Efficient Fine-tuning of SlovakBERT with Epinet

Jozef Kubík[1], Daniel Kyselica[1] and Martin Takáč[1]

[1]Faculty of Mathematics, Physics and Informatics, Comenius University, Bratislava, Slovakia

Abstract
The popularity of creating artificial intelligence models has been incredibly rising. These tools can be used in many different areas, including text analysis. Most modern models offer great accuracy in many different text-based tasks but are limited by a huge number of data required to not only pre-train but also fine-tune these models. This problem only deepens in models trained on data from low and mid-resource languages, such as Slovak. In this paper, we examined the fine-tuning process of such models and tried to enhance it by connecting to Epinet to create Epistemic neural network, a relatively new concept that helps the model to detect its own uncertainty to make better decisions in the long run.

1. Introduction

Text data can be analyzed in many different ways. The most popular approach is to pre-train the so-called Large language models. Some of the most popular are based on original BERT [1] architecture, e.g. RoBERTa [2], ALBERT [3], DeBERTa [4] and others. Most of these models are monolingual. i.e. they are pre-trained on the corpus of only one language. Although it is not necessary, every language would potentially benefit from having its own monolingual Large language model pre-trained (and consequently fine-tuned) on available data in this exact language.

Not all languages are equal in this aspect, though. For example, the most popular languages, such as English, are easier to use with these kinds of models as the data available in this language is abundant. That's one of the main reasons why almost every new model is first trained in such high-resource language. Subsequently, models can achieve better results as they can be trained longer and better on a variety of different topics, each possibly enhancing the models' language understanding potential. That leaves low-resource languages in a tough spot - either to use multilingual models, pre-trained on text corpus consisting of different languages, or pre-train their own models with a high probability of under-training them. Under-training a model can lead to worse results and thus lowers its usefulness.

SlovakBERT [5], based on RoBERTa architecture, is the first and only strictly monolingual Large language model trained on Slovak text data. It achieved state-of-the-art results in several Natural language processing tasks, such as Sentiment analysis, Document classification, and others. Compared to other models pre-trained in English, its results in many tasks (including the ones with state-of-the-art results) lacked behind, though. The most probable reason for this is the lack of data in its pre-training phase, where compared to the original RoBERTa, the amount of data used was nearly 10 times less. This undertraining thus reflects in fine-tuning phase, where the highly limited amount of annotated data often doesn't allow the model to adapt to the task as well as in other models pre-trained in other languages.

To mitigate this problem, it would be beneficial to somehow help the model in the fine-tuning phase. This could be done in several ways, one of which is to let the model know more about its own uncertainty. One of the most promising concepts working with uncertainty estimation via estimating joint predictions was recently introduced in Epistemic neural networks [6]. The authors managed to lower the joint logarithmic loss of the model by connecting it to the so-called 'Epinet' network. This network uses another input, the epistemic index, to help the model differ between aleatoric (relating to chance) and epistemic (relating to knowledge) uncertainty. In experiments, network ResNet connected to Epinet reached substantially lower values of joint logarithmic loss.

In this paper, we present several experiments of connecting the SlovakBERT language model to the Epinet network to create an Epistemic neural network suited for text classification. Although in original experiments with the ResNet network, the classification error did not lower, possibly under-trained low-resource language models such as SlovakBERT could theoretically improve also in accuracy of classification when given external help to estimate uncertainty. Improving joint log-loss values could also indicate possible better decision-making in the long run. We present the results of the Epinet + SlovakBERT network with different hyperparameters and conclude its possible usefulness in this combination.

ITAT'23: Information technologies – Applications and Theory, September 22–26, 2023, Tatranské Matliare, Slovakia

jozef.kubik@fmph.uniba.sk (J. Kubík);
daniel.kyselica@fmph.uniba.sk (D. Kyselica);
martin.takac@fmph.uniba.sk (M. Takáč)

2. Networks details

2.1. SlovakBERT

SlovakBERT is the first monolingual Large language model (LLM) trained on Slovak text data. It was introduced in the year 2021, although many different BERT-line of models were known for a longer period of time. As it is in fact RoBERTa model, its strength comes from the clever design of Transformer architecture [7], more concretely its encoder part. The attention mechanism used in this architecture not only provides great means of text understanding but also allows for partial process parallelization, accelerating the training process. The training process, which can be divided into pre-training and fine-tuning parts, allows for flexibility not only in a variety of possible tasks but also in the language of input data. The size of these models, measured in hundreds of millions of parameters, was higher when compared to older architectures, but is a little compared to other models, such as GPT [8] models with the number of parameters counting in hundreds of billions. Although the size of data used for pre-training seems reasonably big (almost 20GB), it is little compared to the original RoBERTa (160GB), which is the base architecture for this model.

2.2. Epistemic neural network

Epistemic neural network (ENN) differs from the traditional network by taking additional input called epistemic index. This index, usually sampled by some reference distribution such as normal distribution or a finite set of values, is used to help the model recognize its own epistemic uncertainty. This uncertainty is easier to show in the joint probability distribution, where multiple samples (inputs to the network) are considered.

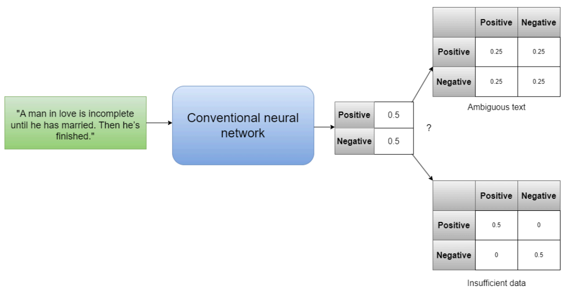

Figure 1: Example of resulting probability distributions for text classification with conventional neural networks [6].

In figure 1, for the text classification task, the conventional neural network takes as an input a text sequence and outputs a vector containing probability distribution for 2 classes - positive or negative sentiment. In an edge case (the table in the middle), the probabilities assigned are the same - 0.5. This shows the model's uncertainty for the prediction, but the reason for this uncertainty is not clear. If we took the text sequence and put it as an input for the neural network a second time, we can model joint probability distribution by the table of size 2x2. In this case, we can get two different edge cases (the tables on the right). In the upper table, all possible combinations of classes when given two input samples show the same probability of 0.25. Summing up for marginal prediction, for each sample network assigns a probability of 0.5. This case suggests aleatoric uncertainty which most probably won't be solved with future training. The more interesting case is shown in the table at the bottom right. Here, the networks' choice is always consistent - when the class for the first sample is chosen, the same class is assigned for the same sample for the second time. The only problem is that the network is not sure which class to assign in the first place. The sum of the marginal predictions is, as in the first case, 0.5, but the difference in distribution is visible. This second example suggests epistemic uncertainty, i.e. more training could resolve models' uncertainty.

The epistemic index comes as an additional input to ENN to help with epistemic uncertainty. In figure 2, we see two different use cases. In the first case, ENN doesn't use the epistemic index to its benefit and thus indicates ambiguous data. The possible use of the epistemic index is shown in the tables below it. Here, according to the sign of epistemic index, the class is chosen - that holds true not only for prediction of one input but also when the same input is given two times in a row. If a prediction of the network can be manipulated in this way by some additional input, then we can assume that future training will resolve some of its uncertainty.

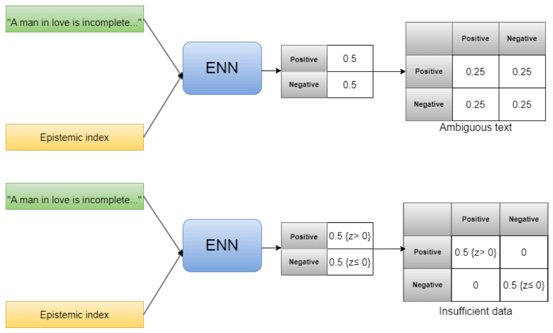

Figure 2: Example of resulting probability distributions for text classification with epistemic neural networks [6].

One concrete use of ENN was introduced with the Epinet [6] network. Visualized in figure 3, Epinet consists of two networks: learnable and prior. Both of the networks use the aforementioned epistemic index.

In most cases, these networks are standard Multi-layer

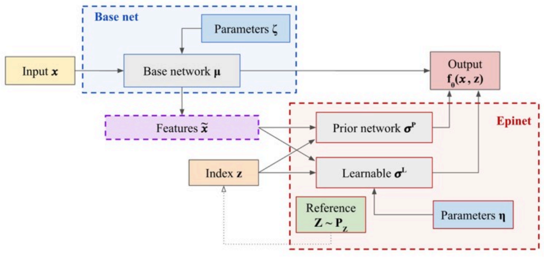

Figure 3: Visualization of the Epinet network [6].

perceptrons with the same architecture but different starting parameters. The prior network has no trainable parameters and aims to introduce initial knowledge about uncertainty. Over time, learnable networks' parameters are updated, whereas prior networks' parameters stay the same. Both of these networks take two inputs: epistemic index z and so-called features $\tilde{x}$ of the base network. These features are usually the last layers or (parts of them) of the base network, which we are trying to train. These two inputs are concatenated and fed into the learnable and prior network, which results are then summed. The final result of ENN f is the sum of the Epinet networks σ result and result of the base network μ:

$$\sigma(\tilde{x}, z) = \sigma^P(\tilde{x}, z) + \sigma^L(\tilde{x}, z)$$

$$f(x, z) = \mu(x) + \sigma(\tilde{x}, z)$$

3. Experiments

In our experiments, we use the base version of the pre-trained SlovakBERT network. On top of it, we add a Multi-layered perceptron (MLP) with one hidden layer of size 50. When connecting to Epinet to create an Epistemic neural network, the result of the SlovakBERT network present in the representation of [CLS] token at the last layer is used as the input to this MLP network and also as 'features' for the Epinet.

In Epinet, we follow the original implementation in [6]. We use two MLP networks - the learnable one is initialized with parameters via Glorot initialization [9] and takes arguments, which are concatenated together:

$$\sigma^L(\tilde{x}, z) = MLP(CONCAT(\tilde{x}, z))$$

The prior network has no trainable parameters. Its design is the same as for the learnable network:

$$\sigma^P(\tilde{x}, z) = MLP(CONCAT(\tilde{x}, z))$$

Both of these networks are the same in size as the MLP network connected to the SlovakBERT. The epistemic index is sampled from a uniform distribution.

For the size of epistemic indices, we iterate through multiple possible values. In our experiments, we try to sample this vector with sizes 10, 30, 50, and 100. The number of indices sampled in each stochastic gradient descent step is always 5. With each size we fine-tune the model from scratch, giving us the best setting for a reasonable comparison.

For optimization we use Adam [10] optimizer with starting learning rate $\alpha = 5 \cdot 10^{-5}$.

We fine-tune our model for sentiment analysis classification task on CSFD [11] dataset, a publicly available collection of movie reviews. For this, we preprocess the dataset in two ways:

- we keep only two data features: text review and rating
- as the rating column contains values ranging from 0 to 6, we edit these values to get sentiment analysis task of 3 possible classes: 0 - negative, 1 - neutral, and 2 - positive. We do this by mapping original values to predefined classes - values 0 and 1 are mapped to 1, values 2 and 3 are mapped to 1, and values 4 and 5 are mapped to 2.

No other changes to the text data itself were made. We fine-tune the model for 5 epochs on 25k train data and validate the model after every epoch on 5k validation data. The loss function used is the standard cross-entropy function. For determining joint logarithmic loss value during validation we use the vanilla version of the dyadic sampling heuristic as described in [6]. In this method, we first sample k independent random samples. After that, we sample another $\tau - k$ samples with equal probability from these two points to create a batch of size τ which we use for calculating joint loss. This heuristic is a good alternative to classic calculation as in high-dimensional space the size of the batch required to distinguish joint prediction would be too large to compute in a reasonable time.

We recognize that other settings could be potentially better suited for our experiments, but this exact preprocessing allowed us to easily train the model while keeping reasonable values for the potential sentiment in the text.

4. Results

In this section, we present the results of our experiments. To easily recognize the networks, we will always refer to SlovakBERT connected to only standard MLP as 'base SlovakBERT' and SlovakBERT connected not only to standard MLP but also to Epinet network as 'SlovakBERT Epinet' or simply 'Epinet'. Each different SlovakBERT

Epinet will also contain in its name the size of the epistemic index used.

Training accuracy nor training loss did not improve significantly when using SlovakBERT Epinet. The resulting values did not differ in a major way, which holds not only when comparing to base SlovakBERT, but also different index size parameters of SlovakBERT Epinet. As calculated training loss is in its nature marginal (each sample is independent), it seems logical that no major improvements are observed. Results can be seen in figure 4 and 5. Similarly as in [6], the difference in results between the base networks with Epinet and only the base network is negligible. Nevertheless, after training for only 5 epochs on a small dataset, training accuracy seems to reach great results around 90%.

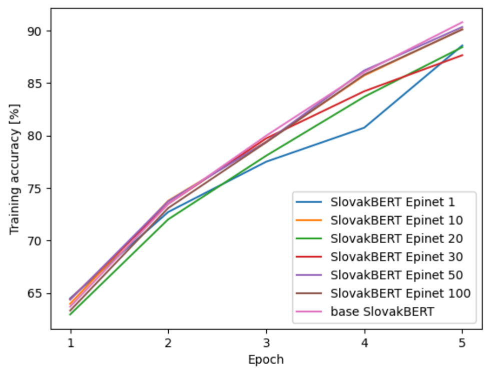

Figure 4: Training accuracy of SlovakBERT models.

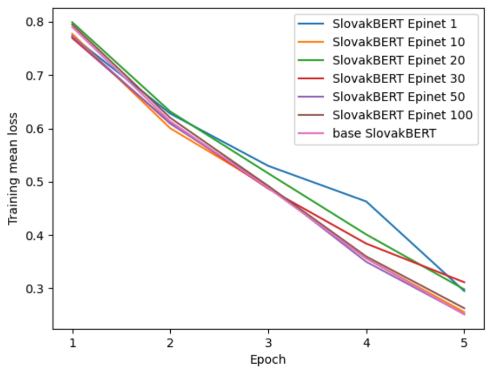

Figure 5: Training loss of SlovakBERT models.

In the case of validation accuracy, we also see no apparent improvement when using Epinet. This confirms the results of the original authors of Epinet, who also did not find any significant change in accuracy while using Epinet. Although it is logical as in the case of training accuracy, longer training could still potentially reveal the problem with under-training. On our small dataset, no major changes were discovered, though.

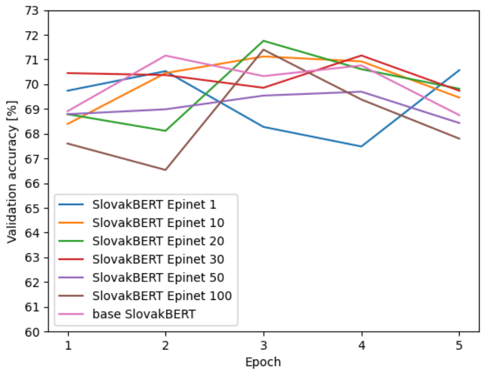

Figure 6: Validation accuracy of SlovakBERT models.

For values of final joint logarithmic loss, calculated on validation dataset, results show that the networks using epistemic index performed much better than base SlovakBERT without Epinet in average. We also found out that the networks are really sensitive to the size of the epistemic index. In the table 1 below, we present the percent value of the final joint logarithmic loss of Epinet networks when compared to the value of the base SlovakBERT.

Table 1
Difference in final joint logarithmic loss with respect to base SlovakBERT model

Network	Index size	% value
SlovakBERT Epinet	1	+30.98
SlovakBERT Epinet	10	-12.12
SlovakBERT Epinet	20	-10.77
SlovakBERT Epinet	30	+2.30
SlovakBERT Epinet	50	-17.63
SlovakBERT Epinet	100	-3.67

These results in joint logarithmic loss indicate improvement, but more experiments are needed to determine the exact improvement potential and reasons for inconsistency in results among different index sizes through training. This inconsistency can be observed when comparing the loss values of some networks during training depicted in 7. For clarity, we only show a comparison of two Epinet networks with biggest epistemic indices and the base SlovakBERT model. In general, reasonable big epistemic indices (such as index of size 50) produce better (i.e. lower) joint loss values as networks with small epistemic indices sizes such as 10. It doesn't mean that each consecutive epoch in training helps in lowering joint loss value, though. This can be also seen in the final loss values for Epinet 1 and 30, where the final value was actually higher than the final loss value of base SlovakBERT. As in [6] the authors worked with image data, is hard to make a meaningful comparison to their results, nevertheless, our results show that benefit of using Epinet expands to Large language models as well.

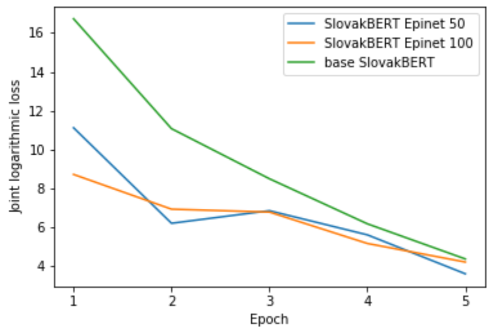

Figure 7: Joint logarithmic loss of SlovakBERT models during training.

As the results suggest, ENN can serve as an improvement to conventional neural networks, but our results indicate some sort of instability, possibly caused by a lack of fine-tuning data or weak adaptability of the network parameters when used with a low-resource language model. Both of these possible causes can be solved with additional research, though. Still, finding this instability is an interesting aspect of the ENN network, which was not presented yet. The argument that in [6] the experiments were focused mainly on image data can be made, although it doesn't seem very likely that this can be a cause for instability. A more reasonable conclusion is that the under-trained model itself creates this instability.

5. Conclusion

We showed that Large language models pre-trained in low-resource languages, such as Slovak, can benefit from modeling their uncertainty in the fine-tuning process to reach better results. This was made possible by connecting this model to the Epinet network, creating an Epistemic neural network. Results don't indicate a major improvement in classification accuracy, but joint logarithmic loss seems to improve substantially. More research and experiments on this topic are needed, though. Our next potential step is to not only enhance these kinds of LLMs in terms of raw joint log-loss values but also possibly lower the amount of annotated data needed in the fine-tuning process, e.g. with the help of Active learning methods. As most of these methods try to estimate models' uncertainty on the input data, their combination and uncertainty estimation with epistemic index and Epinet, in general, can bring interesting results when fine-tuning low-resource language models.

Acknowledgement

This research has been supported by grants APVV-21-0114 and UK/96/2023.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

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

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

[4] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, arXiv preprint arXiv:2006.03654 (2020).

[5] M. Pikuliak, Š. Grivalský, M. Konôpka, M. Blšták, M. Tamajka, V. Bachratý, M. Šimko, P. Balážik, M. Trnka, F. Uhlárik, SlovakBERT: Slovak masked language model, arXiv preprint arXiv:2109.15254 (2021).

[6] I. Osband, Z. Wen, S. M. Asghari, V. Dwaracherla, M. Ibrahimi, X. Lu, B. Van Roy, Epistemic neural networks, arXiv preprint arXiv:2107.08924 (2021).

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[8] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training (2018).

[9] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[10] D. P. Kingma, J. Ba, Adam: A method for stochas-

tic optimization, arXiv preprint arXiv:1412.6980 (2014).

[11] Fewshot-goes-multilingual, 2022. Data retrieved from Huggingface site, https://huggingface.co/datasets/fewshot-goes-multilingual/sk_csfd-movie-reviews.