Effect of data filtering techniques on neural machine translation performance for English-Polish biomedical domain

Kamil Kojs
IT University of Copenhagen
kako@itu.dk

Janos Mate
IT University of Copenhagen
janma@itu.dk

Jorge del Pozo Lerida
IT University of Copenhagen
jord@itu.dk

Mikołaj Barański
IT University of Copenhagen
mikba@itu.dk

Abstract

Not only a shortz version of the paper; this is your sales pitch. Cover the problem/research question, why is it relevant and how did you solve it? Commonly ended with your most important result/finding.

1 Introduction

Providing the context, the problem, the motivation and a clear research question (or several, if you break them down into multiple).

Recent advancements in Large Language Models (LLMs) have resulted in a notable increase in the size of model architectures. State-of-the-art models that are publicly accessible, such as mBart-50, reach parameter sizes upwards of 600 million (Ushio et al., 2023). This escalation in parameter size has consequentially led to increased training times for these models, necessitating the use of the most advanced GPUs available for their training. A common approach in training LLMs involves utilizing the entirety of the data present in the prepared training dataset. However, it is often the case that these datasets are not entirely cleansed of low-quality entries. Given that training datasets frequently comprise over 1 million samples, it is impractical to manually inspect and purify the entire dataset of such substandard samples. Consequently, it is typical that only a minimal fraction of the dataset is of high quality, with the majority being of poor quality.

We hypothesize that the inclusion of such low-quality data in the training process contributes minimally to the overall quality and performance of LLMs. Considering the recent developments in model parameter sizes and the associated increased training times, we propose that it is imperative to carefully inspect the training data. By limiting the training dataset to only high-quality data, we anticipate a substantial reduction in training time without detrimentally impacting model performance. To achieve this, we advocate for the implementation of filtering techniques designed to selectively retain only the highest quality data within the training dataset.

In this study, our attention is specifically directed towards the task of neural machine translation (NMT) within the medical domain, focusing on the translation of sentences from English to Polish. We contend that when fine-tuning NMT LLMs for a specific domain and language pair, the application of filtering techniques can significantly reduce the size of the training dataset, without adversely affecting the resultant model's performance. To empirically validate this claim, we introduce the following research question: "Does the application of filtering techniques in the fine-tuning of NMT models for a medical domain for EN-PL language pair lead to substantial reduction in training times, without greatly compromising model performance?" This research question underscores our belief in the efficiency of selective data usage, particularly in the context of domain-specific NMT tasks.

JORGE: I would then, after stating hypothesis, how we tried to prove if this holds and what results (summarised) we got. So basically a summary of our methodology and results

2 Related Work

Include at least 2 relevant scientific research paper which provide the basis or motivation for (parts of) your work.

Neural Machine Translation models require data of large quantity and high quality in order to be adapted to new domains (Koehn et al., 2018). Large data sets translated or inspected by humans are not available nor are they feasible to develop given the plethora of languages and domains. As a result, the NMT field largely depends on large web-crawled corpuses offering varying sentence quality.

To increase the quality of these large corpuses, automated filtering methods have been studied by the NMT field. Approaches include a plethora of methods such as outlier detection (Taghipour et al., 2011), discriminator models trained on synthetically noisy translations (Xu and Koehn, 2017) graph-based unsupervised models (Cui et al., 2013), and more recently LLM based classifiers or scorers (Açarçiçek et al., 2020). With the advent of LLMs, encoders specifically designed to create language-agnostic embeddings such as, LASER / LaBSE / MUSE and more, allowed researchers to directly score bi-lingual sentence similarity for dataset filtering with research indicating this approach can be competitive with more complex classifier-based models (Chaudhary et al., 2019).

Bane and Zaretskaya (2021), explored the use of filtering based on the quality of sentence-pairs in English-Japanese and English-German. The models used for filtering included a pre-trained Marian-scorer, and LASER, MUSE, and XLM-R embedding models. The authors found that all models, which limited the dataset size to 54%-73% of the original size, achieved comparable translation BLEU scores or even outperformed training on the full dataset. However, the majority of filtering methods couldn't outperform a random dataset downsizing. Only the Marian-based filtering consistently achieved higher BLEU scores than random. Another finding was the wide variance of performance between the two language pairs for the MUSE approach, which performed best for German while significantly reducing model performance for the Japanese translation. The findings of Bane and Zaretskaya (2021), motivated our study design, while indicating that results are likely to be nontransferable to different language pairs or topic domains.

Bane et al. (2022) conducted further experimentation on the strengths and weaknesses of specific filtering methods. The authors, collected and generated a dataset of translation pairs with 10 different types of noise or errors (artificially induced where needed). The authors found a custom trained Marian-Scorer achieved highest cleaning performance, while embedding-based methods such as XLM-R, MUSE, and LASER performing worse but still at a good level. The embedding-based methods were specifically good at eliminating number mismatches and missing segments, while also catching a large share of spelling and word order permutations in texts. Research of Bane et al. (2022), indicates that embedding based methods successfully tackle noise in translated segments without the need for additional costly calibration such as the Marian-Scorer.

3 Data

Here you describe your data sources and summarize what they consist of. In the final report, you are allowed to use additional data which you collect/find on your own. Make sure you motivate the choice and provide appropriate references.

The source for training the medical language model was the UFAL Medical Corpus (ufa). Access to this corpus requires registration, and it has a size of approximately 25 GB. However, for the project, only sentence pairs in Polish and English were utilized, significantly reducing the size of the dataset. In total, 1,116,773 translations were obtained from English to Polish. The medical domain corpus had already undergone preprocessing to eliminate duplicates, but the overall quality required additional refinements. JORGE: I'd be more concise and direct in this section in general

To enhance the training data quality, several modifications were implemented. Sentences where the Polish and English counterparts were identical (untranslated sentences) were excluded, sentence pairs with a length smaller than 15 alphabetical characters or longer than 200 characters in either language were removed, and sentences containing unique characters from Russian, French, Greek, Bulgarian, and Romanian were eliminated. These adjustments resulted in a final training dataset size of 711,720 — of which we sampled randomly 700k —, markedly improving its quality as it contains less noise.

JORGE: should this go in methods along with model? To tokenize the text, the MBart50Tokenizer (MBa) class from the Hugging Face transformers library was employed, using pre-trained from "facebook/mbart-large-50-one-to-many-mmt" with source and target language codes "en_XX" and "pl_PL" respectively.

For the test corpus, a dataset from Khresmoi (Dušek) was employed. This publicly available dataset consists of 1,500 high-quality Polish-English sentence pairs in the medical domain, and the quality was found to be satisfactory, thus it did not require any specific data cleaning procedures.

4 Methodology

Here you define and describe your methods, with precise mathematics where applicable. It is highly recommended to add a visualization of your main model. Make sure it is clear what data it is trained on, what features your use. Include a description of your evaluation setup, including a description of which evaluation metrics you chose.

To test our hypothesis we use four language-agnostic embedding models - LASER, MUSE, LaBSE and BERT - to filter the medical-domain data. The sentence pairs of the in-domain training data, were scored by the three methods using cosine similarity. That is, for each sentence an embedding was generated for its English and Polish language versions and a cosine similarity was calculated between these. Based on these scores we kept 20% and 40% of the best scoring sentences for each method. These six (two sizes per method) filtered datasets were used to create fine-tuned mBART50 models. To allow us to compare the effect the effect of filtering, a baseline model was trained on the full dataset[1] and further models were trained on randomly selected 20% and 40% subset.

4.1 mBART50

For machine translation, mBART50, developed by Facebook AI, was used (Tang et al., 2020). The model was chosen because it is one of the state-of-the-art models that can be employed for various tasks and a wide variety of languages, which makes it ideal for comparability and reproducibility of our results. The model is a multilingual Sequence-to-Sequence model pre-trained using the Multilingual Denoising Pretraining [1]. During this pre-training, the source documents are noised in two ways. Firstly, it masks 35% of the words in each instance by randomly sampling a span length according to a Poisson distribution ($\lambda = 3.5$). Secondly, it uses sentence permutation to change the original sentences' order. After that, the model has to reconstruct the original text. The decoder input is the original text shifted by one position. It uses an initial token called the language id <LID> to predict the sentence.

For our fine-tuning, the previously introduced UFAL Medical Corpus (ufa) was used, including its Polish and English translations. The corpus

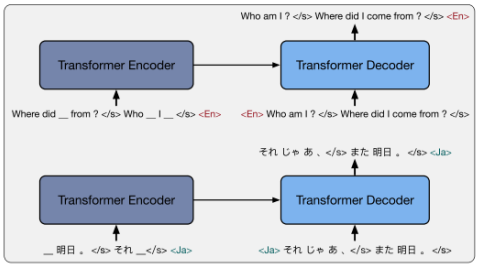

Figure 1: Multilingual Denoising Pre-training (mBART)

was tokenized using MBart50Tokenizer (MBa). In the training process, English sentences were translated into Polish sentences, and the evaluation was based on Sacre BLEU. Sacre BLEU is a metric used to evaluate the quality of translated text. In Sacre BLEU, the transition quality is measured on a scale from 0 to 100. Higher scores mean better transitions. It calculates the number of overlapping n-grams between the generated translation and the source translations.

TODO: add formula here and explain why better to use this bleu implementation briefly (some reference)

4.2 BERT

The BERT (Bidirectional Encoder Representations from Transformers) model is a popular model used for a wide variety of tasks. It was developed by Google and was published in the paper "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" (Devlin et al., 2019). As it employs bidirectional training, it achieves a deeper understanding of language context compared to single-direction language models.

4.3 LASER

The LASER (Language-Agnostic SEntence Representations) model (LAS), developed by Facebook AI Research, is an advanced method for generating language-independent sentence embeddings, offering significant advantages in multilingual natural language processing tasks. The core of the LASER model is a BiLSTM (Bidirectional Long Short-Term Memory) neural network architecture, which is trained on a large-scale, multilingual corpus. This BiLSTM encoder reads input sentences in any language and encodes them into a fixed-size vector, effectively capturing the deep semantic content of the sentence.

A key feature of LASER is its language-agnostic design. Unlike traditional language models that

[1] Note that while no embedding based filtering is applied for the baseline model's fine-tuning, it is trained on the cleaned dataset (see Sec. 3).

Dataset	Size	Training time	Baseline	LaBSE	MUSE	LASER
Base-none	n/a	n/a	✓			
Base-all	700k	00H:00M	✓			
Base-60%	420k	00H:00M	3 seeds			
Filtered-60%	426k	00H:00M		✓	✓	✓
Base-20%	140k	00H:00M	3 seeds			
Filtered-20%	142k	00H:00M		✓	✓	✓

Table 1: Dataset specifications and average training times. Base-none is the raw pre-trained model without any fine-tuning. Base-all, Base-20%, , Base-60% are models fine-tuned on all, 20%, and 60% of the training data respectively. Equivalently the filtered models were fine-tuned on subsets of the data selected as the highest scored sentence pairs for each filtering method.

require separate models or training processes for different languages, LASER is trained to handle input from over 90 languages, including low-resource languages, with a single model. This is achieved by training the model on parallel corpora, where the same sentence in multiple languages is used to teach the model to generate similar embeddings for semantically equivalent sentences, regardless of the language.

Furthermore, the model includes a Byte Pair Encoding (BPE) based tokenizer. This tokenizer segments text into subword units, allowing the model to handle diverse languages, including those with complex morphology or lacking word boundaries, like Chinese. The BPE tokenizer also aids in generalizing across languages by breaking down words into common subword units shared across different languages.

Once trained, the LASER model can be used to generate sentence embeddings for sentences in any of the languages it was trained on. These embeddings can then be used in various downstream tasks, such as cross-lingual text classification, information retrieval, and semantic similarity assessment, used in our work.

4.4 LaBSE

LaBSE - Language-agnostic BERT Sentence Embeddings, is method developed by a Google research team for creating BERT based cross-lingual sentence embeddings for over 109 languages (Feng et al., 2022).

4.5 MUSE

MUSE - Multilingual Unsupervised and Supervised Embeddings facilitates the faster and easier development and evaluation of cross-lingual word embeddings (met, 2023). It was developed by Facebook and it was published in Word Translation without Parallel Data paper (Conneau et al., 2017). The model incorporates unsupervised alignment of monolingual word embeddings involves three steps: adversarial training to create a linear mapping from the source to the target embedding space, extracting a synthetic dictionary from the shared space, and refining the alignment using the Procrustes solution. This enables alignment without the need for cross-lingual annotated data or parallel corpora (Conneau et al., 2017).

4.6 Experimental setup

For each of the filtering methods, we defined two different cutoffs that resulted in filtered versions containing 20% (∼150k) and 60% (∼ 420k) of initial data, on which we fine-tuned our model. We sampled equally sized sets with three different random seeds — to account for random variability — on which we fine-tuned our model too (*Base-20%* and *Base-60%*), which served as baselines trained on equal sizes to compare to. We also fine-tuned the model on the whole corpus, which represented the absolute fine-tuned baseline (*Base-all*). Altogether, this made a total of 15 models to train. Evaluation was computed for them all as well as for the untouched pre-trained model, which was our absolute non fine-tuned baseline (*Base-none*).

All computations were done in LUMI supercomputer, which uses Slurm as job scheduler. More specifically, we used the *standard-g* partition with AMD MI250x GPUs. Each model took different time to train due to varying train set sizes, and combined with evaluation computing time, we utilized a total of 646 GPU hours. For training, we used a 16-bit precision, a batch size of 15 for training and 20 for evaluation, a linear learning rate scheduler using first 100 steps and AdamW optimizer. All models were trained for 3 epochs, and best model based on evaluation loss was saved in each case.

5 Results

Here you provide the technical results of your method over the data. Any tables with numerical results should be put in this section. Use the most meaningful representation of your results that best illustrate the points you want to make. Make sure tables and figure have descriptive captions. If you use figures, make sure they contain labels for each axis.

JORGE:let's refer to baseline models as Base-none, Base-all, Base-20% and Base-60%. See section 4.6

JORGE: mention we could not get BERT 40 deu to technical issues

6 Discussion

Here you interpret the results back into the setting of the original research question and you dig deeper. What were major findings? What were major shortcomings of your methodology/data?

TODOs:

- Comment somewhere on how the nature of the embedding can have influenced results
- Explain that we expected all filtering methods to outperform, and did not find that for all
- Explain that BERT was expected to perform better, which it did not
- Discuss more complex vector comparison to be done to include more diverse methodologies

JORGE: I could not add direct references to table in results bcs not there yet

First things we can notice from our results — and that was already assumed — is that fine-tuning on in-domain data improves performance when testing on an unrelated in-domain dataset. *Base-none* shows bleu score of 14.936, worse than any other model that has been fine-tuned. The opposite extreme is represented by *Base-all*, which has been trained on the whole corpus and shows a sacrebleu of 17.402, an increase of 2.466 with respect to *Base-none*. When training or random subsets of the data, *Base-20%* and *Base-60%* show average performance increases of 1.931 and 2.298 respectively, showing less performance increase than when using whole corpus, as would be expected. With these baseline results for different training sizes in hand, we can now compare to our results using filtered versions of dataset with equal sizes — whose random variations have been suppressed by the triple run. These are supposed to represent a cleaner version of the dataset supposedly — as we hypothesized before starting the study — more relevant than random samples. We believe that can be observed for *LASER-60%*, where performance is higher than *Base-60%* (increase of 0.177) and even than *Base-all* (increase of 0.009), meaning that removing 40% of the data yielded increased performance. The case of *MUSE-60%* is also illustrative, since it only meant a decrease of 0.163 with respect to *Base-all*, and performance was higher than *Base-60%* by 0.005, which does not seem very significant. The case of *LaBSE-60%* is different, since it actually decreased the performance by 0.083 with respect to *Base-60%*, indicating its use was not benefitial.

JORGE: I am not sure if we should comment on MUSE as good result above, hat you think?

When comparing the smaller sizes of subsets of 20% of the data, none of the filtering methods helped obtain a model that was better than *Base-all*. However, we still see LASER and MUSE (in this case more noted) methods showing beneficial results when compared to *Base-20%*, specifically showing 0.246 and 0.203 increase respectively. Again, LaBSE does not seem to help, but rather to do the opposite. The same holds for BERT method.

Altogether, the relation seems to hold that $\mathbf{LASER_n > MUSE_n > Baseline_n > LaBSE_n}$ for n being the size of the subset. **LASER is the best performing filtering method**, since its subset of 60% shows even better performance than using the whole corpus. The latter is very significant, since by reducing by 40% your corpus, you can obtain even better performance and reduce your computing time by almost a half. Also, if you are able to sacrifice some performance, using methods like MUSE would give you better results than randomly sampling the same size. Results for LaBSE and BERT methods seems to suggest that we would rather randomly sample than using them as a filter. The latter are unexpected results for as, as we expected all methods to outperform or be equally good as the random sampling.

JORGE: I would comment here on why we think each method performed better than the others based on nature of the embedding maybe. Also, why BERT and LaBSE might be doing so badly

JORGE: we could comment too on the time

Filter	Model	Avg.	Min	Max
-	Base-none	14.936	-	-
-	Base-all	17.402	-	-
-	Base-40%	17.234	17.174	17.296
LASER	Filtered-60%	**17.411**	-	-
MUSE	Filtered-60%	17.239	-	-
LaBSE	Filtered-60%	17.151	-	-
BERT	Filtered-60%	-	-	-
-	Base-20%	16.801	16.516	17.041
LASER	Filtered-20%	**17.114**	-	-
MUSE	Filtered-20%	17.071	-	-
LaBSE	Filtered-20%	16.376	-	-
BERT	Filtered-20%	16.273	-	-

Table 2: Experimental Results

taken to do the embeddings, which is too much and performance not very much increased then maybe not worth it (specially BERT)

One possible line for future work to improve our analysis would be to manually evaluate the performance. For instance, we could select top k predictions of the models fine-tuned on filtered dataset and from the model fine-tuned on unfiltered dataset and randomly shuffle them. Then, a Polish speaker could evaluate the quality being blinded on the prediction model origin. Also, significance testing could improve further the interpretability of our results.

It must be noted that filtering was done using only cosine similarity. A possible next step would be to include more advanced distance metrics to compare the embeddings vectors and see if some is more benefitial than others. Also, filtering methods not solely based on comparing embeddings could be included. An even more interesting approach could be to use some filtering methods that do not only rely on quality, but also on how much in-domain the sentences are. We observed in our corpus that some sentences were not so specific to the medical domain, containing just a few medical terms and the rest being pretty general language. If we were able to filter based on how specific to the domain the language is, probably the performances would be even higher when testing on a very specific test set as ours.

All in all, ...

7 Concluding remarks

Here you succinctly summarize you work; what do your findings mean for the broader field? what are limitations of your work?

References

Github - facebookresearch/laser: Language-agnostic sentence representations — github.com. https://github.com/facebookresearch/LASER. [Accessed 10-12-2023].

Mbart50tokenizer. Accessed [2023.10.21].

Ufal medical corpus. Accessed [2023.10.21].

2023. Muse. https://ai.meta.com/tools/muse/. [Accessed 16-12-2023].

Haluk Açarçiçek, Talha Çolakoğlu, Pınar Ece Aktan Hatipoğlu, Chong Hsuan Huang, and Wei Peng. 2020. Filtering noisy parallel corpus using transformers with proxy task learning. In *Proceedings of the Fifth Conference on Machine Translation*, pages 940–946.

Fred Bane, Celia Soler Uguet, Wiktor Stribiżew, and Anna Zaretskaya. 2022. A comparison of data filtering methods for neural machine translation. In *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)*, pages 313–325.

Fred Bane and Anna Zaretskaya. 2021. Selecting the best data filtering method for nmt training. In *Proceedings of Machine Translation Summit XVIII: Users and Providers Track*, pages 89–97.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. *arXiv preprint arXiv:1906.08885*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Lei Cui, Dongdong Zhang, Shujie Liu, Mu Li, and Ming Zhou. 2013. Bilingual data cleaning for smt using graph-based random walk. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 340–345.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Ondřej; et al. Dušek. Khresmoi summary translation test data 2.0. Accessed 2023-12-09.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels. Association for Computational Linguistics.

Kaveh Taghipour, Shahram Khadivi, and Jia Xu. 2011. Parallel corpus refinement as an outlier detection algorithm. In *Proceedings of Machine Translation Summit XIII: Papers*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

Asahi Ushio, Yi Zhou, and Jose Camacho-Collados. 2023. An efficient multilingual language model compression through vocabulary trimming.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950.

A Example Appendix

This is an appendix.