Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1059–1068
DOI: 10.15439/2023F895
ISSN 2300-5963 ACSIS, Vol. 35

Controllability for English-Ukrainian Machine Translation by Using Style Transfer Techniques

Daniil Maksymenko,
Nataliia Saichyshyna,
Oleksii Turuta
0000-0003-3223-5130
0000-0002-5145-0015
0000-0002-0970-8617
Kharkiv National University of Radio Electrics
Nauky Ave. 14,
61165 Kharkiv, Ukraine
Email: {daniil.maksymenko, nataliia.saichyshyna, oleksii.turuta}@nure.ua

Marcin Paprzycki
0000-0002-8069-2152
Systems Research Institute, Polish Academy of Sciences,
ul. Newelska 6, 01-447 Warsaw
Email: marcin.paprzycki@ibspan.waw.pl

Mirela Alhasani
0000-0002-9110-394X
EPOKA University,
Tirana, Albania
Email: malhasani@epoka.edu.al

Maria Ganzha
0000-0001-7714-4844
Faculty of Mathematics and Information Science
Warsaw University of Technology
Koszykowa 75,
Warszawa, Poland
Email: m.ganzha@mini.pw.edu.pl

***Abstract*—While straightforward machine translation got significant improvements in the last 10 years with the arrival of encoder-decoder neural networks and transformers architecture, controllable machine translation still remains a difficult task, which requires lots of research. Existing methods like tagging provide very limited control over model results or they require to support multiple models at once, like domain fine-tuning approach.**

In this paper, we propose a method to control translation results style by transferring features from a set of texts with target structure and wording. Our solution consists of new modifications for the encoder-decoder networks, where we can add feature descriptors to each token embedding to decode input text into the translation with the proposed domain. In conducted experiments with English-Ukrainian translation and a set of 4 domains our proposed model gives more options to influence the result than some existing approaches to solve the controllability model.

***Index Terms*— Machine Translation, Controllability, NLG, Style Transfer.**

I. Introduction

LAST 10 years became very prolific for machine translation solutions as they finally achieved quality, which can be compared to a human processed results in many cases. The first significant step was the usage of recurrent encoder-decoder models, however, they were significantly overperformed by new attention-based transformer networks [1], which compare each part of the input sequence to multiple parts of input simultaneously. Their architecture allowed to reduce training time and made parallelization of the process easier compared to RNNs as they always need the n-1 state to compute the nth one. Pretrained models like mBART[2] gave the ability to capture low-resource languages much better than ever before.

However, these models usually do not give any methods to influence their results. We can get multiple options out of them or interpret their decisions by using SHAP [3] and similar frameworks, but we can't easily change the way these decisions are made. The easiest way to modify model behavior would be to finetune it using a small specialized corpus, but we need to support a whole model zoo for each separate domain or style to implement this approach, which can become expensive and difficult to manage.

The cheapest method to change at least some words in translation would be by applying usage dictionaries and finding another possible translation for a certain word or phrase, which could correspond better. This method will not allow us to modify translation according to a certain external context, we would just search for another option among popular ones.

Another approach proposes adding tags with style or other necessary features to influence the model. It should work well with both recurrent and bidirectional encoders as such tags are usually added at the beginning of the text, so their embeddings can further influence every step of translation generation [4]. This solution is not flexible enough as we can't encode all necessary features into a set of special markers and we do not know how the model will act if we change their order. Also, even a slight change in this marking would require us to completely retrain the model, which would be time-consuming and expensive.

Some new papers propose concatenating vectors with certain features like length, sentiment, officialness, or politeness to text embedding. However, we need to mark each translation with necessary feature values before training and any change in this marking or addition of a new feature would require a full-on retraining of such a model.

Modern generative models like GPT3 [5] and its next versions can conduct translation once they are given some examples even if they were not originally trained specifically for any other language than English. They still can generate some transliterations instead of translations or confuse related languages. Such models can be controlled via prompts, so we can try to add some statements like "make it more serious" to change the features of the translation. So in order to control such a model we need to pass some examples of the desired behavior and know the desired result to some extent.

In this research, we present an architecture based on transformer encoder-decoder models, which should conduct style transfer of a certain domain during translation by concatenating token embeddings with a text descriptor vector before decoding embeddings into the target translation. We provide explanations for this approach, comparisons with other available methods by both token and embedding metrics, and example translations generated by the proposed model.

II. *Datasets*

As our main aim was to increase machine translation models controllability by using transfer learning techniques we needed to gather some domain-specific and datasets, which contain texts of a certain style and structure. Styles should be distinct to enrich the model with knowledge in as many different types of texts as possible. We prepared 4 small specialized datasets with English-Ukrainian pairs with the following styles:

- general texts, which contain photo descriptions from the Multi30k dataset [6], which was translated by our team, and the results were presented in our previous paper;
- official texts, which consist of laws translations gathered from the Verkhovna Rada of Ukraine website [7];
- scientific texts use abstracts from Ukrainian papers gathered from Google Scholar service;
- programming documentation sentences, which were gathered from the official Vue framework website.

More information on these datasets is available in our previous papers. They describe mining, transformations, and cleaning for those specialized corpora.

We targeted sentence granularity for all text pairs, however, pairs in some domains contain one compound Ukrainian sentence and some simple English corresponding ones. Such behavior was spotted mostly in the scientific domain (abstracts from Ukrainian papers). We left them as they were written without splitting them. Other pairs with multiple sentences, which could be split into multiple ones, were transformed into 2 or more pairs of sentences.

Another large set of texts we used was gathered from multiple OPUS corpora [8]. They contain book reviews, subtitles, TED talk transcriptions, etc. They contain lots of messy data, which can even harm the model performance. As an example, there are a lot of texts with incorrect translations or translations, which can be understood only in a full original text context. Some texts contain some scrapping leftovers like tags or links. Many entries propose translations not in Ukrainian but in other similar languages, but it can be useful to learn some similar grammar or words, especially when the target language is a low-resource one. Another group of task is NER in low-resource languages [9].

There are around 60 million text pairs for the English-Ukrainian language set in OPUS, but we used only 2,247,528 texts. Cases like links or tags were cleaned using Python libraries, but we still needed to clean some incorrect translations. We could not check even this small chunk of OPUS manually, so we needed to automatize meaning comparisons of original and translated texts. Siamese XLM-R [10] for the semantic search was used to accomplish this as it supports both English and Ukrainian. We encoded each text into a vector with 512 elements and calculated the cosine similarity to its Ukrainian counterpart. The model was initialized from the **distiluse-base-multilingual-cased-v2** [11] checkpoint from the huggingface hub. This checkpoint was trained by using the Knowledge Distillation method. After checking multiple pairs and their cosine similarities we decided to use 0.4 as a threshold value. Pairs, which have lower similarity scores, are considered to be incorrect.

Some examples which have a value lower than 0.4 were examined. Most of them were really bad translations or missed some crucial part of the original text to understand why they should be translated this way. However, one corpus had lots of phraseologies, which were scored as errors by the XLM-R model as it tried to understand them in their literal sense. As an example, the phrase "murder will out" was translated as "правда всплиє". This is a correct translation, but the score is less than 0.4 as the model does not understand figurative sense. Such cases were not deleted from datasets and were used during model training as such cases can be really useful and hard to learn correctly.

The removal of texts with only links, tags, or empty lines and the removal of incorrect pairs reduced the dataset from 2,247,529 texts to 1,642,849 ones. Table 1 shows the number of pairs in each corpus and assigns a certain domain to each one except OPUS sets.

It is worth noting that OPUS corpora were used only in one step of the conducted experiment. Other specialized sets were used at each step of the experiment. Also, we split 25% of gathered specialized corpora into a test subset, which contains 9,625 text pairs with all 4 mentioned domains.

III. *Proposed Solution*

We propose a further development of the previously mentioned technique with the concatenation of the vector of target features to the input text tokens embeddings. As it was described before this method used vectors with a fixed set of features like length, sentiment, etc. We propose to use semantic descriptors of text, which can be combined with de-

TABLE I.
DATASETS OVERVIEW

Dataset name	Domain	Number of text pairs
Subset of OPUS corpora	-	1 642 849
Laws translations from Verkhovna Rada of Ukraine website	Official	4 000
Scientific articles abstracts from Google Scholar	Scientific	2 000
Vue framework documentation	Documentation	2 500
Photo descriptions from Multi30k	General	30 000
Total	**-**	**1 681 349**

scriptors of domains to conduct style transfer from a certain domain to the input text translation.

Architecture would use a pretrained encoder and decoder and all the changes would happen after the creation of token embeddings and before their decoding into the target language translation. This way we can use an already established model as a foundation for our new solution instead of training the MT model from scratch, which would require millions of text pairs, lots of computational resources, and time.

Semantic descriptors of texts can be obtained from an external pretrained model trained for semantic search task. It would return a vector descriptor of the input text, which would allow us to place it into a certain embedding space and compare input to other texts the model has previously seen. Similarity to other texts can point the model towards the usage of certain words and styles as it can find suitable examples of target translations in this embedding space. It means that the descriptor does not carry any information about the style or features of translation output, but it shows the model text pairs which can be used as examples of necessary behavior as their descriptors are similar to the input descriptor. Fig. 1 shows 2D projections of semantic descriptors obtained from the semantic search model (in this case it was siamese BERT).

X and Y here are values generated by TSNE to reduce vectors from 384 elements to just 2, which we can easily visualize as a scatterplot. Texts from all 4 domains form multiple clusters and even subclusters based on their meaning, usage of words, sentiment, and tone. That is exactly what we need as further we can point the model toward one of these subclusters to gather translation features out of it and pass them to the decoder.

We need not only the input text descriptor to control the translation process but also domain descriptors. We will consider the average vector descriptor of all texts in a certain domain as a descriptor of this domain. In the next formula, we show the calculation of each element of the domain descriptor.

Fig 1. 2D projections of text semantic embeddings

$$V_{mean\ domain_j} = \frac{\sum_{i=0}^{N} V_{i,j}}{N} \quad (1)$$

So in this approach, we need to combine original text descriptor and this domain descriptor. We propose to do it by conducting a linear combination of the original text descriptor and vector of difference between text and domain. It is shown in the following formula, where is a transformation power coefficient, $V_{original}$ is an embedding vector of original input text, $V_{mean\ domain}$ is a mean embedding vector of texts in certain domain and descriptor is the final vector, which provides context on the way the text should be translated:

$$difference = V_{original} - V_{mean\ domain} \quad (2)$$

$$descriptor = V_{original} - \alpha * difference \quad (3)$$

Our hypothesis is that usage of semantic search embeddings should provide more control over the way the encoder-decoder model translates a text by showing it the desired domain and putting the text among ones with similar features. The transformation power *coefficient* α should indicate the power of changes, which we want to make and how much should descriptor be shifted into a certain embedding subspace.

However, the concatenation of the vector to each row of the token embeddings matrix will change its form. Let's say that matrix of token embeddings has the form NxM, where N is the number of tokens and M is the dimensionality of the embedding. Let's mark the size of the semantic descriptor as K. After concatenation our token embedding matrix will have the form Nx(M+K). Such a matrix would be impossible to pass into the original decoder as it still expects just an NxM matrix. We either need to create our own decoder and train it from scratch or create a dimension reduction layer, which would reduce the new concatenated matrix to its original size, so we can use a pretrained decoder. However, even the second option still requires some tuning as we add a new raw layer, which will make values in the matrix different from the initial ones. We would lose the connection between the encoder and decoder, so it has to be restored by tuning a new dimension reduction layer, so it would make new em-

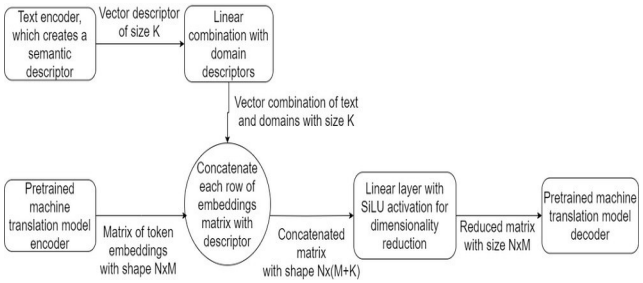

Fig 2. Diagram of proposed machine translation model

beddings closer to the original ones and incorporate new knowledge obtained from the semantic search model.

Fig. 2 shows the architecture of a proposed machine translation model with style transfer abilities.

Currently we implemented dimension reduction as a linear layer with the number of output features equal to M. We used SiLU activation with batch normalization after the linear unit. We see the usage of Variable Selection Networks as another good option for this task, which we want to explore in our further research. This layer proved its effectiveness for classification tasks and time series feature selection, so we would like to see if it preserves such quality in the case of embedding dimensionality reduction to pass only the most significant values to the decoder for each token.

We want to use transformed semantic search embeddings of texts (combined with a certain domain embedding) concatenated to each token embedding as an additional mapping to give the decoder a hint of the necessary translation style, words domain, and tone, which we want to get. These vectors will not carry the style features themselves and they will not be hardcoded in there with a certain allowed range like in other similar approaches. They should be generated by an external model and linearly combined with a domain descriptor vector to shift features into the necessary cluster and make the overall vector closer to the texts with the desired style in the feature space. Semantic embeddings should only point toward texts with a translation style similar to the one we would like to achieve.

This way we can use any pretrained encoder-decoder machine translation model as a foundation for this architecture. Then we need to choose an external model to obtain sentence embeddings for texts and add a concatenation step for each row of the token embedding matrix to add values from sentence embedding at the end of each row. The final modification step would be to add a dimensionality reduction layer, which would restore the original dimension of token matrix rows, so we can use the original pretrained decoder. This process is shown in Fig.3 with an example where we have 512 feature embeddings for tokens and 384 number sentence embeddings.

The modified model will still need some fit as a new layer will not be trained at all, which would cause wrong translations due to the decoder getting previously unseen values. The only change for the train and validation datasets would be the need to calculate sentence embeddings.

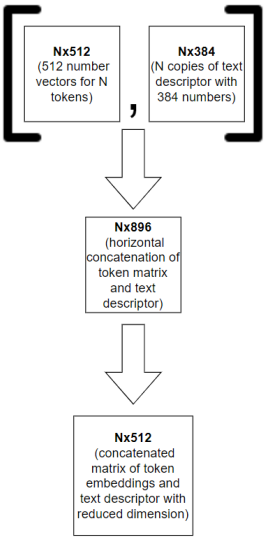

Fig 3. Semantic search embeddings injection process

The perfect case for this architecture would be to transfer style from a single provided example, but we should check this hypothesis. The primary use case would still be to combine text and domain descriptors to modify translation generation.

IV. *Metrics*

We use both token and embedding metrics in this research to measure the performance of obtained solutions. BLEU [12] was chosen as a default machine translation token metric and METEOR [13] was chosen as it works better with morphologically rich target languages due to the usage of stemming and synonyms dictionaries during scoring.

As for the embedding metrics we decided to use BERT Score [14] as it proposes a method to measure text generation quality by measuring semantic similarities of token embeddings obtained from the BERT model. So this way we would be able to compare texts by both their structure using token metrics and meanings by embeddings.

Generated examples of controllable translation [15] should be scored as well, however, we do not have references for all possible modifications of final translations, so mentioned token and embedding metrics will not have any chance to measure the quality of results as they need benchmarks in the target language. We will use a siamese XLM-R trained for multilingual semantic search to measure cosine similarity of original English text to each new generated Ukrainian translation. We will use a siamese XLM-R trained for multilingual semantic search to measure cosine similarity of original English text to each new generated Ukrainian translation. Model will be initialized from clip-ViT-B-32-

multilingual-v1 [11] checkpoint, because it was trained to replicate image domain embeddings, which should be useful for model, so it can gather knowledge not only from text information but also from pictures.

V. *Model Zoo*

Our main model for experiments will be MarianMT [16] pretrained for English-Ukrainian translation by Helsinki University on OPUS corpora. We used implementation from huggingface transformers library, which uses the BART interface. The model has 6 layers bidirectional encoder, like in BERT models, and 6 layers autoregressive decoder, like in GPT models. In this research we use this model with the following modifications to achieve controllability:

1. MarianMT fine-tuned with a small specialized corpus to capture its style, structure, and common words. We create a separate version of the model for each domain;
2. MarianMT tuned with all gathered specialized corpora to check how cross-domain knowledge can help the model learn the language better and if it still would be able to distinguish styles;
3. MarianMT with the addition of a special token-marker of necessary style at the beginning of the input text without fine-tuning (for example: "[official] Ukraine is a sovereign and independent, democratic, social, law-based state");
4. MarianMT with the addition of a special token-marker of necessary style at the beginning of the input text tuned on full specialized corpora to use all the advantages of bidirectional encoding and autoregressive decoding to better distinguish provided domains;
5. MarianMT modified with our proposed solution (concatenation of text-descriptor on each token embedding and dimensionality reduction for the obtained matrix to pass it into the original decoder). As it was mentioned before we would need to train the dimensionality reduction layer to restore the connection between the original encoder and decoder, so that is where we are going to use cleaned OPUS corpora. We will train this model using both our specialized datasets and OPUS data.

So we train variants 1, 2, and 4 only on 4 small specialized datasets. Version 3 will not be fine-tuned at all and version 5 gets trained with both OPUS and our datasets, as it needs to teach a new layer from scratch and learn how to use text descriptors for domain adaptation. We train each model with a fixed budget of 36 hours on Nvidia T4 GPU and then compare them by whole test dataset results and on separate domains in it to measure the controllability of obtained solutions.

Text descriptors will be gathered from siamese BERT for semantic search initialized from all-MiniLM-L6-v2 [17] checkpoint trained by the sentence-transformers team. It returns vectors with 384 elements, while MarianMT encodes each token in a vector with 512 elements. So after the concatenation of the token and text descriptor, we will have vectors with 896 elements. It means that the dimensionality reduction layer should reduce the size from 896 back to 512 elements, so we can pass the results to the original, pretrained decoder.

VI. *Experiments. Comparing models on full test dataset*

First of all we will train separate models for each domain and one, which would receive all gathered, specialized corpora. All these models should be scored on the full version of the test dataset. Table 2 shows the results of the training.

The best score on full test dataset was achieved with the model, which got all specialized corpora, which is expected as this model saw every style features. The second place is occupied by the model, which was trained with official texts (laws translation), which can be explained by the difficulty of this specific domain. Sentences there contain a lot of specific, uncommon words and phrases or even common words with new senses. The structure is strict and differs significantly from other styles.

We expected higher results from the scientific domain as it can also provide some unique knowledge to a model, which would not be possible to retrieve from other groups of texts. However, it gets lower scores in all 3 metrics than the official domain model. As we mentioned earlier this domain contains some difficult cases, where English text consists of multiple simple sentences and its Ukrainian counterpart has just one big compound sentence. It can confuse the model and also makes it difficult to calculate token metrics correctly.

TABLE II.
SEPARATE MODELS FOR EACH DOMAIN

№	Model variant	BLEU	METEOR	BERT F1 Score
1	Original OPUS MT MarianMT	11.20	0.2807	0.8115
2	MarianMT tuned with general texts	12.70	0.3034	0.8380
3	MarianMT tuned with official texts	25.34	0.3861	0.8630
4	MarianMT tuned with scientific texts	18.80	0.3347	0.8448
5	**MarianMT tuned with all special corpora**	**34.16**	**0.4754**	**0.8983**

Original OPUS MT MarianMT and version tuned with general texts have the lowest and quite similar scores. Photo descriptions from Multi30k did not give any new insights as they mostly consist of simple sentences with just a subject, an action, and sometimes a brief description of the environment. It does not differ from the original OPUS corpora, which contain lots of general domain texts too.

The next step is to check how a special token marker would affect the model performance. Table 3 shows scores for the model, which gets such markers without any fine-tuning, and for the fine-tuned version, where each text is marked with a style tag.

Special token without any fine-tuning does not give any significant boost to the original model scores and even makes the BERT Score worse. However, a tuned version of MarianMT, which learned how to use such tokens, overperforms even the previous best result obtained with MarianMT tuned with all specialized corpora without any tags. As we said before this special tag can influence every other token embedding and the decoder behavior due to their bidirectional and autoregressive natures respectively. The model distinguishes styles better and still learns information from all of them simultaneously.

The final step would be to teach our proposed model. We were able to teach it for 5 epochs with our set budget. Fig. 4 shows the metrics plot for each epoch and compares them to the previous best solution (MarianMT tuned with all specialized corpora with style tags).

We scaled BLEU to the 0-1 range in this plot to place all plots in the same subspace. The model completely loses the ability to translate after modification of the addition of concatenation with semantic descriptor and dimensionality reduction layer. The new embedding matrix has the same shape as the original one, but the values are not matched with what the decoder was getting earlier. Token metrics show it really well as they become almost equal to 0. However, BERT Score still gives average scores, which can be a huge misdirect if we do not calculate token metrics simultaneously. The model just generates random Ukrainian texts without any connection to the original English one. For example, our input is “Laws have been around for over 4000 years”. Generated translation without any fine-tuning was “Це означає, що ми маємо право вирішувати, що робити, а що ні.”. The model completely lost the ability to translate and embedding metrics were not able to capture it properly. The results of this experiment proved that BERT Score can not be used as a single metric to measure translation quality as it should be accompanied by some classic token approaches. It can be explained by the usage of multilingual BERT as an encoding model, as we compared it to other models for the Ukrainian language in previous papers. It completely loses to ones like XLM-R, so the default implementation of the BERT Score does not work well as a benchmark for Ukrainian language text generation tasks.

TABLE III.
SPECIAL TOKEN-MARKER MODELS

№	Model variant	BLEU	METEOR	BERT F1 Score
1	Original OPUS MT MarianMT	11.20	0.2807	0.8115
2	MarianMT tuned with all special corpora	34.16	0.4754	0.8983
3	MarianMT with special token without tuning	11.72	0.3086	0.8085
4	**MarianMT with special token tuned**	**37.08**	**0.4923**	**0.9019**

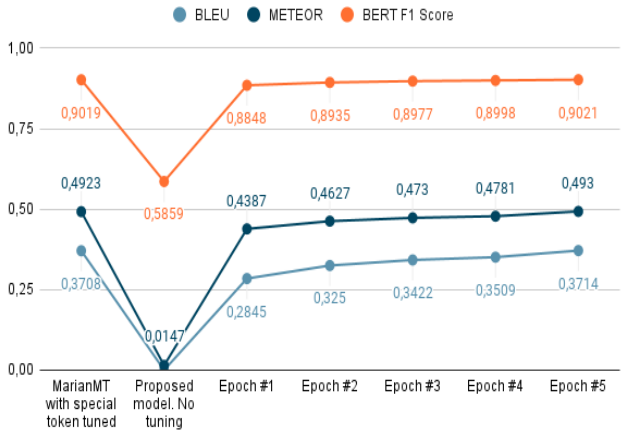

Fig 4. Training plot of proposed model

5 epochs of training with both OPUS corpora and our specialized sets were enough to restore the connection between the encoder and decoder of MarianMT and incorporate new knowledge obtained from semantic embedding space. Our model was able to overcome the previous best results scored with MarianMT tuned on all specialized corpora with special tokens. It achieved BLEU equal to 37.14, METEOR 0.4930, and BERT F1 Score 0.9021 on the full test dataset.

This first part of the experiment proves that our model can generate translations on the same level as some established controllable translation solutions. Now we need to check models on different domains included in the test set to check how all models distinguish different styles and to check if our model is able to beat separate specialized models for each domain.

VII. *EXPERIMENTS. MEASURING CONTROLLABILITY*

We measured each metric for 3 domains individually for each model to compare their performance and to understand how well the proposed model distinguishes domains. The desired result for our proposed model would be to perform on par with specialized models for each domain or at least get a close score. We will start with official texts. Scores can be found in Table 4.

The proposed model gives better results for all 3 metrics in comparison to all other models and most importantly it overcomes model trained only for the official style translations. It surpasses 50 by BLEU, which indicates that it is capable to provide fluent law translations. As it was said a few times before this style contains lots of difficult cases like uncommon words or strict structure of the sentence. Metrics show that model with provided semantic descriptors was able to capture these cases well enough.

TABLE IV.
COMPARISON BY OFFICIAL TEXTS DOMAIN

Model variant	BLEU	METEOR	BERT F1 Score
MarianMT tuned with official texts	49.48	0.6044	0.9247
MarianMT tuned with all special corpora	48.60	0.5987	0.9239
Original OPUS MT MarianMT	08.06	0.2444	0.7778
MarianMT with special token without tuning	9.10	0.2764	0.7862
MarianMT with special token tuned	51.93	0.6141	0.9285
Modified MarianMT with semantic descriptors	**53.25**	**0.6473**	**0.9303**

Measurements on general texts retain high quality too, as the proposed model still gets BLEU higher than 50. It overcomes all other models here and gives a significant boost to the translation quality. Also, it is interesting to see how a special token without tuning made results only worse for the initial model here as it was already trained to deal with general texts and here it gets an unknown entity, which only creates more errors in comparison to the target translation. Both models trained on all specialized corpora lost to the model trained only on general texts. Even style marker did not help the model distinguish other styles from image descriptions well enough to beat the specific model.

The results for general texts are represented in Table 5 (in this case image descriptions from Multi30k).

The last ones are abstracts from scientific articles. Results are represented in Table 6.

The proposed model works better than other ones for this domain too, however, the translation quality is still low. Such BLEU indicates that it still makes significant errors. METEOR and BERT Score show that probably the model still tries to replace original constructions with similar, synonymous ones. Such a decline in performance was probably provoked by the mentioned difference in the structure of English and Ukrainian counterparts. Also, it is interesting how the model tuned with all texts overcomes the one with abstracts only, which can be explained by a high diversity of abstracts in terms of their topics. Additional texts provide the model with more knowledge of some less-represented subdomains. It can be seen in Fig. 1, where a subset of scientific texts divides into 2 categories. We clustered these texts additionally and obtained 3 large clusters, which mainly can be described as articles about laws in spheres of economics and education, mechanics articles, and ones about biology and chemistry. The number of clusters was found via the silhouette method. Fig. 5 shows 2D projections of sentence embeddings of scientific texts clustered into 3 categories (where blues points are laws articles, green ones are about mechanics, and red ones are about chemistry and biology). So laws domain should help with the first

Fig 5. Scientific articles clusters

TABLE V.
COMPARISON BY GENERAL TEXTS DOMAIN

Model variant	BLEU	METEOR	BERT F1 Score
MarianMT tuned with general texts	42.40	0.4083	0.9181
MarianMT tuned with all special corpora	40.06	0.3948	0.9128
Original OPUS MT MarianMT	22.90	0.3264	0.8599
MarianMT with special token without tuning	22.022 3	0.3730	0.8428
MarianMT with special token tuned	40.89	0.4029	0.9164
Modified MarianMT with semantic descriptors	**53.46**	**0.5301**	**0.9264**

TABLE VI.
COMPARISON BY SCIENTIFIC TEXTS DOMAIN

Model variant	BLEU	METEOR	BERT F1 Score
MarianMT tuned with scientific texts	21.93	0.4127	0.8495
MarianMT tuned with all special corpora	23.42	0.4291	0.8548
Original OPUS MT MarianMT	10.94	0.2710	0.7956
MarianMT with special token without tuning	10.89	0.2736	0.7952
MarianMT with special token tuned	25.22	0.4523	0.8648
Modified MarianMT with semantic descriptors	**26.64**	**0.4686**	**0.8618**

TABLE VII.
EXAMPLES OF CONTROLLABLE TRANSLATION

№	Original	Translation	Modifications of text descriptor	Cosine similarity score (original to translation)
1	acquire ownership of intellectual property rights	набуття права інтелектуальної власності	Absent	0.9714
	This translation is close to the original meaning, but does not fully stick to the law language style. The issue here is that both “ownership” and “rights” can be translated as “власність” here, which would be a more correct translation. “право” and “власність” can be used as a more general style options.			
2	acquire ownership of intellectual property rights	набути право власності на інтелектуальну власність	Linear combination with official domain (=3.5)	0.9748
	Here we get “ownership” translated as “право власності”, which would be a more correct translation of this sentence according to laws terms. So model correctly used one word to translate two English ones in the same sentence and preserved the official style.			
3	I began asking the students themselves to compile multiple translations of a single poem for class presentation.	Я почала просити студентів скласти кілька перекладів одного вірша для презентації класу.	Absent	0.9810
	Just a correct translation with general style words used.			
4	I began asking the students themselves to compile multiple translations of a single poem for class presentation.	Я почав просити самих студентів скомпілювати кілька перекладів одного вірша для презентації класу.	Linear combination with a subset of a casual domain where the subject is male (=4.5)	0.9849
	Same style, but model captured the change of subject gender in provided examples and changed words forms (like “почав” instead of “почала”).			
5	I began asking the students themselves to compile multiple translations of a single poem for class presentation.	Я почав вимагати від студентів складання декількох перекладів єдиного вірша для презентації класу.	Previous modification + official domain with =3.5	0.9854
	Official style made the request “began asking” sound more as a requirement (“почав просити” became “почав вимагати”). English-like word “скомпілювати” got replaced by Ukrainian original word “складання” and the form became closer to English passive voice.			
6	In case you broke something you must pay for this	Якщо ти щось зламав, ти повинен заплатити за це	Absent	0.9812
	Just a normal translation with correct meaning			
7	In case you broke something you must pay for this	Якщо ви розбили що-небудь, ви повинні відповісти	Linear combination with official domain (=7)	0.9683
	Combination with official domain, which mostly consists of juridical documents and laws, made “pay for this” translation sound more like “carry responsibility” instead of paying money, which could be used one of more interpretations			
8	why don’t you come sit down with me?	чому б тобі не присісти зі мною?	Absent	0.9766
	Straightforward, correct translation			
9	why don’t you come sit down with me?	чому ви не приєднаєтеся до мене?	Official domain with coefficient 5.5	0.9137
	“you” gets translated as a more formal address and the sentence gets interpreted as “why don’t you join me”, which could be one of the possible translations depending on a larger context			
10	Do you want to hear a dirty joke? Ok. A white horse fell in the mud.	Ви хочете почути брудний жарт? Гаразд. Білий кінь впав у грязюку.	Absent	0.9692
	-			
11	Do you want to hear a dirty joke? Ok. A white horse fell in the mud.	Хочете почути грязну анекдоту? Гаразд. У грязюку впав білий кінь.	Old literature domain with coefficient value 4.5	0.9720
	By combining the source text with old literature we change the word “жарт” into “анекдота”, which would be an outdated way to translate “joke”. This word can still be used, but more as a joke genre name.			
12	Excuse me. Do you know the way to the zoo?	Вибачте, ви знаєте шлях до зоопарку?	Absent	0.9715
	This translation is correct and would be understood by a native Ukrainian speaker, but it copies the structure of the English source instead of adapting it.			
13	Excuse me. Do you know the way to the zoo?	Вибачте, ви знаєте, як пройти до зоопарку?	Casualness domain with coefficient 5.5	0.9702
	Here we get a more correct adaptation of “Do you know the way to the zoo” phrase, which would be a more common way to build this phrase in Ukrainian.			

cluster and technical documentation can provide more insight into terms from the second cluster, which can also be seen in Fig. 1, where a subset of scientific articles gets placed right between laws and documentation.

These measurements proved that our proposed model is able to translate 3 different domains with high quality and it distinguishes their features well by using information obtained from semantic descriptors. So these vectors can really help the model find enough examples of necessary translations among learned examples and they can be used to control translation style.

VIII. *Controllability examples*

We created 4 domain descriptors to test the proposed model. Each one of them was calculated as a mean embedding vector of texts corresponding to each domain. We encoded only English input texts, so the model searches for pairs close to the descriptor vector and uses their features to decode embeddings with the proposed style.

- Casual domain was calculated from 1000 image descriptions from the Multi30k dataset;
- Official domain was calculated from 1000 laws sentences;
- Instruction domain was calculated from 1000 documentation sentences;
- Old literature domain was calculated from 1000 sentences gathered from English literature from Project Gutenberg.

So now we have 4 domain descriptors with 384 elements each. We conducted some experiments on controllability to find optimal values of the transformation coefficient . Values below 3.5 usually do not change output text at all or change it slightly (as an example the only change can be the form of a single word). However, values lower than 3.5 can be used when we use multiple domains at once. Values higher than 7 shift descriptor values too much, so most of them become more than 1 or less than -1. It breaks the decoding process, so we get just a single word repeated as many times as the maximum number of output tokens allows or we just get some random symbols.

We show some examples of controllable translation in Table 7.

The model still can make some significant errors during style transfer. For example, we caught some errors with high coefficient values for the official domain. If we set it to 6.5 or higher it transforms some texts too much and literally changes their meaning. The input text was “Excuse me. Do you know the way to the zoo?”. Translation with official domain and coefficient equal to 7.0 was “Вибачте, чи знаєте ви шляхи до участі у виборчому окрузі”. In our opinion, it could be solved by using just one example of the desired style, so transfer could happen without setting of transformation coefficient.

We tested mentioned style transfer without transformation coefficient or creation of domain descriptors. We just pass another text as an example of the desired style and create its descriptor. It is then passed to the model instead of the input translation text descriptor. However, currently, the model does not make any changes based on just one example. It translates the text as if nothing was passed at all. In our opinion model needs more tuning to start working in a one-shot learning mode and transfer style from just one example instead of a whole set.

So the model can be used for controllable machine translation task but needs some precalculated domain descriptors to transfer the style of certain text set.

IX. *Conclusion*

In this research we proposed a solution to increase the controllability of machine translation models by using style transfer. We proposed a modified encoder-decoder architecture, which concatenates text semantic descriptor to each token embedding before decoding it into the target translation. This way we can point the model towards texts with necessary features, which we want to transfer into the final translation. The proposed solution was compared to established approaches like domain fine-tuning and the addition of a style marker by token and embedding metrics. Models were compared on a full multi-style test dataset and on each style separately. Examples of style transfer from a set of references were provided and a hypothesis for working in a one-shot learning mode was tested. Currently model needs more tuning to transfer style from just one example.

During our experiments, we tested the proposed solution only for 3 domains for English-Ukrainian translation. Also, we chose the optimal values range for the transformation coefficient by checking the changes after tweaking its value. The proposed model can be tuned further to learn new domains better. This solution can be scaled to a larger number of languages by changing the external model, which generates semantic descriptors.

As a further development, we propose to tune the model enough to finally run it in a one-shot learning mode. Also, we would like to interpret semantic descriptors in more detail to provide better control over text features and get a better understanding of each value influence. It can be done by using the sparse embeddings approach. Also, we would like to further modify the proposed architecture by trying other semantic encoders or changing the structure of the dimensionality reduction module.

Acknowledgment

This article is based upon work and STSM from COST Action Multi3Generation (CA18231), , supported by COST (European Cooperation in Science and Technology).

References

[1] A. Gillioz, J. Casas, E. Mugellini and O. A. Khaled, "Overview of the Transformer-based Models for NLP Tasks," 2020 15th Conference on Computer Science and Information Systems (FedCSIS), Sofia, Bulgaria, 2020, pp. 179-183, doi: 10.15439/2020F20.

[2] M. Lewis, ‘BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension’. arXiv, 2019.
[3] S. M. Lundberg and S.-I. Lee, ‘A Unified Approach to Interpreting Model Predictions’, Advances in Neural Information Processing Systems, 2017, p. 30.
[4] Neural Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability and Learning Erdem, Kuyu, Yagcioglu, Frank, Parcalabescu, Plank, Babii, Turuta et al. Journal of Artificial Intelligence Research 73 (2022) 1131-1207. https://doi.org/10.1613/jair.1.12918
[5] T. B. Brown et al., ‘Language Models are Few-Shot Learners’, arXiv [cs.CL]. 2020.
[6] Saichyshyna N., Maksymenko D., Turuta O., Yerokhin A., Babii A. and Turuta O. 2023. Extension Multi30K: Multimodal Dataset for Integrated Vision and Language Research in Ukrainian. In Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP), pages 54–61, Dubrovnik, Croatia. Association for Computational Linguistics.
[7] https://zakon.rada.gov.ua/rada/main/en/llenglaws
[8] R. Hanslo, "Deep Learning Transformer Architecture for Named-Entity Recognition on Low-Resourced Languages: State of the art results," 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS), Sofia, Bulgaria, 2022, pp. 53-60, doi: 10.15439/2022F53.
[9] J. Tiedemann, S. Thottingal, ‘OPUS-MT -- Building open translation services for the World’, Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, 2020, pp. 479–480.
[10] A. Conneau *, K. Khandelwal *, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzman, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov Unsupervised Cross-lingual Representation Learning at Scale
[11] N. Reimers and I. Gurevych, ‘Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks’, in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, 11 2019.
[12] M. Post, ‘A Call for Clarity in Reporting BLEU Scores’, Proceedings of the Third Conference on Machine Translation: Research Papers, 2018, pp. 186–191.
[13] S. Banerjee, A. Lavie, ‘METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments’, Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, 2005, pp. 65–72.
[14] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, ‘BERTScore: Evaluating Text Generation with BERT’. ArXiv, 2019.
[15] D.Maksymenko, N.Saichyshyna, O.Turuta, O.Turuta, A.Yerokhin, and A. 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.
[16] M. Junczys-Dowmunt, ‘Marian: Fast Neural Machine Translation in C++’. arXiv, 2018.
[17] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2