Controllability for English-Ukrainian Machine Translation Based on Specialized Corpora

Daniil Maksymenko, Olena Turuta, Nataliia Saichyshyna, Maksym Yerokhin and Oleksii Turuta
Kharkiv National University of Radio Electronics / Nauky Ave. 14, Kharkiv, Ukraine
{daniil.maksymenko, olena.turuta, nataliia.saichyshyna, oleksii.turuta}@nure.ua

Abstract

Significant difficulty in translation tasks is usually caused by the possibility of having multiple correct results. That is where human translators usually beat modern machine learning models, as they have much more external context, which can be useful to create a correct translation both from the meaning and style sides.

The purpose of this article is to provide a possible solution for the lack of context during machine translation, which would provide an ability to increase the controllability of existing machine translation architectures. We propose a new architecture, which would incorporate this additional embedded context into the translation and compare this new approach to some classic ones like just transfer learning of some new features using an existing, trained model.

We conducted some experiments using the proposed architecture to check if it indeed allows controlling of the translation process and measured the new model using both token and embedding metrics.

1 Introduction

Usage of encoder-decoder architecture with different approaches like LSTMs or transformers allowed for achieving human-like translation (Sutskever et al., 2014). However, such models still cannot outperform professional translators with years of experience. Models can capture meaning, and they can translate some difficult terms or even ones they did not see during training, but usually, they work with a black box approach, when we just provide input text and wait for the result.

The most classic approach to change the model and its behavior is to make some fine-tuning or apply transfer learning to some existing architectures (Kocmi, 2020). However, we need a good dataset to make it work and tuning can take a long time, depending on the amount of available data, model size, and hardware.

Text generation models like T5, which can also be used for translation, allow us to add some special tokens or just descriptions of style or sentiment (Raffel et al., 2022). This approach should work well without any fine-tuning, as it is based on the zero-shot learning concept (Xian et al., 2018). However, a special token or short description can be not enough to significantly alter the result of the model. This method works much better with recent models like GPT 3 or ChatGPT, but they are available only as APIs and still make many errors in any other language than English, as they were trained for it originally (Brown et al., 2020). Some solutions propose adding a topic modeling result into the translation, but it also does not provide too many opportunities to affect the model (Eidelman et al., 2012).

In this work, we propose an architecture to get better controllability over the machine translation tasks by adding some external context in there, which can be obtained from another model. We provide examples of how our approach works, show the theory behind it, and provide some ideas for further development in this area. New architecture gets measured with both token and embedding metrics. It should be compared with some machine translation models trained during our previous research with the usage of transfer learning, so we can check if this new concept works better than the classic one.

2 Datasets

In the process of preparing the study, we reviewed, downloaded, and analyzed a large number of existing datasets for the Ukrainian language. Moreover, here we used datasets prepared and collected earlier by ourselves, which also contributed to the results. We paid attention to the collected data, its analysis, cleaning, and checking the accuracy since the data directly affects the results of the task. In addition, when solving the controllability problem, we must be sure that the data is unambiguously related to the declared domain. Four specific domains were collected, which are described below.

- **Common texts** compiled on the basis of the manual translation of the Multi30k dataset (Elliott et al., 2016). Covers general topics.
- **Scientific articles** are sufficiently large and informative translations of scientific articles with the appropriate scientific style.
- **Ukrainian laws** are certified translations of legislation intended for foreign organizations. The style of the texts is official.
- **Technical documentation** is guidelines for using a web application programming framework

The collection process and more information are described in the previous article (Maksymenko et al., 2022). The domains were chosen in such a way that they have distinctive styles of texts and the controllability of the resulting translations can be clearly traced.

In addition to the texts described above, which have an explicit style, we trained our model using large datasets for the Ukrainian language. These include OPUS datasets that contain datasets of hundreds of thousands of lines, but do not guarantee the correctness and exact correspondence of the English and Ukrainian translations (Zhang et al., 2020a). Because of this, the preparation of these datasets involved checking the cosine similarity, determining the source language, and more.

Initial processing means filtering out duplicates, empty lines, lines with incorrect values in the form of characters that do not carry semantic value. After that, the resulting sentences were processed using distiluse-base-multilingual-cased-v2 (Reimers and Gurevych, 2019) multilingual model in order to calculate the cosine similarity of strings to compare their identity in meaning. In this way, we were able to clean the existing datasets from mistranslations, "shifted lines" and semantic errors. A value of 0.4 was chosen as the threshold value for cosine similarity, which is considered sufficient to maintain semantic similarity between sentences. We also examined sentences that were beyond the cosine similarity line of 0.4. In most cases, the sentences were screened out fairly, but there were also cases where the sentences were in a figurative sense and were also marked as incorrectly translated.

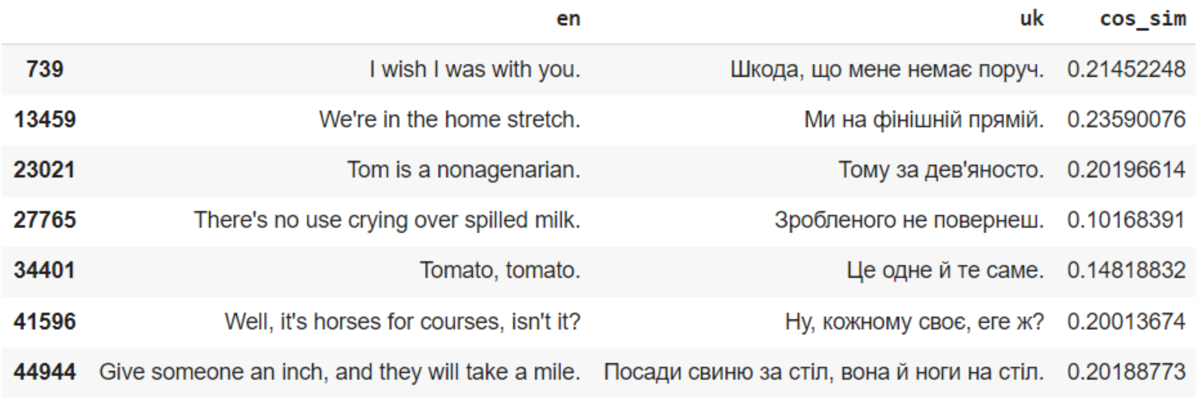
Such cases were found and were not excluded from the data sets, which gives us the opportunity to train the model to understand the figurative meaning. In this way, we prove that these metrics cannot be used as a standard benchmark, since they do not handle phraseological units and slang. We have achieved a large amount of clean data, which helped us to restore the decoder and became the basis for retraining the model on the specific domains described above.

3 Proposed solution

For the last decade translation models use an encoder-decoder architecture, which takes a vector of tokens, creates their embedding matrix, passes it through some recurrent or attention layers, and then creates a new vector of tokens from this original text embedding. As we mentioned before we can try to affect translation by using some special tokens to show the network the desired style or tone, but it does not give great results for controllability. Usually, good human translation is based on not only an understanding of both input and target languages but also on knowledge of a greater context of certain text, like having some good past examples, knowing events that are described in the text, and emotional and sentimental features in it. Modern neural translators can capture some of it by just getting fed with terabytes of data, but we still can not modify or tweak their understanding of the input. We can't interfere with the translation style without finetuning or we can just hope that adding some instruction or special tokens, will change the output of the model. A possible solution can be to use an idea proposed for instant voice transfer in text-to-speech tasks, like SV2TTS architecture (speaker verification to text-to-speech) (Jia et al., 2018)). This architecture uses an external model to create an embedding of the speaker's voice, which then gets merged with an embedding matrix of tokens sequence (each column of a matrix gets merged with this voice vector). This external model gets trained on some other tasks, like speaker verification, which allows it to learn necessary features, which can be transferred somewhere else later. We can use semantic search models in the case of machine translation as they learn the meaning and some stylistic features of texts, which allows us to put the original text in a vector space before translation and move it towards chosen domain in this space to change

Dataset name	Initial row count	Row count after initial processing	Row count after cos sim checking
OPUS-kde4-v2-eng-ukr	233 611	172 898	145 796
OPUS-multiccaligned-v1-eng-ukr	1 400 000	1 080 177	1 069 201
OPUS-opensubtitles-v2016-eng-ukr	612 127	486 564	427 355
OPUS-eubookshop-v2-eng-ukr	1790	725	497
Total	**2 247 528**	**1 740 364**	**1 642 849**

Table 1: OPUS datasets analysis.



	en	uk	cos_sim
739	I wish I was with you.	Шкода, що мене немає поруч.	0.21452248
13459	We're in the home stretch.	Ми на фінішній прямій.	0.23590076
23021	Tom is a nonagenarian.	Тому за дев'яносто.	0.20196614
27765	There's no use crying over spilled milk.	Зробленого не повернеш.	0.10168391
34401	Tomato, tomato.	Це одне й те саме.	0.14818832
41596	Well, it's horses for courses, isn't it?	Ну, кожному своє, еге ж?	0.20013674
44944	Give someone an inch, and they will take a mile.	Посади свиню за стіл, вона й ноги на стіл.	0.20188773

Figure 1: Figurative sentences

the output. Figure 2 shows us 2D projections of text embeddings obtained from Siamese BERT in miniLM implementation (Reimers and Gurevych, 2019). Here you can see how some texts start to form clouds based on topic, style, wording, and sentiment. For example, we can see how abstracts from scientific articles are getting close to some general texts, which can be explained by their attempt to describe something difficult with more casual terms to easily explain the main point of the article. Also, clouds for programming documentation and laws are distanced from all the other samples. Even within laws, we can see a few big groups, like laws that describe education or laws, which describe agreements. That can become a solution for the outer context problem in machine translation as we would provide not only tokens but also the position of the input text in this embedding plane described by the semantic and stylistic features vector. Also, we conducted some further research on these groups to prove that semantic search embeddings can be used to distinguish between different categories of texts, so we can affect the translation and help our network learn faster by using this external context. We created heatmaps of mean vectors for each category of texts to check how much they usually differ. In figure 3 you can see one small slice of this heatmap that shows how some parts of more serious texts like laws and acts tend to get more negative values. Their counterparts usually get higher values for the same features. Cases, where this distribution is opposite, are also possible, but all 4 vectors can still be distinguished well. Increasing or decreasing certain features in the initial embedding simultaneously to make it closer to some group of texts should allow us to save the original text features and add more information on a desired domain, style, and sentiment. For example, we put input text on the plane shown in Figure 2 among all the other texts. This text was used: “He came to the throne at the age of 73, an age when most people are thinking more about retirement than taking up a big and important job.”. It falls somewhere between articles and general ones, as it is part of an article, but does not contain any specific words or stylistic features. We will try to move it to the laws-like domain so that the translation should get written in a more official language. In order to achieve it let’s calculate the difference between each element of the input text vector and laws mean embedding. Then we will multiply these differences by a coefficient, which can be called a transformation power. It shows how much we want to move this text in a certain cluster

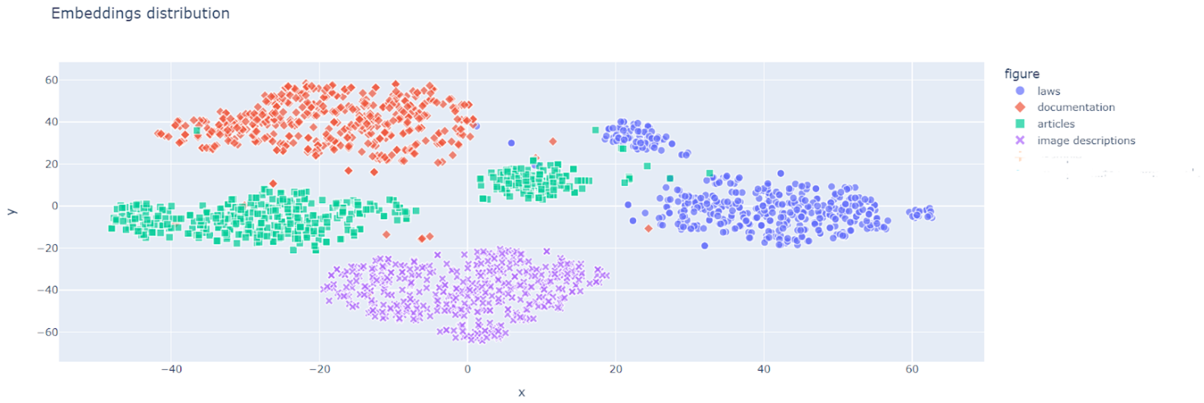


Figure 2: 2D projections of text embeddings.

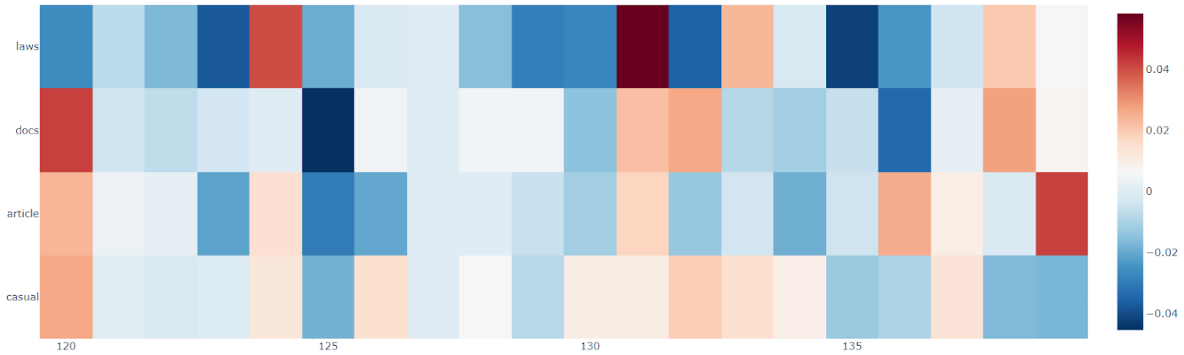


Figure 3: Slice of a heatmap of category mean embeddings

and how many changes should we apply to the original vector. Finally, we will subtract this multiplied vector of differences from the original embedding to create a new embedding skewed into a certain domain space. In figure 4 we show how the original text embedding (big yellow cross) gets moved into laws space more and more as we increase the transformation power by 1 starting with 1.5 (big bright blue circles are laws-transformed original text embeddings).

So our theory is that usage of semantic search embeddings should allow getting more control over the way the encoder-decoder model translates a text by showing it what the desired domain in a certain case.

4 Model architecture

We used huggingface transformers implementation of MarianMT as a basis for our model (Junczys-Dowmunt et al., 2018). It uses the BART interface and weights pretrained in the Marian C framework (Lewis et al., 2020). So original architecture can be described as an encoder-decoder model where both parts have 6 layers. Encoder can get up to 512 tokens and returns a matrix of embeddings with a dimension of 512x512. Siamese BERT in miniLM format was used in our modification to capture general text features. It gives us a vector with 384 values to describe a domain of the text, its meaning, and its style. This vector gets merged with each token embedding, so we get a matrix with a dimension of 512x896, which then gets reduced to the original 512x512 dimension using a fully-connected layer and SELU activation. This transformed matrix is used as an input for the decoder, so by modifying this semantic and stylistic embedding we can change the results of the model.

OpusMT English-to-Ukrainian model by Helsinki NLP was used as initial weights for the encoder and decoder in the modified architecture (Tiedemann and Thottingal, 2020).

5 Modified model training

Such a change of architecture would definitely affect the performance of the model and ruin the connection between the encoder and decoder, so the modified model would need massive tuning before further measurements and comparisons. In

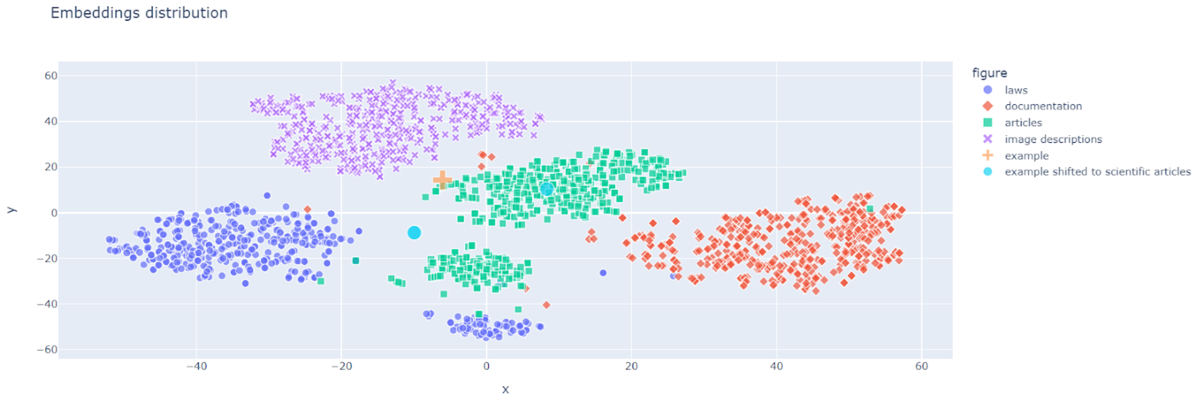


Figure 4: Example of shifting a text in a certain domain

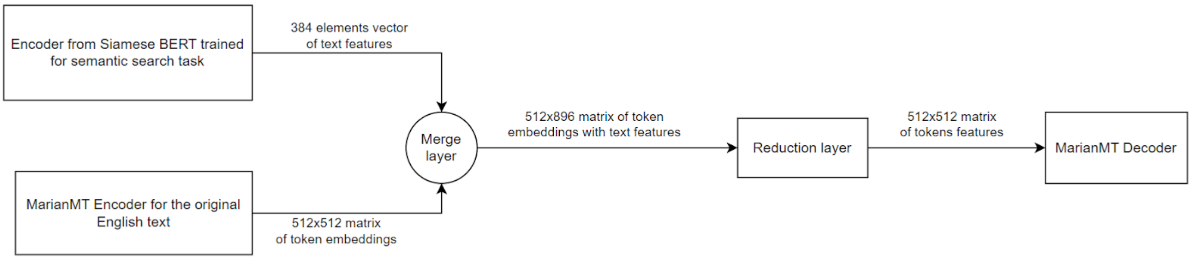

Encoder from Siamese BERT trained for semantic search task
384 elements vector of text features
MarianMT Encoder for the original English text
512x512 matrix of token embeddings
Merge layer
512x896 matrix of token embeddings with text features
Reduction layer
512x512 matrix of tokens features
MarianMT Decoder

Figure 5: Modified MarianMT architecture with external context vectors

order to restore the encoder-decoder connection we trained the modified architecture using previously described datasets (2 million texts) on a single Nvidia T4 GPU for 5 epochs, which took us around 34 hours to complete. A subset of our gathered multidomain texts was used as a validation set to measure the validation loss and metrics (the subset contains 25% of all gathered texts from general, law, and scientific texts). The best epoch gets saved and it will be used in further experiments. We used token-based metrics like BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) to measure translation quality and embedding-based metric BERT Score (Zhang et al., 2020b) to check if translation possibly has the same meaning but uses a different set of words or text structure than the ground truth value. This way we can compare the new model with our previous research. Model fitting and restoration of performance can be seen in the table below, which shows metrics values on our custom validation subset:

One of the most interesting details here is that embedding metric BERT Score did not show how bad the performance really was after modifying MarianMT architecture and before fine-tuning when some old and proved token metrics showed how much progress did the model do in those 5 epochs of tuning. If we use only BERT Score to judge the model, then we will most likely think that the performance is not critically bad. However, here is one example of how a ruined connection between the encoder and decoder affected translation quality. Here is the original English text: "He has to come back in the next movie", which should be translated to Ukrainian as "Він має повернутися в наступному фільмі" . Modified MarianMT before fine-tuning gives a translation, which is absolutely not related to the original: "Це означає, що ми маємо справу з іншими людьми, а не з ними". We consider that the bad performance of the BERT Score was caused by the model beneath it. Metric uses English BERT for English texts and Multilingual BERT for any other language like Ukrainian. Model gets trained on 104 languages and it was proved multiple times that it performs much worse than language-specific models like UkrROBERTA (Panchenko et al., 2022). So probably BERT Score was still able to obtain some similar token-embedding pairs in ground-truth and wrongly translated texts and it was enough to give an average score, even if the model was absolutely wrong. This proves that embedding metrics are still not ready to be used as main performance measures for machine translation tasks. In order to

Model state	**BLEU**	**METEOR**	**BERT Score F1**
Original MarianMT before modification	11.20	0.2807	0.8115
Modified MarianMT without tuning	0.02	0.0147	0.5859
Epoch 1 of tuning	28.45	0.4387	0.8848
Epoch 2 of tuning	32.50	0.4627	0.8935
Epoch 3 of tuning	34.22	0.4730	0.8977
Epoch 4 of tuning	35.09	0.4781	0.8998
Epoch 5 of tuning	**36.14**	**0.4830**	**0.9021**

Table 2: Metrics values on our custom validation subset.

finally confirm that the modified architecture is ready for use we compared it to a set of individually finetuned MarianMT models from our previous research. They were tuned using our gathered texts to check if we can achieve controllability of translation style and domain with a small set of data for low-resource languages, so there are 3 models tuned with laws, scientific articles, and image descriptions separately and 1 model tuned with all these samples. The comparison was based on validation results on our subset of multidomain texts, so it proved that we were able to restore the performance of our best model from the previous research and even surpassed it. Measurements can be seen in the table below:

Such an architecture should also ease tuning for new domains, as we can try to distinguish them by placing a text on the semantic embeddings plane before translating it. Once the model regained its original performance and even improved it, we can move to the experiments on controllability to check our theory about additional context vectors.

6 Experiments description

Now we can take some texts, which can be interpreted in multiple ways, and try to translate them with some modifications of the embedding vector. We will take the text “Give my money back” as a first example, as it can be translated straight forward or in a more serious or even mean way. First of all, we will just translate the text using the tuned model. The result is "Поверни мені мої гроші" , which is a correct translation, which would work in most cases. Let’s try to make it more serious and official. We will shift the embedding towards the laws text domain with transformation power equal to 1.5 in order to achieve it. New embedding allows us to get the following result: "Повертайте мої гроші назад". If we make the transformation power coefficient higher (like 5.5 for example) we can obtain the following results: "Повертайте мої гроші", which sounds like a short and official request. Also, we tried to move it closer to the documentation domain with coefficient 5.5, which gave us this output: "Віддати мої гроші". This translation does not look like something, which could be used in a real life, but it was still interesting to see how the network made the text sound like an instruction you could read in some manual. Here is the visualization of where the original embedding fell and where did the other vectors appear. Let’s take a look at another example: “Then, about seven years after the gold rush began, it finished”. Initially model gives a correct translation, which sounds like that: "Через сім років після початку золотої лихоманки все закінчилося". However, it lacks some stylistic features of the original text, but we can move the embedding closer to scientific articles to make it sound more like the original text. We will use transformation power equal to 3.5 and it gives this output: "Потім, близько семи років після початку золотої лихоманки, вона завершилася". This text sounds much closer to the English original than the first obtained one. Let’s show how the text moves deeper into the scientific articles domain. One modified embedding has power equal to 1.5 and moves to the articles cloud and our final embedding with 3.5 power moves somewhere in between laws and articles, which allowed us to get a better translation in the end.

Even if the model did not get any historical documents or descriptions of historic documents, it was able to use features it learned in other domains to form some understanding of how provided text should be translated to become closer to those historical documents.

Here is one more example of how translation controllability works in our model. We have the following English text: “What are you going to

Model	**BLEU**	**METEOR**	**BERT Score F1**
Laws-only tuned MarianMT	25.34	0.3861	0.8630
Science-only tuned MarianMT	18.88	0.3347	0.8448
Descriptions-only tuned MarianMT	12.70	0.3034	0.8380
All texts tuned MarianMT	34.16	0.4754	0.8983
Modified MarianMT with context vector	**36.14**	**0.4830**	**0.9021**

Table 3: Performance of models.

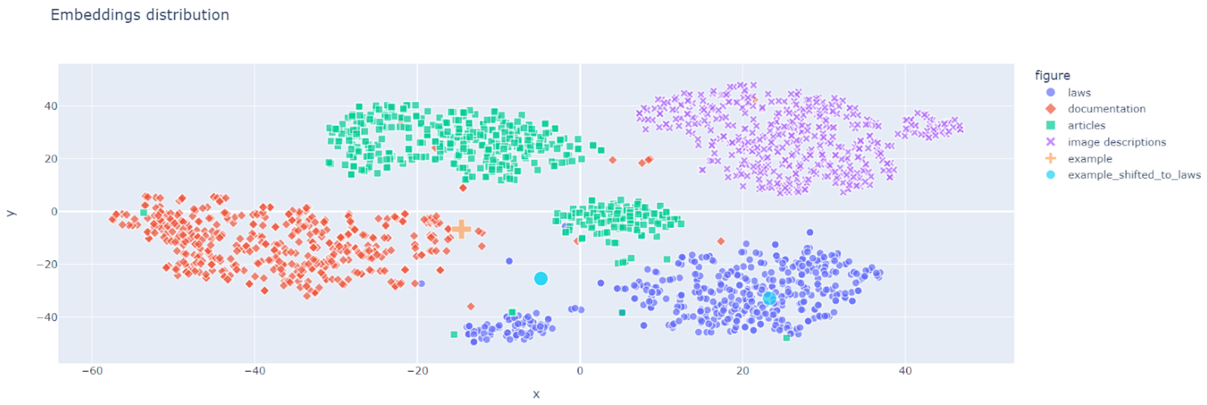


Figure 6: Change of the original text embedding towards laws

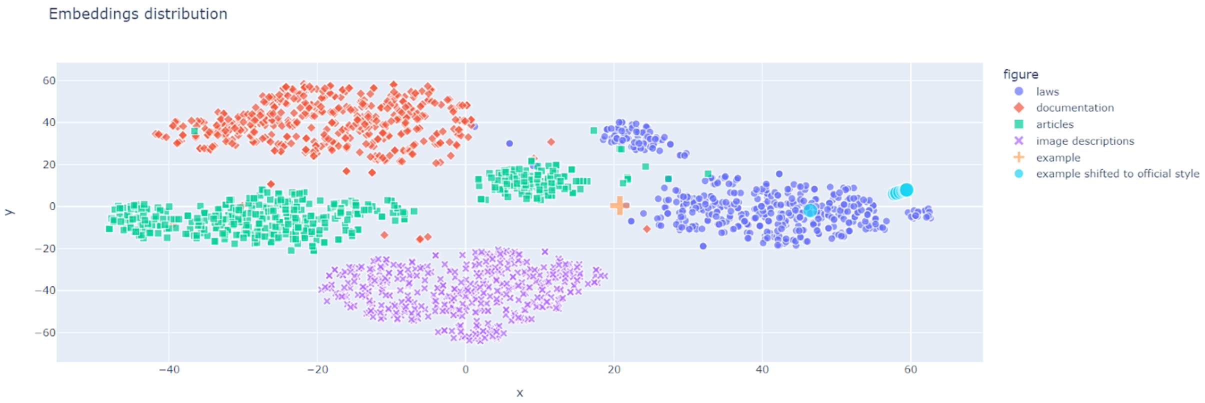


Figure 7: Text shifted to the scientific articles domain

eat with your sandwich?". It gets translated to Ukrainian like that: "Що ти їстимеш зі своїм бутербродом?". This translation is fine, but let's make it sound like a more modern speech (by moving it toward casual texts with power 6.5). The new text uses words, which are more expected from some modern kids and it sounds like that: "Що ти будеш їсти зі своїм сендвічем?". Not only did it change the translation of "sandwich", but it also changed the structure of the sentence to make it sound lighter.

So, this way we can make a translated text sound differently without some additional model finetuning or modifications. We just need to get a library of examples for different states, like historical texts, which use old words and phrases, laws, documentation, manuals, news, some jokes, or casual dialogues. Mean embedding vectors should be calculated for these categories. Then we can move a text feature vector toward chosen cluster and the model output should become more like it, which we were able to do in the examples above.

7 Conclusion

In this research we proposed a solution to achieve better machine translation controllability by ingesting some external context into the original text tokens embeddings. We modified MarianMT encoder-decoder architecture to combine the embedding matrix with a semantic search embedding vector of the original text to add more information about style, meaning, and sentiment. The new model was tuned to regain its original performance using 2 million texts from OPUS datasets and our own scrapped sets, which consist of multi30k image descriptions, laws translations, scientific articles abstracts, and programming framework documentation. The model was compared to the ones trained in our previous research, which tried to just tune the original MarianMT into mentioned domains using a small portion of data gathered for a low-resource language. New architecture outperformed all previous models and gave the ability to change translation by shifting the semantic embedding.

Further tests and experiments proved that the new model indeed allows us to change the style, certain words, and structure of the translation. We showed a few examples of how our solution works for different texts and styles. Also, the way to scale this model to support more styles without any significant fine-tuning was described. Our proposed model should just get enough examples of different desired styles in the original language without any translations to capture their features and try to transfer them to the translation. We want to increase the training dataset to improve our model performance as a further development. Also, we have another idea on how to modify the embedding vector to shift it closer to the necessary state. In theory, we could build a hyperplane from the original text embedding vector and target state vectors. Then this original text can be moved by this hyperplane to affect model output.

Limitations

The most significant limitation of our research is that we did not find a way to fully interpret obtained semantic and stylistic embeddings of texts. This would allow us to make the domain change algorithm easier and more conscious. We would change just some single features or areas of the embedding vector to provide some new characteristics, which we want to see in the output. There is still a plan to get more clear interpretations to improve developed algorithms. Another limitation is related to the lack of computing resources as we could pass more data, but that would take much more time on our configuration.

Ethics Statement

The team of authors supports and agrees with the accepted ethical rules, which, in our opinion, contribute to the development of scientific activity. Such principles increase communication between authors, significantly improving the quality of the results.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Vladimir Eidelman, Jordan Boyd-Graber, and Philip Resnik. 2012. Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 115–119, Jeju Island, Korea. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.

Ye Jia, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, and Yonghui Wu. 2018. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 4485–4495, Red Hook, NY, USA. Curran Associates Inc.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Tom Kocmi. 2020. Exploring benefits of transfer learning in neural machine translation.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Daniil Maksymenko, Nataliia Saichyshyna, Oleksii Turuta, Olena Turuta, Andriy Yerokhin, and Andrii Babii. 2022. Improving the machine translation model in specific domains for the ukrainian language. In *2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT)*, pages 123–129.

Dmytro Panchenko, Daniil Maksymenko, Olena Turuta, Mykyta Luzan, Stepan Tytarenko, and Oleksii Turuta. 2022. Ukrainian news corpus as text classification benchmark. In *ICTERI 2021 Workshops*, pages 550–559, Cham. Springer International Publishing.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Yongqin Xian, H. Christoph Lampert, Bernt Schiele, and Zeynep Akata. 2018. Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. *TPAMI*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020a. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.