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

Paraphrasing Sentences

Nela Petrželková, Žiga Patačko, Žan Horvat

Abstract

Sentence paraphrasing is a widely studied problem in natural language processing. In this paper, we discuss the problem of sentence paraphrasing and compare different approaches to this task. We introduce different evaluation metrics that are used to assess the quality of the models created. We present our approach to sentence paraphrasing and compare it with a model based on traditional methods. We also introduce our data set that was used to the model training. Our data set was generated using OpenAI's GPT-3 model, which is a novel approach to data set generation, and we also compare it to the traditional back-translation method.

Keywords
natural language processing, paraphrasing sentences, GPT-3, T5

Advisors: Slavko Žitnik

Introduction

Sentence paraphrasing is a widely studied problem in natural language processing. The main idea is to generate paraphrase for a given text - a sentence with the same semantic meaning but using different different words or structure of the sentence. This task is a challenge in many other downstream tasks, such as question answering [1], machine translation [2], information retrieval, text summarization [3] or semantic parsing [4] [5]. It is also used for data augmentation [6] or varying the answers of chatbots.

In recent years, the deep neural network approach overshadowed the classical approaches. Several architectures were tried and we will discuss that in Section 1 as well as the traditional methods. In Section 2 we will present our approach to the problem. In particular we will introduce the dataset, compare different methods and models. Results will be discussed in Section 3.

1. Related Work

In this Section, we will compare different approaches to sentence paraphrasing, mainly the traditional ones and the neural ones. We will also discuss different evaluation metrics used in this type of task.

Approaches to Sentence Paraphrasing

The first attempts to solve the problem of sentence paraphrasing were built on hand-crafted [7] and later also on automatically obtained [8] paraphrase rules. The next approach was based on substituting some words with their synonyms extracted from a thesaurus and the best word was chosen given the context in the image [9]. Alternative approach is using statistical machine translation and its main idea is that the task of paraphrasing is treated as a monolingual machine translation [10].

With recent rise of neural networks, these models outperformed the traditional approaches. Namely the sequence-to-sequence models [11] were the first models to be used for the task of sentence paraphrasing. They were an inspiration for other models and nowadays the majority of models are sequence-to-sequence models consisting of encoder and decoder. Encoder compresses the initial text into a vector representation where the semantic meaning is captured. Decoder than generates the paraphrases based on the vectors. These models were then improved in two ways. First is model-focused, enhancing the encoder and decoder and second is attribute-focused, which aims to enhance the quality of the generated sentences.

Evaluation Methods

One of the problematic parts of sentence paraphrasing is how to evaluate the generated paraphrases. There is no ground truth in this case. There are however three important aspects, that must be considered in the paraphrase generation.

1. **Preserving semantic meaning** – From the definition of paraphrase, the sentences must have the same meaning.
2. **Variation in word choice and grammar** – The sen-

tence must be different on syntactic and lexical level.

3. **Language fluency** – The generated sentence must be meaningful.

One of the approaches to measure this is simple human evaluation, which provides accurate and qualitative evaluation. However, it is more costly than an automatic evaluation.

There were developed several automatic evaluation metrics, that take in account the three previously mentioned aspects. Namely these are BLEU [12] (originally developed for machine translation), METEOR [13] that improves BLEU in its deficiencies, ROUGE [14](originally developed for text summarization) and TER [15] (also for machine translation) and basically they measure the number of edits human translator would need to make to match the reference sentence.

2. Methods

This section discusses the methods used to obtain the data set as well as the base model, that will be used to compare our trained model, that is also introduced in this section.

Data set

First, we had to create the data to be able to evaluate the base model and train our model. The corpus we have chosen is called Corpus of comma placement Vejica 1.3 [16]. It consist of 104184 sentences, which come from five different sources.

The corpus was first filtered to get rid of most of the sentences that don't make sense, are too short as well as some of those, that are not grammatically correct. This yielded 71174 sentences.

In order to be able to train our model, a data set needs to have a list of tuples, consisting of an original sentence and its paraphrased version.

In order to create the dataset with original sentences and paraphrases, methods like back-translations are usually used. However, we decided to use large language models in order to generate the data set for training our smaller model. OpenAi's GPT-3 model (specifically `text-davinci-003`) and its API interface allow us to essentially ask a high-level model to generate paraphrased sentences for us.

Several methods for prompting the model to generate a paraphrased sentence were tested. The best one appears to be to instruct the model to pretend it is a paraphrasing bot and then request the completion of the text. The exact prompt was defined as follows:

```
I am a Slovene paraphrasing bot. I will
    read a Slovene sentence and rewrite
    it with a the same meaning but in
    different words.

Sentence: [sentence]
Paraphrased:
```

The model the completes the text by generating a sentence. This is a novel approach that only became viable recently. Our aim is to explore and analyze its properties and compare it to a well-established back-translation method.

Due to limitations in the OpenAI's API and the steep pricing, only 12600 pairs of the original an paraphrased sentences were generated.

Reference data set - Back translation

The back translation method was used to establish the reference data set. This method generates paraphrased sentences by using machine translation. It starts with the original sentence in a language and then uses two or more translations in order to get a similar but not the same sentence in the original language. For example, to generate a paraphrased Slovene sentence, one might first translate it into Chinese, then German and finally back to Slovene.

In our case, we utilized the Google Translate Python API and back-translated 11990 sentences from Slovene to English and back to Slovene.

2.1 Thesaurus-based approach: Synonym substitution

One of the classical approaches of paraphrasing sentences is simple substitution of some words by their synonyms. This method does not bring much variability in the output but it is one of the first approaches examined and we wanted to try it as well.

We used Stanza [17] NLP library for a task of part-of-speech tagging. We performed this action because we wanted to replace only nouns, adjectives or verbs (tags NOUN, ADJ, VERB respectively).

For the purpose of looking up the synonyms we used the Thesaurus of Modern Slovene 1.0 [18] which was automatically obtained by English-Slovene dictionary, monolingual dictionary and a corpus. What makes things more complicated in Slovenia is the presence of inflection of the words. Therefore often we cannot simply lookup the words in the dictionary since it does not match them in they are not in the default form.

Not surprisingly, this approach did not yield very good results. Many of the sentences were the same since no suitable word for synonym was found and even if yes, the syntax stays the same so the paraphrased sentences suffered from low diversity.

Base model

In order to design a Slovene language model for paraphrasing sentences we decided to base it on pre-trained small Slovene T5 model. A GPT-2 based model was also tried out but did not yield sufficiently promising results.

The usage of a pre-trained model allows us to save valuable time and processing power for training while enabling us to train it for the specific task at hand at the same time. The base model can be found on the Hugging Face website under the name `cjvt/t5-sl-small` [19].

Fine tuned model

The model was fine-tuned on two previously mentioned data sets with the following settings:

1. Using the AI-generated dataset
    - Train dataset size: 12600
    - Evaluation dataset size: 1400
    - Learning rate: $4e-5$
    - Batch size: 16
    - Weight decay: 0.01
    - Epochs: 128
2. Using the back-translation dataset
    - Train dataset size: 11990
    - Evaluation dataset size: 1132
    - Learning rate: $4e-5$
    - Batch size: 16
    - Weight decay: 0.01
    - Epochs: 384

A custom loss function was also implemented in order to penalize models that simply repeat back whatever was their input. This did not improve the training results and was therefore discarded after a few tests.

The trained model can then be prompted and responds with a paraphrased sentence. Some examples from the model trained using the first set of settings on an AI-generated data set:

```
Prompt: Poleg tega je tudi zapustil
    tiste, ki ga imajo najraje.
Response: Prav tako je zapustil tiste,
    ki ga najbolj ljubijo.

Prompt: Torej je Ismena zelo
    upogibljivo drevo.
Response: Zato je Ismena zelo
    upogibalno drevo.
```

2.2 Evaluation methods used

The model was only evaluated by hand based on the metrics presented above We main criteria of high quality output were preserving semantic meaning, variation in word choice and grammar and language fluency. 100 random original sentences were chosen from each of the data set and the paraphrasing was performed. Then, the transformation was ranked from 1 to 3 in each of the criteria.

The data sets ranked were:

- GPT-3 generated data,
- back translation generated data,
- T5 model trained on GPT-3 generated data and
- T5 model trained on back translation generated data.

The results are presented in Chapter 3 and discussed in more detail in Chapter 4.

3. Results

The evaluation results for each of the models and data sets are presented in Table 1.

Model Name	Meaning	Fluency	Variation
Back-translation	2.48	2.62	1.72
Model using AI data	1.96	1.63	1.55
Back-transl. model	1.34	1.52	1.10
OpenAI's model	2.40	2.36	2.52

Table 1. Evaluation results

Figure 1 depicts a comparison among the models based on their scores in preserving semantic meaning, language fluency, and variation in word choice and grammar.

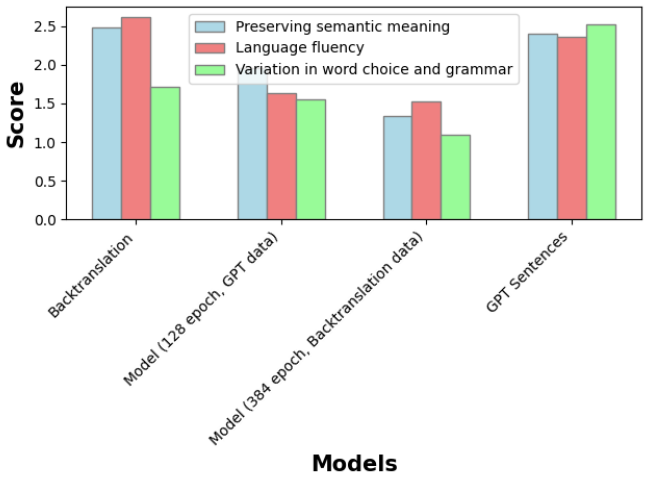

Figure 1. Comparison of different models

4. Discussion

Looking at the results in Table 1, it is evident that the back-translation method outperforms the other models in terms of preserving semantic meaning and language fluency, with scores of 2.48 and 2.62 respectively. However, it scored relatively lower in terms of variation in word choice and grammar, with a score of 1.72. The relatively lower variation score may be attributed to the nature of back-translation, which can be a bit conservative in order to preserve the original meaning.

The T5 model trained on GPT-generated data shows a moderate performance in terms of semantic meaning preservation and variation in word choice and grammar, while lan-

guage fluency is somewhat compromised. This might be due to the fact the original sentences are quite complex.

The same model trained on back-translation data, with a higher number of epochs (384), shows inferior performance in comparison. This suggests that simply increasing the training epochs does not necessarily improve the model's performance.

Interestingly, the results from the OpenAI's GPT-3 model show high scores in all aspects, with the highest in variation in word choice and grammar, suggesting that the model is very capable of generating diverse sentences while preserving semantics and maintaining fluency. It is important to note, however, that these results are for the high-level GPT-3 model generating paraphrases, and not a fine-tuned model.

5. Future Work

Based on the results of this study, we can propose several directions for future work.

- A data set with a bit simpler sentences should be chosen for any further work. This would greatly simplify the task each model needs to learn.
- A GPT model did not provide any promising results, probably due to its small size. A bigger GPT-based model should be tested in the future.
- Additionally, the evaluation method could be refined. Using automatic evaluation metrics in combination with human evaluation could provide a more comprehensive view of the models' performances.
- It might be worthwhile to investigate the use of more sophisticated data augmentation techniques, such as those leveraging different types of paraphrasing other than back-translation and synonym substitution.

In conclusion, sentence paraphrasing is a complex task in the field of natural language processing, and while current techniques and models show promising results, there is still room for improvement. Future research could focus on developing novel approaches to improve the generation of diverse, fluent, and semantically coherent paraphrases.

References

[1] Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[2] Tomoyuki Kajiwara. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6047–6052, Florence, Italy, July 2019. Association for Computational Linguistics.

[3] Ziqiang Cao, Chuwei Luo, Wenjie Li, and Sujian Li. Joint copying and restricted generation for paraphrase. *CoRR*, abs/1611.09235, 2016.

[4] Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[5] Ruisheng Cao, Su Zhu, Chenyu Yang, Chen Liu, Rao Ma, Yanbin Zhao, Lu Chen, and Kai Yu. Unsupervised dual paraphrasing for two-stage semantic parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6806–6817, Online, July 2020. Association for Computational Linguistics.

[6] Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. Paraphrase augmented task-oriented dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 639–649, Online, July 2020. Association for Computational Linguistics.

[7] Kathleen R. McKeown. Paraphrasing questions using given and new information. *American Journal of Computational Linguistics*, 9(1):1–10, 1983.

[8] Regina Barzilay and Lillian Lee. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 16–23, 2003.

[9] David Kauchak and Regina Barzilay. Paraphrasing for automatic evaluation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 455–462, New York City, USA, June 2006. Association for Computational Linguistics.

[10] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[11] Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek V. Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. Neural paraphrase generation with stacked residual LSTM networks. *CoRR*, abs/1610.03098, 2016.

[12] 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, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[13] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

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

[15] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, volume 200. Citeseer, 2006.

[16] Peter Holozan. Corpus of comma placement vejica 1.3, 2018. Slovenian language resource repository CLARIN.SI.

[17] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

[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.

[19] Hugging Face. Cjvt/t5-sl-small · hugging face.