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

Multilingual paraphrasing of sentences

Jannik Weiß, Nikolay Vasilev, and Jan Jenicek

Abstract

This paper compares and evaluates monolingual paraphrasing of English, German, Czech, and Slovene sentences, along with multilingual paraphrasing across these languages. Monolingual datasets are generated and evaluated using human evaluation. By comparing their scores with the scores of existing monolingual datasets, we estimate their quality. The models, on the other hand, are evaluated using the metric Parascore, which helps us analyse the effectiveness of each model for each language. This way we discover the advantages and disadvantages that come with using a mono- or multilingual dataset and training for multilingual paraphrasing of sentences.

Keywords

paraphrasing, multilingual paraphrasing, natural language processing, Parascore

Advisors: Slavko Žitnik

Introduction

Paraphrasing is the process of restating a text or sentences using different words or sentence structures while retaining the original meaning. It is used to help people clarify information, adapt the information to a specific audience, avoid plagiarism and improve communication. Paraphrase generation is a sequence-to-sequence natural language processing (NLP) task (Zhou and Bhat, 2021) closely related to translation and summarization. A good paraphrase is semantically similar and lexically different to the original sentence. One approach of generating paraphrases is by using machine translation techniques. This approach is based on the assumption that by translating a sentence into another language and then translating it back to the original language, the generated sentence can be a valid paraphrase. In Figure 1 we can see example of this process. In this case the original sentence is in English (EN) and is translated into German (DE), Slovene (SI) or Czech (CZ) language in order to generate its EN paraphrase. From Lample and Conneau (2019) we know that leveraging translation models like that produces good paraphrases, however it still makes sense to create and train separate models for paraphrase generation. Reasons for this are for example the inevitable loss of information during translation into certain languages, better control and adaptability of the generation process, and interpretability of the model or its internal representations. However, translation approaches are a good asset for paraphrase corpus generation, because there is much more translation data available than paraphrase data.

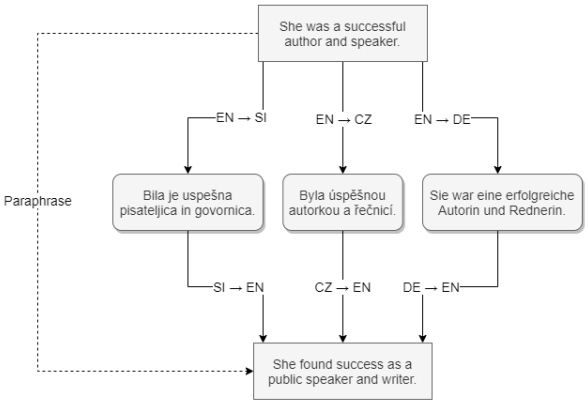

Figure 1. Multilingual paraphrasing of a sentence using the translation approach. Example of paraphrasing a sentence in English (EN) by translating it into Slovene (SI), German (DE) or Czech (CZ) language.

Paraphrase generation is closely related to translation as it can be seen as translation from one language into the same language, yet it is not inherently a cross-lingual task. Research has shown that in general multilingual models can outperform monolingual models on various cross-lingual NLP tasks (Lample and Conneau, 2019), and we ask ourselves how this applies to paraphrase generation. In other words, does train-

ing a multilingual model for paraphrase generation in one language actually help it with generating paraphrases in other languages (i.e. the model learns some abstract concept of what it means to paraphrase) or is this learning rather limited to the training language?

The paper aims to provide a comprehensive evaluation of monolingual paraphrasing models on four different languages and investigate the potential benefits of a multilingual paraphrasing model. We create paraphrase data from the paracrawl dataset Bañón et al. (2020) using machine translation models from huggingface. We then create simple monolingual paraphrasing models using data in the following languages: English, Slovene, German, Czech. After these baseline models are evaluated, we create two multilingual models, which are trained on all four languages. We find that while averaged across languages the multilingually trained models do not outperform the monolingually trained models, training a model multilingually can smooth out performance differences between languages compared to their monolingually trained counterparts.

Related work

The most popular approach nowadays for paraphrase generation is to use sequence-to-sequence systems (Zhou and Bhat, 2021), as was first done by Prakash et al. (2016). As is the case with other NLP tasks modern approaches are based on the Transformer architecture (Vaswani et al., 2017). Two main training approaches can be identified to develop paraphrase generation systems:

1. Training on paraphrase data – intuitive training approach that uses sentence pairs which are paraphrases of each other as training data. A recent example of this is Bandel et al. (2022) which also introduces a mechanism to control for semantic, syntactic and lexical dimensions of the generated paraphrase.

2. Training on translation data – Guo et al. (2019) and Thompson and Post (2020) formulate paraphrase generation as translation from one language to the same language, having trained a multilingual model on parallel datasets. This approach makes the system capable of multilingual paraphrase generation by definition. The multilingual training makes this approach better than pivoting (a.k.a. backtranslation) for paraphrase generation, because it avoids the potential errors made in the two translations. However, if the parallel dataset itself is created using machine translation this approach is prone to translation bias.

We use the first training approach with paraphrase data in our experiments.

Evaluation of paraphrase generation is not trivial, due to the generative nature of the task. Like in translation or summarization there are multiple correct answers to any particular paraphrase generation problem. While syntactic and lexical differences can be evaluated using rather simple metrics, evaluating semantic similarity is more involved. A current state of the art metric is Parascore (Shen et al., 2022), which we also use here in our experiments.

Dataset

We create 4 paraphrase datasets for our approach, one for each language. We aim to create similarly sized datasets for each language, which is why we do not use off-the-shelf popular paraphrase datasets like the PPDB by Pavlick et al. (2015) or the Tatoeba dataset by Tiedemann (2020). In our research these mostly contain enough data for English and often for German, but next to nothing for Slovene and Czech.

Figure 2 shows our dataset generation pipeline.

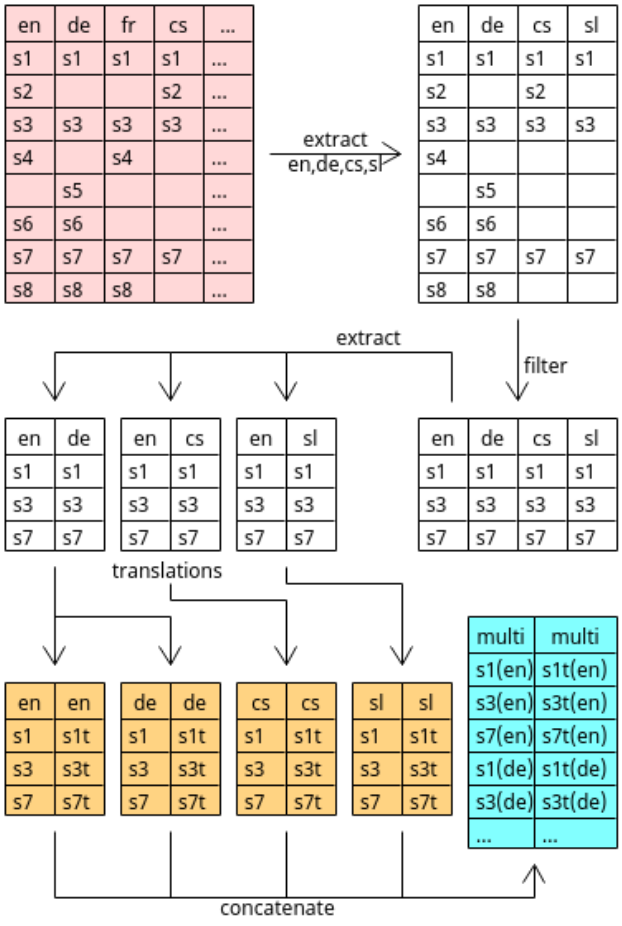

Figure 2. Dataset generation pipeline. ParaCrawl (pink), our monolingual datasets (orange) and our combined multilingual dataset (cyan).

We use the ParaCrawl dataset as a starting point. This large parallel dataset contains lots of sentences in lots of languages. We extract the Slovene-English, Czech-English, and German-English parts and join them on the English sentence to generate a parallel multilingual Slovene-German-Czech-English dataset. This ensures us that we only use sentences

that exist in all four languages. We also filter the length of the multilingual dataset to make sure that we have the same amount of data for all languages - 76,879 records.

The paraphrase datasets for individual languages are obtained by back-translation of another language. To generate the Slovene, Czech, and German paraphrase datasets, we back-translate the English sentences contained in the English column of the generated multilingual dataset. For the English paraphrase dataset we decided to back-translate the German sentences, as this language proved to provide the most consistent results compared to the other two.

We use the Huggingface Abid et al. (2023) framework to obtain the back-translations. We implemented two methods of translation, the inference API and the local translation, both via pre-trained models. We eventually used the local translation as the use of the inference API proved not to be feasible. From the pre-trained models, we have decided to use the ones from *Helsinki-NLP* for back-translation, namely:

- *opus-mt-en-sla* – back-translation of English language sentences to Slovene language sentences.
- *opus-mt-en-cs* – back-translation of English language sentences to Czech language sentences.
- *opus-mt-en-de* – back-translation of English language sentences to German language sentences.
- *opus-mt-de-en* – back-translation of German language sentences to English language sentences.

Following these steps, we extracted four monolingual paraphrase datasets – one for each of the four languages English, German, Slovene, Czech. We call them *mono-enen*, *mono-dede*, *mono-cscs* and *mono-slsl*. These are the datasets, on which we train our four monolingual paraphrasing models. For the training of the multilingual paraphrasing model, we simply create the following two datasets: *a)* one with total of 307,516 records, which we generate by merging all four monolingual datasets into one multilingual dataset; *b)* the other will be generated the same way as the first one, but will use only one fourth of each dataset (different one fourth from each dataset) and therefore will have total of 76,879 records e.g., the same size as each of the monolingual datasets. We call these multilingual dataset *multi-all* and *multi-small*. All generated datasets are published and freely available on huggingface.

Dataset evaluation

We evaluate the quality of our datasets via human evaluation of a dataset sample and in direct comparison to other popular paraphrase datasets. We rate semantic similarity and lexical divergence of a given sentence pair and calculate a score based on these 2 ratings. For this, we create a small framework which mixes sentences from different datasets and presents them to the user, which reduces the bias a user might have, when knowing, which dataset they are evaluating. Semantic similarity (sim) and lexical divergence (div) are scored with values from 0 to 10 and the score S is calculated as follows.

$$S = f_{sim}\left(\frac{\text{sim}}{10}\right) * f_{div}\left(\frac{\text{div}}{10}\right) \tag{1}$$

In the simplest form, f_{sim} and f_{div} could be the identity function, but other functions can be plugged in to punish or reward low (close to 0) or high (close to 1) values of sim and div. We use $f_{sim}(x) = x^2$ and $f_{div} = -1 * ((x-1)^2) + 1$, which is a mathematical formulation of our opinion that a candidate sentence with high semantic similarity, but low lexical divergence, is still an okay paraphrase, while a candidate sentence with high lexical divergence, but low semantic similarity is not a good paraphrase. Table 1 contains the results of our human evaluation of our datasets in comparison to the corresponding parts of the Tatoeba dataset. For English all 3 of us submit a human evaluation, the other languages only have one each. For Slovene, the Tatoeba dataset only really has 2 or 3 actually Slovene sentence pairs, others are also mislabeled as Slovene, which explains the extremely low score.

Language	Our Dataset	Tatoeba
en-en	0.256	0.307
de-de	0.291	0.588
cs-cs	0.189	0.210
sl-sl	0.271	0.015

Table 1. Average human evaluation scores of our datasets vs. Tatoeba

For a deeper insight into our data, we analysed a few sentence pairs. That helped us make the following discoveries:

1. **Small changes** – some sentences, lots of which come from the Tatoeba dataset, differ only in one word. We usually give sim: 9/10 and div: 1-3.

   Example from Tatoeba:

   "I thought Tom might be thirsty."

   "I thought that Tom might be thirsty."

2. **No changes** – some sentence pairs are identical, which is why we usually give sim: 10 and div: 0.

   Example from our dataset:

   "Sami baptized his children."

   "Sami baptised his children."

3. **Nonsense** – some sentences make no sense semantically or grammatically. These are very difficult to evaluate, because semantic similarity is hard to define. We usually give sim: <5 and div like we normally would.

   Example from our dataset:

   "Yes, with such a book can hold her husband every day journey to a new country for a romantic dinner."

   "Yes, you can keep with such a book her husband every day travel to a new country for a romantic dinner. All were agreed to get."

4. **Good paraphrases** – find some very good paraphrases can also be found in the datasets, which we usually rate sim: $\geq$7 and div $\geq$7.

   Example from our dataset:

   "Europe's intellectual property regime needs to be modernised."

   "The copyright system in Europe needs to be modernised."

Overall, it seems that our generated datasets include more nonsense sentence pairs and some identical sentences, but also some good paraphrases with large lexical difference. The Tatoeba dataset contains probably almost entirely grammatically correct and always different sentence pairs, but the lexical difference is often very small, to the point where one might not even colloquially call it a paraphrase yet. As we can see from the table, the Tatoeba datasets for the Czech and Slovene languages not only have low amount of data, but also have sentences with lower quality.

Methods

Monolingually trained models

To be able to best compare our models we use an mt5 model Xue et al. (2022), which is a T5 model pre-trained on multilingual data. There exist separate monolingual models for each of the languages we use, but they differ for example in architecture, training methods or training data. By using the same multilingual model for the monolingual baselines we do not use the model in its full capacity, but we have the same conditions for all models we want to compare. This puts us in a better position to interpret the results later.

We train 4 different mt5 models, each on one of the monolingual datasets, that we created. We refer to these models as mono models or monolingual models, even though they are originally multilingual mt5 models, because we train them only on the generated monolingual datasets.

We create a generic training script, of which we create specific versions for our 4 monolingually trained t5 models. We also use the Parascore metric to evaluate our models on the validation dataset split during training.

Lastly, we evaluate these 4 baseline models, by generating paraphrases for each of the monolingual datasets. Our evaluation scripts, as well as the training scripts, we run on the in Google Colab environments using the testing split of each dataset. The scripts help us generate tables in a CSV format, where each row contains the original sentence, its reference (its paraphrase from the monolingual dataset), its paraphrase (the generated paraphrase from the model) and their calculated Parascore result.

Multilingually trained models

We finetune a two multilingual mt5 models for multilingual paraphrase generations in all four languages. For one, we use our *multi-all* dataset and for the other one we use our *multi-small* dataset. We do this, so that we train one model with an equal amount of data as the baseline models and one with all the data.

We use the same type of base training and evaluation script as for the monolingual models and also evaluate these two models using Parascore e.g., create evaluation tables.

Results

All models, which are evaluated in this section, are published and freely available on huggingface.

Monolingually trained models

We train all 4 monolingual baseline models for 5 epochs with a batch size of 16 on their corresponding dataset. This usually took around 2 to 3 hours on a Tesla T4 GPU in the Google Colab environment.

Model	Parascore score
en-en	0.961
de-de	0.925
cs-cs	0.922
sl-sl	0.890

Table 2. Parascore test results of the 4 monolingually trained models

The Parascore evaluation results of the trained models are shown in Table 2. We see that the English model has a comparatively higher score and the Slovene models has a comparatively lower score. The scores of the German and Czech models are in the middle and very similar to each other.

Multilingually trained models

We train the multi-all model for 3 epochs with a batch size of 16 on the combined multilingual dataset. Since the multi-all dataset contains a lot of data, this training took around 7 hours on a Tesla T4 GPU in Google Colab. In the same way we train the smaller multi-small model on the reduced sized multilingual dataset, but for 5 epochs.

Test set part	Score multi-small	Score multi-all
whole test set	0.925	0.925
English part	0.938	0.939
German part	0.926	0.925
Czech part	0.922	0.922
Slovene part	0.915	0.914

Table 3. Parascore test results of the 2 multilingually trained models

The Parascore evaluation results of both models are shown in Table 3. It also shows the average scores for each of the 4 different language sub-parts of the test split of the dataset. We see that both models yield similar results with Parascore. On average the score is similar to the average score of the 4 monolingual models. We see an increase in paraphrase generation performance for Slovene, but a decrease for English, compared to their monolingual baselines.

Discussion

Because we train multilingual models with standard paraphrase data, where cross-lingualness does not exist within a sentence pair, it is not surprising to see that the multilingual models do not outperform the monolingual models on average. It seems that each paraphrase pair first and foremost has an effect on the weights of the model that have to do with the sentence pair's language. However, the language capabilities of the multilingual models are not completely separated, because we can see that training a multilingual model on data from different languages smooths out their performance differences - it lifts up its Slovene capabilities, however at the expense of its English capabilities.

We do not see a big difference between the multi-all and multi-small models, which suggests that the 4 times larger dataset size does not have much of an impact. This is against our intuition, even though the multi-all model was only trained for 3 epochs. One reason for this might be that the performance scores are already approaching their limit and such changes would be more visible when comparing model checkpoints from earlier in the training.

Lastly, we have to critically look at the evaluation metric. It is in the nature of the task that automatic evaluation of paraphrase generation is difficult. Even with multiple human annotators the inter-annotator agreement might not be very high, because different people will have different opinions of what a good paraphrase is. Parascore itself relies on a multilingual BERT model under the hood, which likely has had more pre-training in English than in Slovene. When we (Jannik, Nikolay and Jan) take a look at some samples of the model generated test paraphrases we can hardly judge big differences with certainty, because our human resources are very limited.

Further research could do similar comparisons to us, but train multilingual models using translation data like Thompson and Post (2020), which would arguably stimulate more the cross-lingual capabilities of the multilingual model. Our dataset also lends itself for this approach. Concerning the dataset we created, more filtering or augmentation methods could be used to increase its quality.

Conclusion

We have created a multilingual parallel paraphrase dataset for the languages English, German, Czech and Slovene, which are 4 languages with varying levels of available NLP resources. We use this dataset to compare training a multilingual paraphrase generation model to training 4 separate models for each language. Our results show that while a multilingually trained model do not outperform a monolingually trained model on average (over the languages), the multilingual model equals out performance differences between languages. This makes the multilingual model useful for applications, where more equal performance among different languages is desired, or simply to increase the performance of a relatively lower resource language such as Slovene.

References

Abubakar Abid, Matthew Carrigan, Lysandre Debut, Sylvain Gugger, Dawood Khan, Merve Noyan, Lucile Saulnier, Lewis Tunstall, and Leandro von Werra. Hugging face library, 2023. URL https://huggingface.co/.

Elron Bandel, Ranit Aharonov, Michal Shmueli-Scheuer, Ilya Shnayderman, Noam Slonim, and Liat Ein-Dor. Quality controlled paraphrase generation, 2022. URL https://arxiv.org/abs/2203.10940.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.417. URL https://aclanthology.org/2020.acl-main.417.

Yinpeng Guo, Yi Liao, Xin Jiang, Qing Zhang, Yibo Zhang, and Qun Liu. Zero-shot paraphrase generation with multilingual language models, 2019. URL https://arxiv.org/abs/1911.03597.

Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291, 2019. URL http://arxiv.org/abs/1901.07291.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2070. URL https://aclanthology.org/P15-2070.

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. URL http://arxiv.org/abs/1610.03098.

Shen, Liu, Jiang, and Shuming Shi. On the evaluation metrics for paraphrase generation, 2022. URL https://arxiv.org/abs/2202.08479.

Brian Thompson and Matt Post. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic

similarity from lexical and syntactic diversity. In *Proceedings of the Fifth Conference on Machine Translation*, pages 561–570, Online, November 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.wmt-1.67.

Jörg Tiedemann. The tatoeba translation challenge – realistic data sets for low resource and multilingual mt, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. Google's mt5, 2022. URL https://huggingface.co/google/mt5-base.

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, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.414. URL https://aclanthology.org/2021.emnlp-main.414.