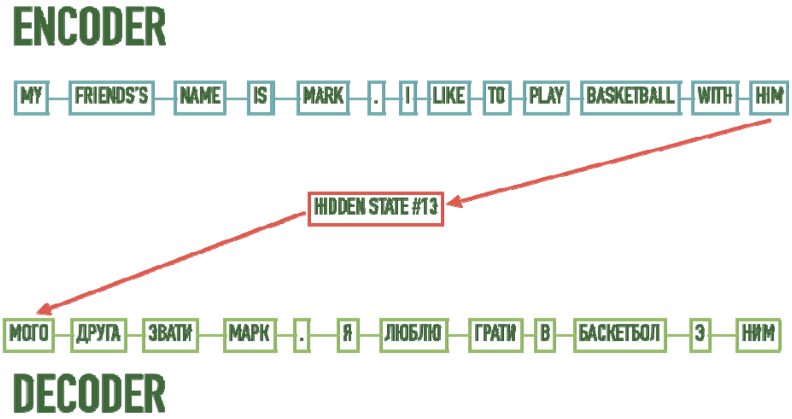

Fig.1. Standard Encoder-Decoder architecture

In this scenario we can clearly see how the attention can help us. Rather than working with final hidden state of the encoder, the encoder passes to the decoder all the hidden states that were produced by processing input sequence. As a consequence, the decoder does an extra step before producing its output. In order to focus on the parts of the input sequence relevant to this decoding time step, the decoder does the following.

As each encoder's hidden state is mostly associated with a certain word, a score is assigned to each hidden state to show how important the hidden state is for producing output at the current decoder time step. After that, each hidden state is multiplied by its softmaxed score to amplify hidden states with high scores and draw out hidden states with low scores.

Finally, we obtain a new context vector by summing hidden states vectors from the previous step. Decoder hidden state vector and context vector are then passed through a feed-forward neural network to obtain an output word. (Output of the decoder is discarded, but the decoder RNN produces a new hidden state).

This procedure is repeated for each time step. As a result, attention mechanism can dramatically improve the performance of sequence to sequence models.

Transformer model uses the above-mentioned attention mechanism to produce state-of-the-art results. It was first introduced in work [8] in 2017. The architecture of the model consists of six encoders and six decoders stacked together. The encoders are all identical in structure and have two components: Self-Attention followed by Feed-Forward Neural Network. The encoder's inputs first flow through a self-attention layer – a layer that helps the encoder to look at other word in the input sequence as it encodes a specific word. The decoder has both of these layers, but between them an attention layer lies that helps the decoder to focus on relevant parts of the input sentence.

As the model processes each position in the input sequence, self-attention allows it to look at other positions in the input sequence for clues that can lead to a better encoding for a particular word. It is similar to how hidden states are used in RNNs to connect words that were processed before with the current word. Basically, self-attention is the method used to incorporate understanding of other relevant words into the one we are currently processing.

For example, while processing the word 'they' in the sentence:

"Kate and Mark didn't listen to their parents and rushed downstairs to open their presents. They were really excited", Kate and Mark would contribute significantly to the encoding of the word 'they' when compared to other words.

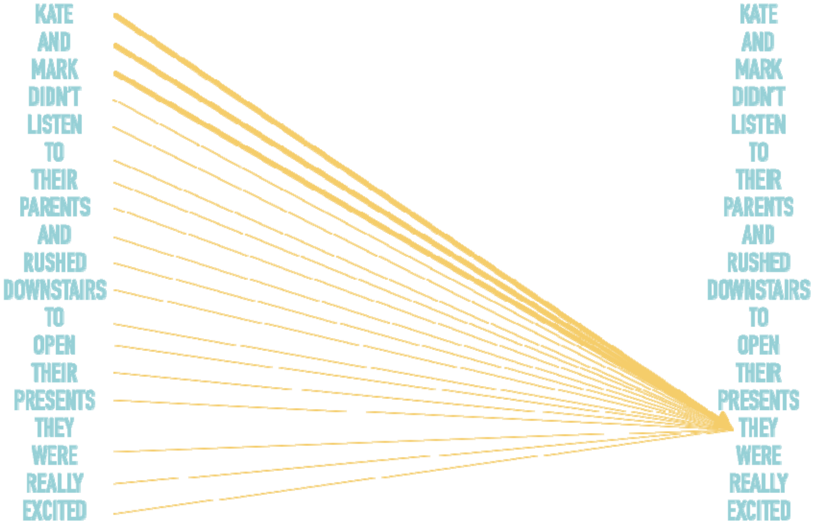

Fig.2. Example of applying self-attention

3 BERT for Ukrainian Corpus

Now let us move on to the most interesting part of this research project to see how all these things are used in real-world applications.

There have been multiple models that are based on transformer with some degree of alterations. Two of the most well-known are GPT (Generative Pre-Trained Transformer) by OpenAI ([3], [9]) and BERT by Google [5].

GPT is a language model developed by OpenAI, company founded by Elon Musk. This model can produce texts that are so well-written; they cannot be distinguished from that written by human. This model, however, only supports English.

The other model is BERT which stands for Bidirectional Encoder Representations from Transformers. BERT is a method of pre-training language representations, meaning that we train a general understanding of the language on a large text corpus to use that model for downstream NLP tasks. BERT is conceptually simple and empirically powerful. It was built with contextual representation in mind, so the word 'bank' would have different representations in 'bank deposit' and 'river bank'. The code and pre-trained models are available at [6].

The model training was based on two things: Masked Language Model and Next Sentence Prediction. In the context of the masked language model we mask a word from the input sequence and try to infer the missing word that suits the blank best. This means that the model looks at both right and left context surrounding the missing word to give a suitable inference. For example, in the sentence:

"the little ___ *played with a red car that his father bought him for his birthday"*

the model places the word "boy" instead of the blank. Next sentence prediction task again tries to understand the underlying meaning of two sentences to compute probability that one sentence can be continuation of another. For example, if we take

"It was really cold outside"

as our first sentence, the model gives 93% probability for

"That's why they stayed at home and watched Netflix"

and 15% probability for

"The Earth is the third planet from the Sun"

as being the continuation for the first sentence.

BERT is particularly interesting when compared to other state-of-the-art models because it was published in English and multilingual versions. Both models were trained on corpus derived from Wikipedia. However, since Ukrainian is a low-resource language, Ukrainian Wikipedia does not reflect the nature of the language which raises a fair question regarding the performance of the model and how easy it would be to improve upon existing pre-training. The initial guess was that although BERT is claiming to be multilingual, it was not performing well on low-resource languages like Ukrainian. The assumption proved itself to be true as you will be able to see later.

One of the biggest challenges that were faced in this project was to find a suitable dataset. Whereas Ukrainian corpuses are not widespread, it was necessary to create one. Since generating a general-purpose corpus would take an unreasonable amount of time, it was decided to focus on specific part of language and use it for the research, in particular Children's literature, including fairytales, novels and short stories.

This segment of Ukrainian language is a perfect candidate to train the model on since the child's vocabulary is not as big as that of an adult and is not as developed as that of an adult. What is more, children's literature usually follows similar patterns in its narration structure which allows model to better understand the underlying meaning and produce better results. However, literature for children still possesses a significant part of the language context, so the model might do a much better job training on it.

For training the model, we hand-picked 742 texts ranging in size, from short novels to chapters from Ukrainian classic books. Each text was preprocessed, cleaned and split into sentences for the later use. Moreover, each sentence was analyzed on specific delimiters it used. All of the delimiters were standardized and similar ones were replaced by one universal representation. This was an important step as it allowed the model to exclude delimiter-specific bias enforced by statistical occurrence of this delimiters in specific contexts. In other words, it allowed the model to focus more on the actual meaning of the sentences rather than picking up on patterns involving delimiters use. All in all, more than 93 000 sentences in Ukrainian were prepared to be used in training phase.

The project focuses on altering stories based on Masked Language Model. The input for this part is comprised of stories for kids. Each story is broken down into sentences and in each sentence one word is picked by random and is masked by [MASK] token. It is important to note that in reality we might have a compound word that, if masked, would be masked by multiple [MASK] tokens, but for the sake of the project each word, regardless of the size, was masked by one [MASK] token.

The idea is to see what the original BERT might place in the place of the masked token and compare the results to the trained version to see which one understands Ukrainian language better.

As suggested by the authors in the original paper [5], 2-4 is the optimal number of epochs to use for training BERT. Each of three possible suggestion were tested with different learning rates, however, only one set of hypermarameters turned out to be the most effective. Specifically, training the model on 4 epochs with the learning rate of 0.00005 caused the loss to decrease from 5.45397 to 1.66947, while other combinations of hypermarameters inevitably caused the loss to increase. This decrease can be seen as a good improvement considering the size of the dataset and the number of weights to be adjusted in the model. Google Colaboratory with 1 free GPU was used for training and took 8 hours for 4 epochs.

Below are the sentences with masked words and predictions based on original and trained models. All sentences considered we give first in Ukrainian and straightaway their translation in English. Again, it might be the case that the word would be masked by multiple tokens as in the original implementation, but for the sake of our project any word, regardless of size, was masked by exactly one [MASK] token to see how two models would perform.

From the inference we can see some interesting results that need to be discussed. First of all, unfortunately, original BERT does a poor job on masked words. For example, in sentence

'[CLS] Раз прийшов лис до [MASK] в гості , та й тхір його гарно погостив. [SEP]' (in Ukrainian)

(*‘[CLS] Once upon a time, the fox came to visit [MASK], so the ferret served him well. [SEP]’* – in English),

original BERT outputs

‘[CLS] Раз прийшов лис дои в гості, та й тхір його гарно погостив. [SEP]’

(*‘[CLS] Once upon a time, the fox came toU visit, so the ferret served him well. [SEP]’*),

and in

‘[CLS] Той чоловік пригонить бички додому та й [MASK]: [SEP]’

(*‘[CLS] That man yarded the bulls home and [MASK]: [SEP]’*)

the output was

‘[CLS] Той чоловік пригонить бички додому та йо: [SEP]’

(*‘[CLS] That man ran the bulls home andO: [SEP]’*).

As we can see, original BERT not only fails to predict correct part of speech, but even fails to make predictions in Ukrainian like in the first sentence. This is a clear proof of the fact that the corpus that was used to train Multilingual Bert for Ukrainian does not generalize language well enough. This can be attributed to the fact that Ukrainian is a low-resource language and it is hard to create a good corpus. It is also possible that due to the fact that multilingual BERT was trained on numerous languages, it would take a lot of time to gather a good corpus for each language. This gives an opportunity for researchers from different countries to improve BERT as it is easier for them to gather corpus specific to their region.

On the contrary, trained BERT gives promising results. In the same sentences:

‘[CLS] Раз прийшов лис до [MASK] в гості, та й тхір його гарно погостив. [SEP]’

(*‘[CLS] Once upon a time, the fox came to visit [MASK], so the ferret served him well. [SEP]’*),

and

‘[CLS] Той чоловік пригонить бички додому та й [MASK]: [SEP]’

(*‘[CLS] That man yarded the bulls home and [MASK]: [SEP]’*)

trained model outputs:

‘[CLS] Раз прийшов лис до нього в гості, та й тхір його гарно погостив. [SEP]’

(‘*[CLS] Once upon a time, the fox came to visit him, so the ferret served him well. [SEP]’*),

and

‘[CLS] Той чоловік пригонить бички додому та й каже: [SEP]’

(*‘[CLS] That man yarded the bulls home and said: [SEP]’*).

These are remarkable results considering the fact that the original word was masked with only one MASK token. As we can see, our model not only identifies the correct part of speech, but also outputs pronoun in the right gender (in Ukrainian). This is due to the fact that BERT uses attention mechanism to infer context from surrounding words and thus, as a result, understands the context better and can make pretty good predictions.

The results clearly suggest that even with a small corpus of more than 700 documents, BERT was able to learn the underlying meanings in sentences and make better predictions in terms of Masked Language Model. For example, in the first sentence

'[CLS] Жили собі [MASK] і баба.[SEP]'

(*'[CLS] There were a [MASK] and a grandmother[SEP]'*)

we can see that our model places a word 'мати' ('mother') instead of the blank, while the original model tries to put a comma. It should be noted, that here our model put wrong word ('mother' instead of 'grandfather'), but it is the correct part of speech and theoretically could be possible.

However, even trained BERT sometimes fails to understand the context. It should be noticed that the model might place punctuation marks like ",", or "." and "–" in place of the actual word. This result is not surprising since the corpus that the model was trained on consisted of many dialogues with those particular punctuation marks.

4 Possible Applications

Models based on Natural Language Processing are widely used for different applications in the modern world. Some of the most common use-cases are:

- Named Entity Recognition
- Sentiment Analysis
- Topic Modeling
- Fake News Detection
- Machine Translation
- Question Answering
- Natural Language Generation
- Information Extraction

and many others.

In terms of possible applications for the above-mentioned model, it can be used for different purposes. In terms of Named Entity Recognition, the model can be used to identify different categories like names, organizations, locations, time expressions, monetary values, percentages, etc. Moreover, the model can be used in classrooms during language classes. It can be incorporated into special educational software that would help young children learn language. The result of this research can also be used for question answering. One of the most common applications of Language Models is to understand the factual information in the given text and based on that find the answer to question. However, for this approach the training of the model should be slightly changed.

Considering the fact that BERT has no alternatives on Ukrainian market, it can become a pretty powerful tool in numerous government-related applications. Machine Learning and especially Natural Language Processing model in Ukrainian can modernize the overall day-to-day tasks.

In particular, similar models can be modified for some government purposes providing potential ways to help solving vital problems. For instance, NLP models are widely used across the globe to identify terrorists and prevent their attacks. If trained correctly, this model can recognize the underlying intent of the message written in Instagram, Twitter and especially Facebook. Police can adopt this model to scan the internet for potential threats and for eliminating them before they caused any damage. For another thing, the model can become quite useful in numerous predictions performed by National Bank of Ukraine. It has been established that raw numbers do not necessarily tell the whole story. Nation's mood, reaction to implementing different monetary policies can provide a very valuable input that can dramatically improve the forecast of inflation, GDP and UAJ exchange rate.

PyTorch-Transformers library can be used for training multiple language models, including BERT, for different purposes. The library supports the following models:

- BertModel – raw fully pre-trained transformer;
- BertForMaskedLM – BERT Transformer with pre-trained masked language modelling head on top;
- BertForNextSentencePrediction – BERT Transformer with pre-trained next sentence prediction classifier on top;
- BertForPretraining – BERT Transformer with masked language modelling head and next sentence prediction classifier on top;
- BertForSequenceClassification – BERT Transformer with a sequence classification head on top (sequence classification head is only initialized and has to be trained);
- BertForTokenClassification – BERT Transformer with a token classification head on top (token classification head is only initialized and has to be trained);
- BertForQuestionAnswering – BERT Transformer for answering questions from the text.

As can be seen from the list, the model is offered in multiple pre-trained states for different NLP-tasks. The model can potentially be used for text summarization, which can be quite useful nowadays. By applying the self-attention mechanism, more contextual information can be extracted and higher scores can be assigned to those parts of the text that possess the most important information.

5 Conclusions

Even though the model shows a clear improvement compared to the original version, it definitely leaves room for improvement and future analysis. One of the possible ways to improve the model would be to increase the dataset size. For example, not only novels for children but also for adults can be added.

Another option would be to experiment with different topics to see which one is trained better than others. What is more, all the punctuation can be removed from the texts on the cleaning stage since the model often produced a dot if the masked word was close towards the end of the sentence.

Moreover, a goal of this research is to attract research community to Ukrainian language in NLP-specific applications. As for now, there is no well-established benchmark to estimate the performance of Ukrainian Language Models. With this paper, the authors hope to make the first step towards including Ukrainian into modern NLP research. Authors would like to continue their work and establish a more robust, reliable approach to estimate the performance of Ukrainian Language models.

As a conclusion, this research project shows that even with little resources, the performance of the model that was trained on Wikipedia text can be improved to later be used for other downstream tasks like classification, question-answering and reading comprehension.

It is also important to mention that, despite being trained on Ukrainian language, the same approach could be used to train similar models for other Slavic languages. This is not surprising as languages of this group share pretty similar grammatical and semantical structures. Nearly all European countries have their own form of folklore, both with the features of the modern language and the older one.

References

1. Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. In: Advances in Neural Information Processing Systems (NIPS) 28, Conference Proceedings, pp. 3079–3087 (2015).
2. Peters, M., Neumann, M., Zettlemoyer, L., Yih, W.: Dissecting contextual word embeddings: Architecture and representation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1499–1509 (2018).
3. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. Improving language understanding with unsupervised learning. Technical report, OpenAI (2018).
4. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 328–339, Melbourne, Australia, July 15 - 20, (2018).
5. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186, Minneapolis, Minnesota, (2019).
6. Google-research/BERT, https://github.com/google-research/bert.
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: Advances in Neural Information Processing Systems (NIPS) 25, vol. 2, Conference Proceedings, pp. 1097-1105, (2012).
8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I. Attention Is All You Need, In: Advances in Neural Information Processing Systems. Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA (2017).
9. Better Language Models and Their Implications. https://openai.com/blog/better-language-models/ (2019).