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

Paraphrasing sentences

Aljaž Grdadolnik, Anže Mihevc, and Luka Galjot

Abstract

This paper discusses the use of deep learning models, particularly transformer-based language models like T5, to implement sentence paraphrasing, which is a crucial aspect of natural language processing tasks. The aim is to use a fine-tuned version of T5 to generate paraphrases of sentences by selecting a corpus that includes semantically similar paraphrases while preserving the original meaning. The paper proposes to validate the model using both manual and automatic metrics. The implementation of this approach has the potential to improve the efficiency of natural language processing tasks.

Keywords
Transformer-Based Language Models, T5, paraphrasing, Transformer-Based Language Models

Advisors: Slavko Žitnik

Introduction

Paraphrasing, the process of expressing the same meaning using different words is an important part of natural language processing tasks. Recently, deep learning models, particularly the transformer-based language models like BERT, have achieved remarkable results in several language processing tasks. This paper aims to implement sentence paraphrasing using T5 Text-to-Text Transfer Transformer model, which has demonstrated state-of-the-art performance in range of processing tasks. Specifically we will use fine-tuned version of T5 [1] to generate paraphrases of sentences. Fine-tuning will be performed by one of the selected corpora, which will be preprocessed to include semantically similar paraphrases while preserving the meaning of the original sentence. We will validate our model with manually defined metrics and automatic methods.

Related work and existing solutions

Recently with emergence of ChatGPT service many of other companies and academic groups have speed up development of their own LLMs such as GPT-2 (by OpenAI), GPT-3 (by OpenAI), T5 (by Raffel et al. [2]), RoBERTa (by Liu et al. [3]), LLaMa (by Touvron et al. [4]).

In the article ***The Multilingual Paraphrase Database***[5] the authors expand their English and Spanish paraphrase database to include 21 other languages including Slovene. They extract the paraphrases from bilingual parallel corpora by "pivoting" words and phrases over a shared translation in another language (English). They make the database freely available.

Initial ideas

This section aims to provide initial ideas on how to approach the classroom project.

0.1 Dataset acquisition/generation

Perhaps the most important part of training a good model for paraphrasing is starting with the right dataset. The dataset should include quality pairs of phrases and their paraphrases. Such datasets already exist mainly for the English language such as The Paraphrase Database (PPDB)[6]. We have found an extended version of PPDB that includes other languages including Slovene[5]. Upon further investigation we have determined that the phrases included in this extended database are often only synonyms and the "source" phrases are not longer than 6 words. This dataset was produced using the back translation method or by "pivoting" over a shared translation in another language, in this case English. We would like to see if we get better results using a different method such as automatic translation of non-Slovene dataset. We will use a translation model provided in the instructions of this assignment [7] to automatically translate the original PPDB to Slovene. Afterwards we will check the quality and similarity of the translated pairs using different metrics. If our method produced better results than the back translation method we would use the automatically translated dataset to train our paraphrasing model.

More recently the QUORA[8] and MSCOCO[9] dataset have been used to train and evaluate paraphrasing models. The QUORA dataset is a dataset of duplicate questions posted on the Quora website. This questions are manually marked by moderators as duplicates so they are confirmed paraphrases. We could use automatic translation to obtain paraphrasing pairs in Slovene. The MSCOCO dataset is not a dataset originally made for paraphrasing or language processing rather it is a dataset of objects and object recognition within images. However the dataset specifically the annotations of the images can still be used to train and test paraphrasing models. To use this model in our assignment we would again need to automatically translate it and check for the quality of the translated pairs.

0.2 Natural language deep learning model

To implement paraphrasing, we can use a deep learning model to fine-tune T5 model on pairs of sentences. The T5 model is a transformer-based model that is pre-trained on a large corpus of text. Fine-tuning the T5 model on pairs of sentences will allow us to train the model to generate paraphrases of the input sentences. The deep learning model will learn to generate paraphrases by minimizing the difference between the predicted paraphrase and the actual paraphrase. This approach has been shown to be effective in generating high-quality paraphrases.

0.3 Result evaluation

The quality of the implemented model has to be somehow evaluated so we can compare it to other already existing models. For this we will use different metrics of evaluation. These are BLEU (*Bilingual Evaluation Understudy)* which was developed to evaluate machine translation systems, but can be also used for paragraph evaluation, METEOR(*Machine Translation Evaluation Overlap Rate*) which aims to address BLEU's weakness of being unable to measure semantic equivalents when applied to low-resource languages and has a better correlation with human judgement at the sentence/segment level than BLEU(*Google-BLEU*), GLEU, ROUGE(*Recall-Oriented Understudy for Gisting Evaluation*) a recall-based evaluation metric originally developed for text summarization, has also been used to evaluate paraphrase generation, BERTscore that is based on contextual embeddings from BERT and cosine similarity between the generated sentence and one or more reference sentences and WER(*Word Error Rate*), which is a metric that measures the similarity between two sentences by counting the number of word insertions, deletions and substitutions needed to transform one sentence into another[10]. And lastly we will use human evaluation as a metric, which will probably be the best in scoring, but most definitely the most tedious and time consuming.

Second submission

This part of the paper aims to describe our efforts in the time between the first and second submission.

0.4 Datasets

In second submission we had a deeper dive into researching and acquiring data to fine-tune our model.

0.4.1 Proposed datasets

After testing some of the proposed data sets we determined that the inputs would not be long enough for our needs. Most of the sentences were not full sentences but only partial sentences which were less than 6 words long. This would not work for our assignment thus we pivoted to other possible solutions.

0.4.2 Subtitles

At first we had an idea that the subtitle would be a great idea, because there are a lot of movies and each of them have defined sentences by time-code.

Without knowing much about usability of subtitles, we have created a small corpora of 60 movies with English and Slovene subtitles. We used Selenium based website scraper to fetch required data. When we gathered the files, we started on editing and aligning them. We realised that would be to much hand work for the end result.

The are many issues with using subtitles as a source. We found out that versions are not line to line aligned and if we wanted to use it, we would need to match them manually. Mostly English versions of subtitles also include description of sounds for deaf people. Slovene versions also include translated movie title somewhere in the beginning and credits for editors and translators of that movie somewhere in the end. All of that unusable words would have to be removed before matching the sentences between English and Slovene versions.

0.4.3 MaCoCu

For the corpus in this part of the project we used *Slovene web corpus MaCoCu-sl 1.0* which was built by crawling the *.si* internet domain in 2021. The corpus is made of 3176311 sentences in both English and Slovene.

The corpus contains one file where in each line there is an English sentence and it's Slovene counterpart separated by a tab.

We then translated (with a help of nlp-course-skupina-8) the English part to Slovene and made a new file which consists of two similar Slovene sentences separated with a tab. After translation there were 2884462 sentences left, because in the translation we pruned some of the sentences if they were too short or had too many special characters or numbers. Once that was done, we uploaded the corpus on a *sharepoint* link and on *hugging face*(but only a corpus of 250 thousand sentences, these are the same we trained our model on) in repository *hieronymusa/MaCoCu-dataset-250k* with three files stored in folder called data. The three files are train, validation and test csv file. These and the one on *sharepoint* can be used by any other group. We can add that there was an update to the *Slovene web corpus MaCoCu-sl 1.0* called *Slovene web corpus MaCoCu-sl 2.0* uploaded just one day after we translated the version one.

Once we had the original Slovenian sentences and the translated sentences we had to pre-process the data so that we could use it to fine tune our model. We wrote a short python script that creates one csv file from the files with original and translated sentences. Each line in the generated csv file consists of 3 columns: ID, Original and Translated[Paraphrased]. ID column is simply the sequential number of the line, the Original column is the original Slovenian sentence and the Translated[Paraphrased] is the the translated Slovenian sentence. The script can create sub-sets of the dataset starting and finishing the subset at arbitrary points in the dataset. We used the script to create smaller sub-sets of the dataset to enable us to train the model faster. This greatly reduces the quality of the results but enables us to check the influence of different parameters such as batch size and number of epochs on the training time. Once we find the optimal parameters we will use the entire dataset to train our model.

0.5 Models

We discovered that the sloBerta model would not work for our task as it is not a generative model. We looked at the generative models that would be better suited for our task and decided to use the T5 model. More specifically the T5-sl model which is already capable of producing Slovene text. For the initial stage we will be using the *t5-sl-small* version of the model as we want relatively fast iterations so we can test our code and different parameters. This model has 8 encoder and 8 decoder layers with total of about 60 million parameters and was trained for five epochs on the corpora:

- Gigafida 2.0,
- Kas 1.0,
- Janes 1.0,
- Slovenian parliamentary corpus siParl 2.0,
- slWaC

By the end of the project we will most likely use the bigger versions of this model called *t5-sl-large* to get better results.

Even with the smaller model our home computers take a long time to fine tune the model. Thus we decided to run the training of our model on the ARNES cluster. Here we set up a singularity container using a Pytorch docker image as a base and installing some additional required packages. We ran into some problems when creating the images as each user only has 20GB of disk space allocated on his home directory. This was solved by saving the container elsewhere on the cluster.

0.6 Code and training the model

The code we used for training our model is available on our GitHub[[11]]. We first load our model *t5-sl-small* then load the dataset from a local .csv file. We experimented with creating and uploading our own dataset to Huggingface website which we were able to do for the 250k long sub-set of our dataset. To have more granular and dynamic control over the data we use we still load the data from a local file but in the future we might upload the entire dataset to the Huggingface website and load the data from there.

After the data is loaded we initialize our metrics and tokenizer. Before we can use the data to train our model we have to add the prefix ”paraphrase” to our original sentences so the model will learn what to do when it encounters the prefix. We use the following arguments for the trainer object:

- evaluation_strategy = ”epoch”,
- learning_rate=2e-5,
- per_device_train_batch_size=batch_size,
- per_device_eval_batch_size=batch_size,
- weight_decay=0.01,
- save_total_limit=3,
- num_train_epochs=1,
- predict_with_generate=True,
- optim=”adamw_torch”,
- fp16=False,
- report_to=”none”,

When the training is complete we save the model to a directory.

To generate outputs using our model we can load our model from the directory in which we previously saved it. Then we provide the model with our input sentence which must start with the same prefix that we used in training the dataset. We use a tokenizer to tokenize the input and pass it to our model which generates a prediction. We then decode the prediction and print it out.

0.6.1 Results

Because we use only a small sub-set of the dataset and a low number of epochs the model returns the same sentence as the input most of the time. It does sometimes correct small grammatical errors and in some cases changes a word. We can see some examples bellow and even more examples in the Appendix1.

- *Same sentence as input:*
- **Input**: paraphrase: Danes je lep dan
- **Output**: Danes je lep dan
- **Input**: paraphrase: S tem bomo preprečili onesnaževanje narave in ohranjali zdravo okolje.
- **Output**: S tem bomo preprečili onesnaževanje narave in ohranjali zdravo okolje.
- *Grammatical correction:*

- **Input**: paraphrase: Danes sem šel v trgovin in kupil kruh, mleko in jajca.

- **Output**: Danes sem šel v trgovino in kupil kruh, mleko in jajca.

- **Input**: paraphrase: Ta denar nisem ukradel.

- **Output**: Ta denar sem ukradel.

0.7 Evaluation of the model

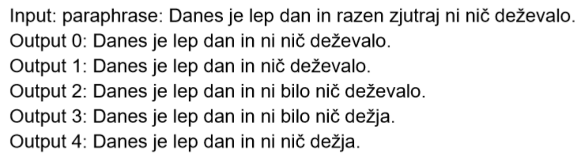
First we used the ROUGE metrics where we got some almost decent results, but since the model so far mostly just returns the given input/sentence, this is expected. In the future we should not use this metrics since it is not a good evaluation for paraphrases but more for text summarization and machine translation, which makes sense, why it gave us pretty good scores.

We also used BLEU metric to evaluate our model. We got a pretty low score, which was expected but we also expect that we will get a higher when the model will be trained on a much larger dataset. The few sentences that we checked by hand were also as good as expected, which means that they are quite similar in the begging, but somewhat differ at the end of the sentences.

0.8 Future work

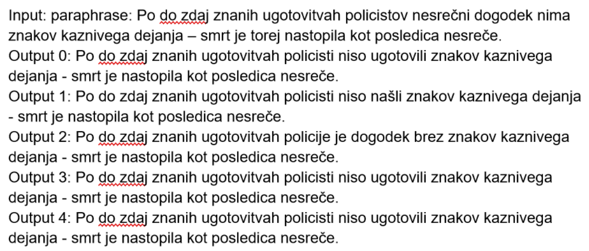
In the time to the final submission our focus will be to train our model on much larger dataset and substitute T5-sl-small with T5-sl-large model to start with larger base. We also need to better define our manual metrics.

1. Appendix



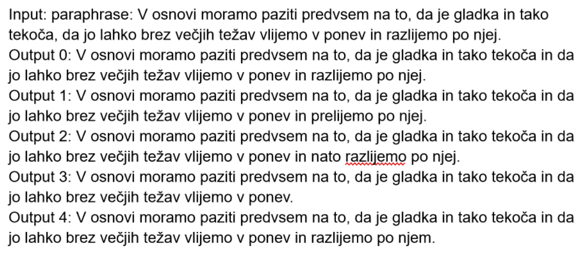
Input: paraphrase: Danes je lep dan in razen zjutraj ni nič deževalo.
Output 0: Danes je lep dan in ni nič deževalo.
Output 1: Danes je lep dan in nič deževalo.
Output 2: Danes je lep dan in ni bilo nič deževalo.
Output 3: Danes je lep dan in ni bilo nič dežja.
Output 4: Danes je lep dan in ni nič dežja.

Figure 1. Input and 5 different outputs of our model.



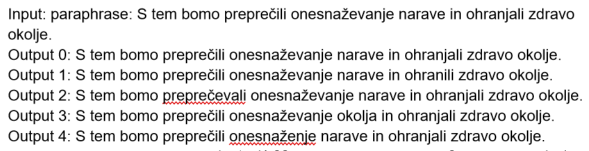
Input: paraphrase: Po do zdaj znanih ugotovitvah policistov nesrečni dogodek nima znakov kaznivega dejanja – smrt je torej nastopila kot posledica nesreče.
Output 0: Po do zdaj znanih ugotovitvah policisti niso ugotovili znakov kaznivega dejanja - smrt je nastopila kot posledica nesreče.
Output 1: Po do zdaj znanih ugotovitvah policisti niso našli znakov kaznivega dejanja - smrt je nastopila kot posledica nesreče.
Output 2: Po do zdaj znanih ugotovitvah policije je dogodek brez znakov kaznivega dejanja - smrt je nastopila kot posledica nesreče.
Output 3: Po do zdaj znanih ugotovitvah policisti niso ugotovili znakov kaznivega dejanja - smrt je nastopila kot posledica nesreče.
Output 4: Po do zdaj znanih ugotovitvah policisti niso ugotovili znakov kaznivega dejanja - smrt je nastopila kot posledica nesreče.

Figure 2. Input and 5 different outputs of our model.



Input: paraphrase: V osnovi moramo paziti predvsem na to, da je gladka in tako tekoča, da jo lahko brez večjih težav vlijemo v ponev in razlijemo po njej.
Output 0: V osnovi moramo paziti predvsem na to, da je gladka in tako tekoča in da jo lahko brez večjih težav vlijemo v ponev in razlijemo po njej.
Output 1: V osnovi moramo paziti predvsem na to, da je gladka in tako tekoča in da jo lahko brez večjih težav vlijemo v ponev in prelijemo po njej.
Output 2: V osnovi moramo paziti predvsem na to, da je gladka in tako tekoča in da jo lahko brez večjih težav vlijemo v ponev in nato razlijemo po njej.
Output 3: V osnovi moramo paziti predvsem na to, da je gladka in tako tekoča in da jo lahko brez večjih težav vlijemo v ponev.
Output 4: V osnovi moramo paziti predvsem na to, da je gladka in tako tekoča in da jo lahko brez večjih težav vlijemo v ponev in razlijemo po njem.

Figure 3. Input and 5 different outputs of our model.



Input: paraphrase: S tem bomo preprečili onesnaževanje narave in ohranjali zdravo okolje.
Output 0: S tem bomo preprečili onesnaževanje narave in ohranjali zdravo okolje.
Output 1: S tem bomo preprečili onesnaževanje narave in ohranili zdravo okolje.
Output 2: S tem bomo preprečevali onesnaževanje narave in ohranjali zdravo okolje.
Output 3: S tem bomo preprečili onesnaževanje okolja in ohranjali zdravo okolje.
Output 4: S tem bomo preprečili onesnaženje narave in ohranjali zdravo okolje.

Figure 4. Input and 5 different outputs of our model.

References

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

[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019.

[3] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[4] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[5] Juri Ganitkevitch and Chris Callison-Burch. The multilingual paraphrase database. In *LREC*, pages 4276–4283. Citeseer, 2014.

[6] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, 2013.

[7] Institute "Jožef Stefan" and Clarin.si. Slovene nmt. https://github.com/clarinsi/Slovene_NMT, 2021. Accessed on March 20, 2023.

[8] Sambit Mishra. First quora dataset. https://www.kaggle.com/datasets/sambit7/first-quora-dataset, 2018. Accessed on March 20, 2023.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in

context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[10] 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.

[11] Microsoft. Github repository, 2023.