The 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)
Taipei City, Taiwan, October 20-21, 2023. The Association for Computational Linguistics and Chinese Language Processing

Fine-Tuning and Evaluation of Question Generation for Slovak Language

Ondrej Megela
Deutsche Telekom
IT Solutions,
Košice, Slovakia
ondrej.megela@telekom.com

Daniel Hládek
Technical University
of Košice, Slovakia
daniel.hladek@tuke.sk

Matúš Pleva
Technical University
of Košice, Slovakia
matus.pleva@tuke.sk

Ján Staš
Technical University
of Košice,
Košice, Slovakia
jan.stas@tuke.sk

Ming-Hsiang Su
Soochow University, Taipei, Taiwan
huntfox.su@gmail.com

Yuan-Fu Liao
National Yang Ming
Chiao Tung University,
HsinChu, Taiwan
yfliao@nycu.edu.tw

Abstract

Automatic generation of questions about the given context is useful for the adaptation of question-answering systems or to support education. We trained and evaluated a model that generates a question in the Slovak language. We have designed an automatic metric where an additional question-answering model is used to evaluate the generated questions. We calculated how many questions have confidence greater than the given threshold. For generating questions, we used contexts from the Slovak question-answering dataset. The fine-tuned Slovak T5 model did generate 38% of the questions that the evaluation model could answer with confidence greater than 50%. We cooperated with partners from Taiwan during these experiments in the frame of a bilateral project and we plan to transfer the knowledge to the Chinese language later.

Keywords: evaluation, natural language generation, neural networks, question answering, question generation

1 Introduction

The idea of natural language processing (NLP) is to teach the computer to understand and respond to the user in natural language and thus prepare the user for comfortable communication. Natural language generation (NLG) refers to the process of automatically generating human-understandable text in one or more natural languages. The ability of a machine to generate text in natural language that is indistinguishable from that generated by humans is considered a prerequisite for artificial intelligence (AI).

The onset of deep learning had a great impact on this area. Indeed, not only has it advanced the state-of-the-art in existing NLG tasks but has sparked interest in solving newer tasks. NLG today includes a much wider range of tasks (Zhang et al., 2022) such as machine translation, text summarization, structured data-to-text generation, dialog generation, question answering, automatic question generation, video captioning, image description, grammar correction, or automatic source code generation.

The rapid progress of NLG in recent years can be attributed to 3 factors:

1. by developing data sets and benchmarks that allow training models (the more data the better);
2. advances in machine and deep learning algorithms have helped stabilize and accelerate large-model training;
3. availability of powerful and relatively cheaper computing infrastructure in the cloud space.

The question of how to evaluate progress becomes very important with such rapid development. Of course, the generated text can be evaluated based on grammatical correctness, however, according to which criteria to evaluate which of the generated texts is better if both are grammatically correct.

More specifically, how can it be convincingly argued that the new NLG system is better than existing state-of-the-art systems? We can let people evaluate and compare multiple outputs. The evaluation scores given by humans can be absolute or relative to existing systems. The scores provided by people provide information about which of the systems was better. However, it requires experienced annotators and specific instructions on what to pay the most attention to, which makes it time-consuming and costly. At the same time, these

assessments can be very subjective. Human evaluations can act as a serious obstacle that prevents rapid progress in this field.

This paper focuses on the problem of question generation (Lopez et al., 2021). The neural network is given a paragraph of text and is asked to generate a set of questions related to the paragraph. The generated question should be grammatically correct, comprehensible, and answerable in the given paragraph. This is a complementary task to the well-known question answering.

Our approach aims to overcome two limitations. Current question-generating methods depend on the quality of the datasets and models for the given language. To overcome this limitation, we use our own dataset of questions and answers in the Slovak language and existing general mono and multilingual models with the support of the Slovak language. The second issue is the process of the evaluation of the generated question. The existing language-independent metrics cannot distinguish between "good" and "bad" questions for the given text. Our method of evaluation uses a mono-lingual neural model, fine-tuned for question-answering.

There are two uses for question generation - education support and question-answering systems. Our research should support the creation of such a system for a lower-resourced Slovak language.

The generated questions are useful in education. With the generated question, the teacher can quickly assess how the student understood the paragraph. (Kurdi et al., 2020) provide a systematic review for educational question generation.

The second use is data augmentation for question-answering or information retrieval. The automatically generated questions for a random paragraph can enlarge the training set, or generate domain-specific questions. (Zhang et al., 2021) proposes a review of question-generation methods from the perspective of data augmentation. There are many possible commercial applications for question-answering systems, such as personal assistants, automated customer services, or medical decision support support systems.

2 Neural Networks for Language Generation

Most of the neural networks for NLG are based on a transformer (Vaswani et al., 2017). Transformer is a neural network architecture that is very widely used in the field of NLP. The main advantage it offers over recurrent neural networks is that instead of sequential processing, parallel processing is used, and a transformer can better capture word dependencies despite their distance. Parallel processing makes it possible to receive the entire input sample at once, thanks to which the power of graphic cards can be better used and thereby speed up training. The architecture of the transformer consists of two main components: encoder and decoder.

2.1 Bidirectional Autoregressive Transformer

Bidirectional AutoRegressive Transformer (BART) is a language model from Facebook developers AI (now under the name META) (Lewis et al., 2020) based on both blocks architecture transformer, i.e. both encoder and decoder. The main strategy during training was a reverse reconstruction of the text into which noise was introduced in various forms. Except for generative tasks on which it is focused, it also manages tasks such as text classification. The main idea of the developers was to expand the original BERT (Devlin et al., 2019) model by the ability to generate text and thereby add a decoder. Besides that modified the activation functions of the transformer architecture from ReLU to GeLU and adjusted the size of the encoder/decoder block according to the size of the model (e.g. the smallest version has 6 layers).

Training consisted of denoising of input text, a combination of several techniques was used for this task: span masking, permutation of sentences, and document rotation. The developers tested the performance of each text noise technique separately and the results show that the most effective of these techniques is paragraph masking.

2.2 Generative Pre-trained Transformer

The Generative Pre-trained Transformer (GPT) family of models uses only a part of the decoder block from the original architecture of the transformer (Brown et al., 2020). The first pre-trained model was GPT-1 and was published in 2018. GPT-1 model was then pre-trained using a language modeling task that can be fine-tuned for a specific task where such a large amount is not available.

The pre-training step used BookCorpus, which contains more than 700 unpublished books, where the model could learn also longer contexts in the text. Regarding the architecture, GPT-1 uses 12 layered decoders, GeLU activation function, and 117 million parameters.

The second generation of the model GPT-2 was more focused on increasing the number of data and numbers parameters. The new corpus was created from the data from the Reddit site and contained 40GB of data, which was a considerable difference from the corpus used for the first generation.

Another concept was "zero-shot task transfer", which describes the model's ability to perform a task without some sample data from the task. The GPT-2 model had these abilities when longer fine-tuning was not needed, but rather showed the model a few examples of the given task, and the model could perform the given task. GPT-2 was published in 2019 and at that time he reached "state-of-the-art" levels on several tasks in "zero-shot" settings.

The third generation of models, GPT-3, continued the trend of larger models and adding corpora to training, in addition, the basis of the architecture was the same as at GPT-2. Regarding the size of the parameters, the largest of the third-generation GPT models was 175 billion of parameters (again, a significant increase). GPT-3 is capable of creating text that seems very human and that is why the developers decided not to publish him, but instead offer interested parties a paid API through which they will be able to use the given model. Further progress continues in the form of GPT-3.5, on which the well-known chatGPT was based, and the fourth generation of GPT (GPT-4).

2.3 Text-to-Text Transfer Transformer

The Text-to-Text Transfer Transformer (T5) model comes from Google developers, who worked with the idea of transferring knowledge of models (English transfer learning) (Raffel et al., 2020).

It uses pre-training on large unlabeled textual data but the idea was extended to include tasks for which the models are fine-tuned together and are related; therefore it should not be necessary to have a different model for each task. This thought translated into practice by transforming each problem into a text-to-text task, which means that in addition to the fact that the input is text, its output is also in the text form that the model generated.

The model can be used for several tasks such as text classification, text summarization, or machine translation. This is possible thanks to the addition of a prefix, which defines what task the model has to perform. T5 is a model in which they use the entire architecture of the transformer (both encoder and decoder) unlike the models like BERT or GPT.

In addition to these versions, a multilingual version of the model called mT5 was also created (Xue et al., 2021). The same authors created training corpus mC4. This corpus is similar to C4 corpus (Colossal Clean Crawled Corpus) in (Raffel et al., 2020), but contains text in 101 languages (including Slovak). mT5 was not trained using other corpora for specific tasks (SQuAD (Rajpurkar et al., 2018), SNLI (Bowman et al., 2015), etc.), that is, to use the model for one task as it is not necessary to add a prefix for fine-tuning. Adding so many languages made an impact also on the number of parameters of the model and, like the T5, it came in different sizes.

2.4 Slovak T5

The Slovak version of the T5 model (Cepka, 2022) is also available, which was created by further training of mT5 (Xue et al., 2021) on the Slovak version of the mC4 dataset. The author extracted Slovak parts from the original mC4 (Xue et al., 2021) and the OSCAR (Abadji et al., 2022) datasets. It is further fine-tuned on multiple machine-translated particular tasks.

For the model evaluation, three related tasks were used:

- SST2-sk – the text sentiment analysis task (Socher et al., 2013).
- STSB-sk – comparison of the similarity of two inputs (Cer et al., 2017).
- BoolQ-sk – answering the yes/no questions from the texts (Clark et al., 2019).

3 Evaluation of Natural Language Generation

The goal of this paper is to create and evaluate a system for question generation, which is a part of the NLG. In this section, we will focus on the metrics used for artificially generated text. As mentioned above, the evaluation of the generative model using an automatic metric is not at all a trivial task, since natural language offers a lot of variability so it is difficult to design the expected output.

An overview of NLG metrics is presented in paper (Sai et al., 2022), but we will focus only on the most popular ones. These can be divided into two categories (Nema and Khapra, 2018):

- metrics based on word overlap – they usually compare words or a sequence of words

between the target (required) and generated (artificial) by text;

- metrics based on the use of pre-trained models - they use pre-trained models to create a vector representation of texts and then the similarity of the texts is calculated.

3.1 Bilingual Evaluation Understudy

The Bilingual Evaluation Understudy (BLEU) score is a metric originally designed for machine translation but can be applied to multiple NLG tasks (Papineni et al., 2002). For the use you need to have:

- candidate sentence – generated artificial sentence or sequence of words;
- reference sentences – one or more reference sentences that represent the expected output of the generative model.

This metric evaluates the generated text based on similarity with reference text. There are several studies that show that BLEU and similar metrics do not correlate well with human evaluation and yet there has been no decline in their popularity.

3.2 Recall-Oriented Understudy for Gisting Evaluation

The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is a metric that was designed for text summarization evaluation (Lin, 2004). Similar to BLEU, it uses overlapping n-grams or a longer sequence of words between reference and candidate texts. The most famous versions of this metric are ROUGE-N, ROUGE-L, ROUGE-S, and so on.

ROUGE-N represents a recall-oriented metric that works very well similar to BLEU. Also, n-grams are used, N in the name describes the length of the n-gram (ROUGE-1 for unigrams, ROUGE-2 for bigrams, etc.). The numerator represents the maximum number of n-grams generated or candidate summarizations identical to the n-grams of the reference summarization. The denominator forms the sum of all n-grams of reference summarization.

Instead of n grams, ROUGE-L uses the longest common sub-sequences. Unlike ROUGE-N, the main advantage is that it is not necessary to define the length of the n-gram in advance. The result is a modification of the F-score, where precision and recall are taken into account.

ROUGE-S uses skip-bigrams that represent pairs of entry words text. Unlike bigrams, skip-bigrams do not have to represent adjacent words.

3.3 Metric for Evaluation of Translation with Explicit Ordering

The Metric for the Evaluation of Translation with Explicit Ordering (METEOR) was also created for the task of machine translation (Banerjee and Lavie, 2005). The motivation behind its development was to improve BLEU and the correlation between automatic and human scoring. Similarly to ROUGE-L, the METEOR calculates the return in addition to precision. The unigrams are used to find a match between the reference and candidate text and the mapping that forms the grouping (alignment).

Every word in the candidate text is assigned to the most one word in the reference text. In the mapping, several strategies can be used; the simplest is a direct match, where only the identical words are mapped, in the same form and time. Other options use stemming, with the help of which it would be possible to map words with the same vocabulary basis or to use the semantic similarity of words when it would be possible to map synonyms.

3.4 BERTscore

BERTscore can be classified into the category of metrics using pre-trained language models (Zhang et al., 2020). As can be deduced from the name, this is the model used precisely by BERT (Devlin et al., 2019), which is not included among the generative models; rather, it can be included in the understanding of natural language, since its task is to create a vector for each word of the sentence. So it is at the beginning for each word of both the candidate and reference sentences, a vector representation is calculated. When these vectors are created, pairs are created between the reference and candidate vector sentences based on semantic similarity, which is calculated using the cosine vector distances.

3.5 Answerability

"Answerability" is a lesser-known metric compared to previous metrics (Nema and Khapra, 2018). This is because the previous metrics could be applied to multiple tasks, however, it is designed for the question generation task. The authors recommend the usage of this metric in combination with another metric, e.g., BLEU. The ambition is to see if everything is present in the question in the necessary context to answer it. Let us imagine the

reference question r: "What is the address of the university?" and two candidate questions $q1$: "University address?" and $q2$ "What is the address?". When using the previous metrics would result in question $q2$ getting a better score, but the person does not find enough context in the question to be able to answer it. On the other hand, question $q2$ is not the best, but we dare say that most people would know.

4 The Slovak Question Answering Dataset

After choosing a question generation task, it was necessary to obtain data to be able to teach a model to perform a task. In the previous section, we covered available datasets that could be used for this task, but there are few datasets in the Slovak language. For this reason, we decided to use a dataset that represents the Slovak version of the SQuAD dataset (Rajpurkar et al., 2016, 2018).

At the end of March 2023, an article about the Slovak version of the original English SQuAD dataset was published in our IEEE Access paper (Hládek et al., 2023). This dataset provides $24,630$ paragraphs from $9,317$ documents for which $91,165$ questions are created. The point was to create a corpus as similar as possible to the SQuAD v2.0 dataset including unanswerable questions. The SK-QuAD dataset consists of Slovak Wikipedia articles that were divided into smaller articles and cleaned of tables and other non-textual parts. Answer types and their share in the dataset can be seen in Tab. 1.

For editing, we created a separate Jupyter notebook, where the input Slovak dataset we first loaded. Subsequently, we extracted contexts and questions from the dataset, so that the prefix "generate questions:" was added before each context, and all questions for the given context were stored one behind the other. We also removed questions that were not answerable based on the given context. We saved the resulting modified SK-QuAD dataset separately in JSON format.

5 Model Fine-Tuning

The main aim was to train a model that would be able based on the input context (longer text) to generate questions. These questions must have been specific to the context. Jupyter notebooks were used together to develop the practical part with libraries such as HuggingFace, PyTorch, Pandas, etc. which we installed in the virtual Conda environment. The practical part was performed on the server with four NVIDIA GeForce GTX 1080 Ti graphics cards, each with 12GB of memory.

The next step after modifying the corpus was to choose a suitable type of model and find the most suitable pre-trained version. In our case, there were not many options available, after examining available Slovak pre-trained generative models, freely available in the HuggingFace library, we had two options to choose from:

- the Slovak T5 model;
- the Slovak GPT-J model.

We decided to use the Slovak T5 model precisely because of the advantage of using the prefix, which ensures that the model does not confuse the question generation task with other tasks. Before we started fine-tuning the model, a modified SK-QuAD was needed to prepare for model processing (data preprocessing). First, we loaded the model together with the tokenizer from the HuggingFace library. Subsequently, we modified the downloaded tokenizer by adding a separation token, which will be used to separate questions. We tokenized the input data. We also added a separation token at the end of the sequences (at the end of the context and the last question).

After data processing, we defined the hyperparameters:

- batch size for training – 4 samples;
- batch size for evaluation – 4 samples;
- gradient accumulation step – we set it to 16 steps (serves for defining how many gradient update steps to take before the backward or forward promotion is performed);
- learning rate - we set it to 1e-5 (how much the model weights can change at most during one step);
- number of epochs – we used 7 epochs (one epoch means one passage through the entire corpus);
- evaluation interval – we set it so that the model was evaluated every 100 iterations.

Table 1: Statistics on the SK-QuAD dataset

	SK-QuAD		SQuAD v2.0	
Number of	Train	Dev	Total	Train
Documents	8,377	940	9,317	442
Paragraphs	22,062	2,568	24,630	19,035
Questions	81,582	9,583	91,165	130,319
Answers	65,839	7,822	73,661	86,821
Unanswerable	15,877	1,784	17,661	43,498

Figure 1: Loss during fine-tuning of the Slovak T5 model

6 Model Evaluation

After fine-tuning the model, it was possible to test its functionality. We used the ”generate()” method from the HuggingFace library together with the parameters:

- max. output length – 128 tokens;
- number of beams – 20, you can decide during generation runs in a directed graph, where the nodes are possible tokens and they are rated by probability. This parameter says that the model maintains knowledge of the 20 most likely paths within the graph;
- length penalty – 0.3, set to increase the score of longer questions;
- repetition of n-grams – set to 3, i.e. in the generated text no trigram can appear more than once;
- early stopping – set so that the generation stops only when the list is of candidate sequences is equal to the number of beams;
- number of generated sequences – tells how many sequences we want to generate, set to generate 5 questions for each context.

To evaluate our model for question generation, we selected a metric similar to the BERT score. First, we fine-tuned a SlovakBERT model (Pikuliak et al., 2022) for the task of answering questions. The fine-tuning process is described in our IEEE Access paper (Hládek et al., 2023). The input of the model is a question in natural language and a paragraph of the corresponding text. The network is trained to select a text span that answers the question. The output of the network is also a number that expresses the confidence of the neural network with the found span with the answer. Confidence can be used to determine if the answer is valid.

The confidence score is calculated as a sum of probabilities of the model answer. The fine-tuned SlovakBERT model is discriminative - it selects the start and end of the span with the answer. The last layer of the model returns softmax probabilities for both the start and end of the answer span. We get a confidence score by adding these two probabilities

Table 2: The ratio of generated questions with confidence above the threshold

threshold	0.5	0.6	0.7	0.8	0.9
model sk-t5	38.01%	28.46%	19.29%	11.43%	5.2%
model mT5	43.54%	25.56%	15.14%	4.73%	4.73%

together.

We used this confidence score to measure the quality of the generated question. We assume that the question is good if it can be answered by the neural network and is bad if it cannot.

The evaluation procedure was as follows:

1. generate 5 questions for each context using the generative model;
2. use each question together with the context as input for the discriminative evaluation model;
3. from the output of the evaluation model, save each answer score and the answer itself the answer;
4. calculate the ratio of questions with scores above the threshold for all questions. We used the threshold values: 0.5, 0.6, 0.7, 0.8, and 0.9.

The results of the experiments are displayed on Tab. 2. The table shows the ratio of generated questions with confidence above the threshold for the two models. The first line marked "sk-t5" contains the results of the fine-tuned Slovak question-generating model, the second line is the multilingual question-generating model. We can see that the fine-tuned model generates questions with more confidence than the basic multilingual model.

7 Conclusion

This evaluation offers the benefit of utilizing a well-explored task of question-answering in which models can rival human performance. However, it comes with several drawbacks. The model does not consider grammatical correctness, which can lead to inappropriate answers that exceed the predetermined threshold. Moreover, the model can generate correct answers that are too difficult for the evaluation model to process.

Acknowledgment

This research was supported by the Scientific Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic and the Slovak Academy of Sciences under the project VEGA 2/0165/21 funded by the Ministry of Education, Science, Research and Sport of the Slovak Republic; and by the Slovak Research and Development Agency under the projects APVV-SK-TW-21-0002, APVV-22-0261, and APVV-22-0414.

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a cleaner document-oriented multilingual crawled corpus. In *Proceedings of the Language Resources and Evaluation Conference*, pages 4344–4355, Marseille, France. European Language Resources Association.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proc. of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates.

Richard Cepka. 2022. Slovak T5 small. Technical report, Comenius University in Bratislava.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proc. of SemEval-2017*, pages 1–14, Vancouver, Canada.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Hládek, Ján Staš, Jozef Juhár, and Tomáš Koctúr. 2023. Slovak dataset for multilingual question answering. *IEEE Access*, 11:32869–32881.

Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30:121–204.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain.

Luis Enrico Lopez, Diane Kathryn Cruz, Jan Christian Blaise Cruz, and Charibeth Cheng. 2021. Simplifying paragraph-level question generation via transformer language models. In *PRICAI 2021: Trends in Artificial Intelligence*, pages 323–334, Cham. Springer International Publishing.

Preksha Nema and Mitesh M. Khapra. 2018. Towards a better metric for evaluating question generation systems. In *Proc. of EMNLP*, Brussels, Belgium.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matúš Pikuliak, Štefan Grivalský, Martin Konôpka, Miroslav Blšták, Martin Tamajka, Viktor Bachratý, Marián Šimko, Pavol Balážik, Michal Trnka, and Filip Uhlárik. 2022. SlovakBERT: Slovak masked language model. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7156–7168, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proc. of ACL (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proc. of EMNLP*, pages 2383–2392, Austin, Texas.

Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2022. A survey of evaluation metrics used for nlg systems. *ACM Comput. Surv.*, 55(2).

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*, pages 1631–1642, Seattle, Washington, USA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems - NIPS*, volume 30. Curran Associates.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proc. of NAACL: Human Language Technologies*, pages 483–498, Online.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2022. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*.

Ruqing Zhang, Jiafeng Guo, Lu Chen, Yixing Fan, and Xueqi Cheng. 2021. A review on question generation from natural language text. *ACM Transactions on Information Systems (TOIS)*, 40(1):1–43.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.