University *of Ljubljana*
Faculty *of Computer and Information Science*

Sentence Paraphrasing

Blaž Rolih, Grega Šuštar, and Žiga Rot

Abstract

This paper presents a Transformer-based approach using T5 models for sentence paraphrasing in Slovene. By leveraging the powerful encoder-decoder architecture and fine-tuning on a backtranslated ccGigafida dataset, the proposed method generates paraphrases of sufficient quality while maintaining semantic equivalence. We experimented with a small and large sized T5 models and compared them to a baseline model which replaces all possible words in the sentence with synonyms. We evaluated our models automatically with four different evaluation metrics: ROUGE, BERTScore, ParaScore and ROUGEp. Furthermore, we had 8 different human evaluators which determined the appropriateness, fluency and syntactic diversity of the generated paraphrases.

Keywords
Paraphrasing, T5, transformers, back-translation, natural language processing

Advisors: Slavko Žitnik

Introduction

Paraphrasing is the process of expressing a given sentence or text in different words while retaining its original meaning. It is a longstanding problem for natural language learning that can be applied in machine translation, question answering, semantic parsing... Traditional methods for paraphrasing rely on rule-based techniques, which have limitations in handling complex sentence structures and capturing the nuances of meaning. To address these challenges, recent research has explored the use of neural networks for paraphrasing. In particular, the current state-of-the-art models use transformers, that are capable of capturing long-range dependencies in sentences, to generate paraphrases. In our work, we will implement and evaluate approaches to this problem using neural networks.

Related Work

To design a sentence paraphrasing system, one must consider 3 basic aspects. The training data, model selection, and evaluation metrics for the evaluation are all key components of a well functioning system.

Zhou et al. [1] overview the current state-of-the-art approaches, and thus cover most of the available datasets for this task. While there are specialised datasets for word-level paraphrase generation, sentence-level datasets are a scarce resource, and even the ones listed come with drawbacks. Quora [2] and WikiAnswer [3] databases consist of questions only, TwitterURL [4] has noisy labels due to automatic labeling and MSCOCO [5] is a computer vision database and is thus not primarily focused on paraphrases.

To tackle this problem, automatically generated databases are often used. By employing machine translation, one can

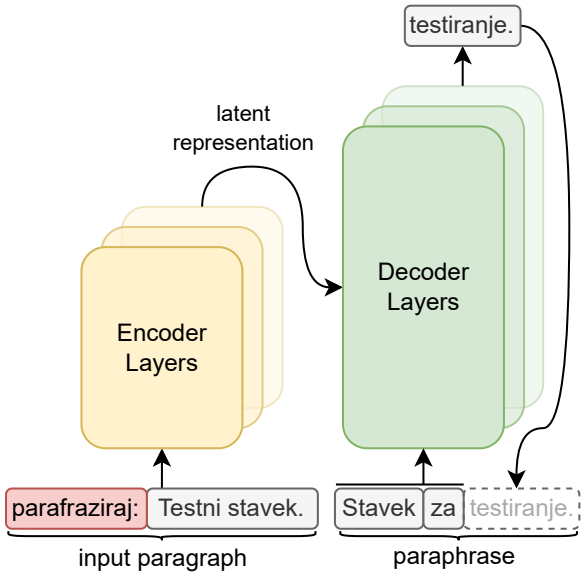

Figure 1. Paraphrase generation with T5. The prefixed input is passed into the encoder, while the decoder autoregressively generates the paraphrase.

gather only the desired sentences, and translate them into a different language and back. Federmann et al. [6] show that such artificial datasets are better than ones produced by human experts from a word diversity aspect.

For the model aspect, it comes as no surprise that deep neural models are the most commonly used. Like in other fields of natural language processing, deep learning techniques have significantly raised the performance ceiling [1]. While there seem to be no task-specific architectures, many of the commonly used models in NLP make an appearance here as well. Due to the problem's nature, most of them employ an encoder-decoder architecture. As in other NLP areas, there has been a shift from the recurrent- to the transformer-based models. Recently, fine-tuned large language models have also been proven to quickly adapt to new domains, while being highly data-efficient.

Once an appropriate architecture is selected and trained, model evaluation remains. In the context of paraphrase generation, this is not a trivial topic. Popular NLP performance metrics, like BLEU [7], ROUGE [8], and BERTScore [9], do not correctly evaluate generated paraphrases [10]. But these metrics can be used to evaluate some aspects, like lexical divergence. Since a good balance between semantic similarity and lexical divergence makes a good paraphrase, frequently used metrics, which only focus on one aspect (and can also punish the other), can not properly score generated paraphrases. Shen et al. [10] propose a new, problem-specific, metric named ParaScore. It combines scoring for both semantic similarity and lexical divergence into a single score, making it better aligned with the experts' scores. A similar idea comes from Patil et al. [11] with metric named $ROUGE_p$. Since these automatic evaluation methods are not always reliable, we can use human judgment to evaluate the results [6]. To get more information about the results, we can also use Pearson correlation of human evaluation with machine output to determine its reliability [12].

Methods

Data

Like in any project based on deep learning, first, a sufficiently large amount of training data needs to be gathered. As alluded to earlier, no large enough datasets have been presented yet. Moreover, since our goal is paraphrase generation for Slovene language, there is no other choice, but to construct the dataset artificially.

0.1 Dataset generation

For this task, we employ a similar method as described by [6]. We use neural machine translation models, provided by [13], to generate paragraph pairs via backtranslation. the input paragraphs are first translated into English, and then these translations are again translated into Slovene. Since the languages are different enough, the backtranslated sentences keep the meaning, while also being syntactically different. We provide figure 2 for easier understanding of the pipeline.

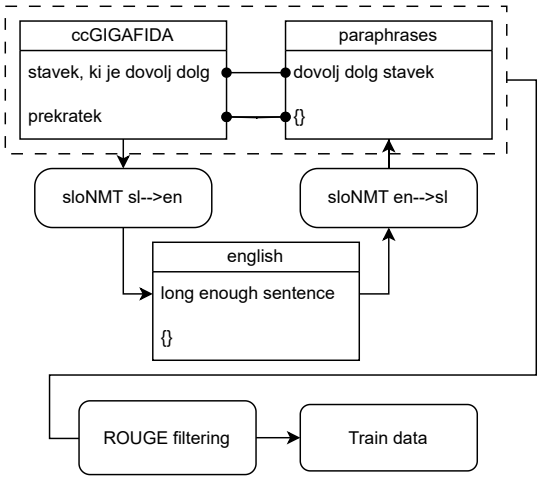

Figure 2. Dataset generation. Shown is the backtranslation and filtering procedures, we used for generating our dataset.

As a basis, we use ccGigafida [14] corpus, as it offers paragraphs, which cover many diverse topics, making our dataset as general as possible. When generating pairs, paragraphs, which are too short are automatically discarded, while the ones being too long are truncated at word-level. This is done to prevent the models overfitting on short sequences, which is usually the case. The decision to truncate by maximum length as well, stems from the models, that we are using. Since they only have a limited amount of context (512 tokens), we set the approximate maximal amount of words to ~ 200. Additionally, we remove some special characters at this stage, since we notice that the NMT models do not play well with them. We ran the backtranslation pipeline for 1 CPU week, to obtain roughly 36.000 paraphrase pairs in the train, and another 450 for the test set only.

As an additional fallback dataset, in case this strategy would not work, we also generated an additional dataset based on the Bible. Being divided into books, chapters and passages and translated into many different languages, the Bible is ideal for easy generation of corresponding paragraph-paraphrase pairs. The downside, however, is the narrow focus on religious topics, which the pairs represent. This means, that models, trained on such datasets, generate paraphrases, which closely resemble ramblings of a schizophrenic patient. We first scraped the Slovene and English passage pairs from *biblija.net* [15]. Then, similarly as with the backtranslation, the English ones are translated into Slovene language, again forming corresponding paragraph-paraphrase pairs.

Lastly, given the recent surge in freely available large language models, we also tried to generate paraphrases with them. We experimented with publicly released Llama models, as well as with the Open-AI's GPT3.5 and 4. Unfortunately, our prompt engineering did not yield satisfactory results for Slovene language. We suspect that the reason lies in small representation of the language in the training data of these large models.

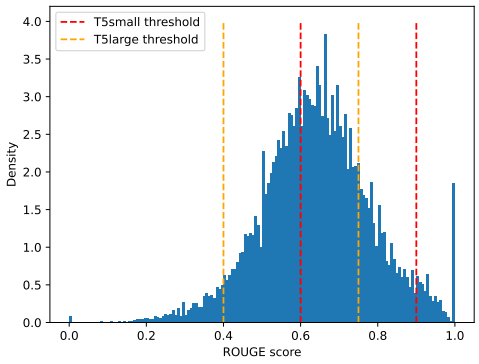

Figure 3. Dataset filtration. Shown is the ROUGE score distribution, as well as low and high thresholds for filtration.

0.2 Data filtration

When dataset generation concluded, we also employed additional clean-up. As mentioned before, we only removed some Unicode characters and short paragraphs during generation, which is not sufficient. Since Figure 3 shows the histogram of ROUGE scores over all the data. We set a pair of thresholds to filter. A low threshold to remove any nonsensical inputs, and a high threshold, to remove paraphrases, that are too close to the input. Examples, of two such pairs can be observed in table 1.

During model training, we also noticed, that small model prefers different thresholds than the large one. Therefore, we construct 2 train sets. One tailored for the small model and one for the large one. For the larger model, we use ROUGE score thresholds of 0.4 and 0.75, while the smaller also preferred sentences closer to the inputs, so we set the low and high thresholds to 0.6 and 0.9, respectively.

ROUGE	Paragraph	Paraphrase
low	raze sožalja, cvetje, sveče in svete maše, g. patru Luki za opravljen obr	Â Â
high	6. 1 L alkohola tehta 785 g pri 25 oC.	6,1 l alkohola tehta 785 g pri 25 oC.

Table 1. Examples of removed pairs. Low ROUGE score implies nonsensical data, while high shows high similarity between the input and the target.

Models

After the datasets were appropriately filtered, we decide on the model selection. Slovene being the target language, our choices in this regard were limited, as language change requires total model retraining. Since training models of such size from scratch is not feasible in our case, we resort to fine-tuning pre-trained models. Additionally, our goal can be classified sequence to sequence task, which furthermore limits our choices. Consequently, we pick small and large Slovene T5 [16], Slovene GPT, for our potential models and a simple synonym replacement model as a baseline.

0.3 Baseline

The baseline model is based on trivial synonym replacement technique. Input paragraphs are firs ran through a general-purpose NLP pipeline, which incorporates part of speech tagging for word form classification and lemmatization for base form extraction. We used classla [17] library for this task, as others lack support for Slovene language. Next, all nouns, verbs and adjectives are replaced with a random synonym from the Sopomenke v1.0 corpus [18]. As one can imagine, the quality of these paraphrases is quite poor.

0.4 Sequence to sequence models

Due to the nature of our problem, sequence to sequence models are the natural first choice for the task. Both large and small T5 models [16] employ a transformer-based encoder-decoder architecture. Essentially, the encoder part takes the original paragraph as an input, embeds it into a latent representation, which is then fed into the decoder layers. The decoder part, on the other hand, is an autoregressive conditional language model. Start-of-sentence token is first passed as an input to start the paraphrase generation. The model predicts the following token, which is then appended to the input in the next time step. This is repeated until an end-of-sentence token is predicted, at which point the generation ends. An illustration is shown on Figure 1.

A peculiarity of the T5 model family lies in the way in which they are trained. Multiple tasks are used during the training and in each one, inputs of the encoder are prepended with a specific prefix. As such, we also, prefix our input paraphrases with ”parafraziraj:”, to not deviate from the pre-training too much.

For training of both large and small models, we employ the huggingface transformers library, as it offers fast experimentation and good documentation. As mentioned before, We used the following hyperparameters during training:

- learning rate: 5×10^{-5} (linear schedule, 500 warm-up steps)
- training epochs: 15,
- batch size: 4 (small), 1 (large)
- train-val split ratio: 0.9-0.1
- checkpointing: enabled

We train both models on an HPC cluster, on a single nVidia V100 32GB GPU.

0.5 Conditional language models

As an alternative to seq2seq models, we also try to train a conditional model. GPT, like T5, is based on the transformer architecture, with the difference that it only consists of the

decoder layers. Similarly, as the authors of ROUGEp metric [11], we set up the training by concatenating the input paragraph and the target paraphrase, with a separation sequence in-between. Like suggested in [11], we also experimented with masking the loss function for the true tokens, so the model would only learn to generate the paraphrases. Unfortunately, all training runs ended with the model producing unsatisfactory results, so we omit it from any further analysis.

Evaluation methods

To ensure thorough evaluation of the models, we carefully selected suitable metrics suitable for our task. Since paraphrasing is a subjective task, determining an absolute measure of quality is challenging. Therefore, we employed four automatic metrics and complemented them with a human evaluation process to provide a comprehensive assessment.

0.6 Automatic metrics

There are many metrics used for automatic paraphrase evaluation, but the standard metrics like BLEU [7] or ROUGE [8] can't capture the entire quality of the paraphrase. For this reason, we in our work use four automatic metrices, each one providing information about the quality of a particular aspect of paraphrase. We use ROUGE to capture syntactic similarity, where optimal score is not too high. We then use BERTScore to measure semantic similarity. We then also used two metrics that were designed to capture more than one quality of our paraphrase, these are ParaScore [10] and ROUGEp [11]. We will discuss how well they align with human judgement in the results section.

0.7 Human evaluation

To have a point of comparison for our automatically computed metrics, we manually evaluated 10 paraphrases generated by each of the models. Additionally, we also evaluated the backtranslated sentences that we used in the test set to see how appropriate they were to begin with.

We first pre-evaluated some example results to harmonize our criterion among us, this way eliminating any imposter scores that looked suspicious and agreed on unified grading rules for a fair and reliable evaluation process.

Then we collected scores from 8 human evaluators. Each evaluator evaluated 40 sentences in total using the webapp, we designed that can be seen in Figure 4. For each paraphrase we evaluated 3 key metrics: Appropriateness, Fluency and Syntactic diversity. **Appropriateness** measures how much information from the original sentence was included in the paraphrase. If the paraphrased sentence conveys the same information as the original sentence, then we have the paraphrase the maximum appropriateness score. **Fluency** dictates how coherent and human-like the generate paraphrase is. If the sentence has a lot of grammatical errors or has an inappropriate structure, then we would deduct points from the fluency score. if on the other hand, the sentence was indistinguishable from a human written sentence, then we gave it a maximum fluency score. Lastly, we evaluated the **Syntactic diversity** of the paraphrase. This metric tells us how different the paraphrase is from the original sentence. If there are a lot of synonyms or if the structure of the sentence is changed (i.e. swapping the word order) then we would give it a high score. On the other hand, if the sentence is an exact copy of the original, then we would give it a score of zero.

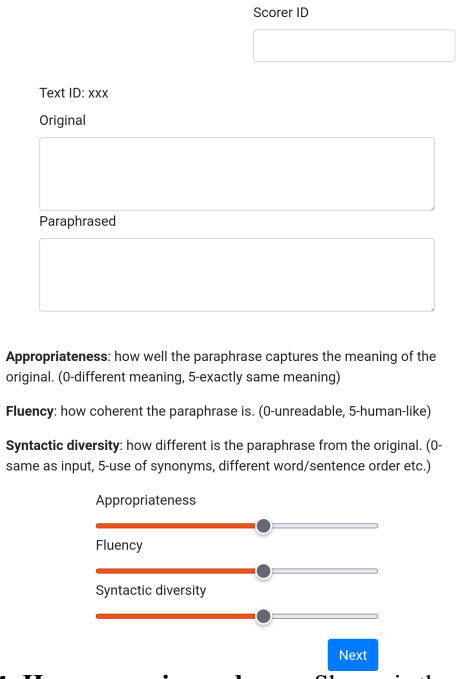

Figure 4. Human scoring web app. Shown is the interface, sliders for scoring and instructions for scoring.

Results

In this section, we will showcase our findings regarding the baseline performance, the quality of our ground truth data, and the effectiveness of the two successfully trained models.

Evaluation results

We present our results with previously described metrics, automatic as well as human, in Table 2.

To see how good our automatic metrics align with human evaluation, we can check the correlation matrix in Figure 5. We notice that ParaScore correlates very poorly, despite working well in original paper, most likely due to poor dataset selection there. We can see that ROUGEp works better, but still not sufficiently. Best results are obtained if we look at individual quality, where BERTScore aligns very well with appropriateness and fluency, and ROUGE has a very nice negative correlation with diversity.

Examples

Table 3 shows a pair of paraphrases, generated by our models. When the models perform well, they can use synonyms, change order of words and even merge sentences. But, they do not always show the same level of performance. As seen,

	ParaScore	ROUGE	ROUGEp	BERTScore	Appropriatness	Fluency	Diversity
test set	0.90 ± 0.10	0.65 ± 0.13	0.64 ± 0.10	0.83 ± 0.06	4.08 ± 1.03	4.45 ± 0.82	3.11 ± 1.10
baseline	$\mathbf{0.80 \pm 0.06}$	0.48 ± 0.11	0.42 ± 0.07	0.63 ± 0.08	1.87 ± 1.33	0.71 ± 0.94	$\mathbf{3.03 \pm 1.16}$
small	0.69 ± 0.14	$\mathbf{0.88 \pm 0.07}$	$\mathbf{0.74 \pm 0.21}$	$\mathbf{0.93 \pm 0.04}$	$\mathbf{4.37 \pm 0.83}$	$\mathbf{4.47 \pm 0.99}$	1.11 ± 0.91
large	0.77 ± 0.16	0.83 ± 0.10	0.72 ± 0.18	0.90 ± 0.05	4.35 ± 0.74	$\mathbf{4.47 \pm 0.82}$	1.51 ± 1.03

Table 2. Results from human and automatic evaluation. We can see that our trained models work better than the baseline in all aspects, except for ParaScore and diversity. That is a side product of our training set, that produced too many hallucinations if it was too diverse.

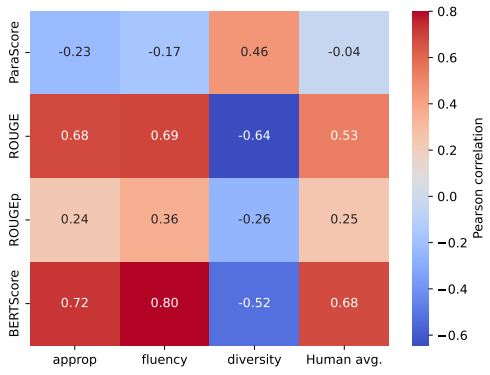

Figure 5. Correlation plot. Shown are the correlation coefficients between the human and automatic metrics. We notice that automatic metric align with specific qualities but are unable to capture entire quality.

the models can just copy input and even remove crucial information.

Discussion

To start a list of future improvements, firstly, the dataset could be manually filtered, as we noticed english input sentences in the filtered datasets. These types of errors have not been accounted for in our methodology, meaning that likely even more errors are present in the dataset.

Secondly, given more time, model training should also be reinspected. As it stands, we do not know, whether just training for more epochs would improve the results, as the models seemed to improve even at the end of training. So training hyperparameters, like learning-rate, should be reevaluated to possibly save time and enable faster experimentation. And a more thorough logging strategy should be employed to better track the training runs of the models.

Lastly, at the evaluation stage, more manual annotation could be done. Currently, only a small amount of generated paraphrases have been evaluated, just to necessitate the comparison with the automatic scores. Due to low number of paraphrase samples, our conclusions might be incorrect. Larger amount of manually scored paraphrases could also pave the way to better model selection, making it easier to automatically pick the best one.

Conclusion

To summarize, in this assignment, we took on the challenge of slovene paraphrase generation. We first had to construct a dataset using paragraph-paraphrase pairs. For this, we utilised machine translation models, where we took paragraphs from a slovene corpus, translated them into english and back, obtaining the corresponding pairs of paraphrases. We then filtered the dataset with ROUGE score, trying prune any nonsensical or identical pairs of paraphrases. Next, we successfully trained two T5-based models, which performed better than the selected baseline. Still, they leave a lot of room for improvement as frequent mirroring of the input sentence plagues the outputs. Lastly, we evaluatied models using automatic, as well as manual scoring, and compared the results in terms of correlation coefficients.

References

[1] Jianing Zhou and Suma Bhat. Paraphrase generation: A survey of the state of the art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086, 2021.

[2] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. *CoRR*, abs/1702.03814, 2017.

[3] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[4] Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. A continuously growing dataset of sentential paraphrases. In *Proceedings of The 2017 Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 1235–1245. Association for Computational Linguistics, 2017.

[5] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[6] Christian Federmann, Oussama Elachqar, and Chris Quirk. Multilingual whispers: Generating paraphrases

Table 3. Examples of good and bad results of our paraphrasing models. When performing well, T5 models can change the paragraph structure, merging the individual sentences into a single one. When performing poorly, paragraphs are just copies of inputs and information is left out.

(a) Good example

Input	V petek, 4. marca, so gasilci Gasilske brigade Ljubljana še preverili madež, ki se je pojavil na obrežju Save. Ugotovili so, da gre za odmrle alge, in ne za naftni madež.
Backtranslation	Gasilci Gasilske brigade Ljubljana so v petek, 4. marca, preverili madež, ki se je pojavil na obrežju Save, in ugotovili, da gre za mrtve alge in ne za oljni madež.
Baseline	V petek, 4. marca, so gasilka Gasilske brigade Ljubljana še iti skozi packa, ki se je pojavil na okolica dokov Save. dojeti so, da teči za mrtev alge, in ne za naftonosen flek.
Slo-t5-small	V petek, 4. marca, so gasilci Gasilske brigade Ljubljana še enkrat preverili madež, ki se je pojavil na obrežju Save, in ugotovili, da gre za odmrle alge, in ne za naftni madež.
Slo-t5-large	V petek, 4. marca, so gasilci Gasilske brigade Ljubljana preverili madež, ki se je pojavil na obrežju Save, ugotovili pa so, da gre za odmrle alge in ne za naftni madež.

(b) Bad example

Input	Storitev Si.mail je v preizkusnem obdobju (beta različica), zato vas v primeru morebitnih napak, prosimo da jih sporočite v naš klicni center na brezplačno številko 080 40 40 40 .
Backtranslation	Storitev Si.mail je v poskusnem obdobju (beta verzija), zato vas prosimo, da v primeru morebitnih napak, le-te pošljete na naš klicni center na brezplačno telefonsko številko 080 40 40 40.
Baseline	zagrešitev Si.mail valjati se v poskusen razdobje (beta inačica), zato vas v zgled možen nerodnost, zavzeti se da jih razglasiti v naš nagovoren sredinski igralec na gratis izvod 080 40 40 40.
Slo-t5-small	Storitev Si.mail je v preizkusnem obdobju (beta različica), zato vas prosimo, da jih pošljete v naš klicni center na brezplačno številko 080 40 40 40
Slo-t5-large	Si.mail storitev je v preizkusnem obdobju (beta različica), zato vas prosimo, da jih sporočite v naš klicni center na brezplačno številko 080 40 40 40.

with translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 17–26, Hong Kong, China, November 2019. Association for Computational Linguistics.

[7] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

[8] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[9] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[10] Lingfeng Shen, Lemao Liu, Haiyun Jiang, and Shuming Shi. On the evaluation metrics for paraphrase generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3178–3190, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[11] Omkar Patil, Rahul Singh, and Tarun Joshi. Understanding metrics for paraphrasing, 2022.

[12] Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity - multilingual and cross-lingual focused evaluation. *CoRR*, abs/1708.00055, 2017.

[13] Iztok Lebar Bajec, Andraž Repar, Jure Demšar, Žan Bajec, Mitja Rizvič, Borut Kumperščak, and Marko Bajec. Neural machine translation model for slovene-english language pair RSDO-DS4-NMT 1.2.6, 2022. Slovenian language resource repository CLARIN.SI.

[14] Nataša Logar, Tomaž Erjavec, Simon Krek, Miha Grčar, and Peter Holozan. Written corpus ccGigafida 1.0, 2013. Slovenian language resource repository CLARIN.SI.

[15] OFMCap br. Matej Nastran. Biblija.net - the bible on the internet.

[16] Matej Ulčar and Marko Robnik-Šikonja. Sequence to sequence pretraining for a less-resourced slovenian language, 2023.

[17] Nikola Ljubešić. The CLASSLA-StanfordNLP model for lemmatisation of standard serbian, 2019. Slovenian language resource repository CLARIN.SI.

[18] Simon Krek, Cyprian Laskowski, Marko Robnik-Šikonja, Iztok Kosem, Špela Arhar Holdt, Polona Gantar, Jaka Čibej, Vojko Gorjanc, Bojan Klemenc, and Kaja Dobrovoljc. Thesaurus of modern slovene 1.0, 2018. Slovenian language resource repository CLARIN.SI.