Can Machine Translation preserve language patterns?

1st Semester of 2025-2026

Vlad Olăeriu
vlad-mihai.olaeriu@s.unibuc.ro

Radu Ionescu
radu.ionescu2@s.unibuc.ro

Toma Nadu
toma.nadu@s.unibuc.ro

Abstract

Modern Neural Machine Translation (NMT) systems are fluent, but do they preserve the subtle style of a text? In this paper, we quantify this through a classification task, that shows how well translation models keep the distinct patterns of news genres, specifically Opinion, Reporting, and Satire, or if, on the contrary, these models flatten nuances into generic speech. Using the *SemEval 2023 Task 3* dataset, we employ the *M2M-100* model to translate English articles into five languages: French, German, Spanish, Polish, and Russian. We then fine-tune a multilingual *XLM-RoBERTa* classifier for each translation and compare its performance against the English baseline. This paper details our pipeline, the handling of imbalanced data, and the results of using genre classification as a proxy for translation quality. (Tan et al., 2020)

1 Introduction

Despite the significant progress in Neural Machine Translation (NMT) that has been achieved with the advent of Large Language Models, translations are still lacking in subtler, more intricate properties. Preserving language-specific patterns remains one of the most difficult tasks in computational linguistics. Modern models excel in fluency and grammatical correctness, but often fail to capture the invisible intent and structure of a language, such as the contextual meaning and cultural subtext, which define natural human speech.

Since the availability of corpora differs dramatically between languages, the performance of NMT models differs just as starkly. It is therefore expected for language patterns not to be preserved equally among languages. (Tan et al., 2020)

Defining ways of measuring the degree of preservation is critical, and also a creative endeavor. A good measurement should be able to allow us to perform comparisons among languages and among different models.

One such measurement, described in the term project which we have chosen, is a straight-forward classification task. For this, we choose one of the *SemEval* tasks, which is part of a real-world research workshop. (Patsesakis et al., 2023)

After training a classifier on the provided dataset (for a single language) and performing inference on a validation split, we will record the scores and mark them as the baseline. Then, we can use any MT model to translate that specific dataset, using the same train-validation split, into any other language supported by the classifier. Finally, we run the classifier on the translated datasets, take note of the results and compare them against each other and against the baseline. This way, we can quantify just how good each model in regards to any other model, and generally, in regards to the baseline.

The reason we chose this project specifically is that we considered that it implies an important theoretical implication, that we should aim at building MT models that preserve the source languages as much as possible. There is a need for a way to measure the preservation rate.

Thoughts and contributions

Vlad Olăeriu

Overall, I think it was a very interesting experiment to do, and I wish we had allocated a little more time towards the project, to address some of the limitations of this first experiment and to be able to complete the second experiment as well.

My contributions were:

1. doing analysis on the dataset we sourced, such as generating figures on sample distributions, top unigrams and bigrams per class

2. when *Google Colab* and *Kaggle* failed on us, I teamed up with Radu to find an alternative (*vast.ai*)

3. the general implementation of the translation pipeline

4. researching and finding an appropriate translation model

5. experimenting with different strategies to overcome the small context window of the model, by splitting samples into chunks

6. researching issues regarding repetitions generated by the model and its tendency to summarize towards the end

7. evaluation of translations (via *COMET*)

8. the general implementation of the classification pipeline

9. researching and finding an appropriate classification model

10. experimenting with various data balancing techniques, in order to overcome dataset imbalance

11. researching and applying evaluation models for the classification model

12. writing parts of the LaTeX documentation

Radu Ionescu

Even though it's my first contact with the field of Machine Translation, I think I have learned a lot about the overview of how modern transformer-based MT models work. It was a very pleasant project to work on.

What I achieved and helped with:

1. finding an alternative to *Google Colab* and *Kaggle* with Vlad

2. researching ways of evaluating the translations produced by the model we chose (*COMET*)

3. writing large parts of the LaTeX documentation

Toma Nadu

It is my first time tinkering with Machine Translation, too, and I found it very thought-provoking. Now I feel that I understand a bit more about the domain generally, and specifically about how the translation model we used works (very high-level).

Some of the things I did are:

1. after failing to contact the organizers and obtain the original dataset, I searched it on alternative platforms, and eventually sourced it (or a version of it) from a "third-party"

2. interpreted some of the data analysis performed by Vlad further

3. researched for models we could have used in the second experiment (comparing generations of MT architectures), even though we ended up leaving the experiment itself for future work

4. writing large sections of the LaTeX documentation

2 Approach

The particular task we chose as measurement is the *SemEval 2023 Task 3, Subtask 1* [1], which is concerned with the classification of news article as one of the following: opinion piece, reporting, and satire piece.

2.1 The dataset

The dataset provided by the organizers consists of samples from 5 languages *(English, Italian, Polish, French, German)*, on topics such as COVID-19, climate change, abortion, migration, the Russo-Ukrainian war, and specific events such as elections, etc.

Even though we requested the original dataset by contacting the organizers through email, after a brief response from them asking what our intentions and justification were for requesting the data in the first place, there was no further response. This is why we resorted to a public, sort of "leaked" dataset on GitHub[2], but we could not verify if it matched the original one.

[1] https://propaganda.math.unipd.it/semeval2023task3/

[2] https://github.com/nitanshjain/news_genre_classification_semeval_2023/tree/main/semeval2023task3/subtask1/data

We chose to use the English subset in particular based on a few convincing criteria. Firstly, due to the general corpora imbalance, most MT models perform best when translating from or to English, and we are interested in measuring the best possible outcomes of preserving language patterns. Secondly, the dataset is not particularly large, especially for Transformer-based models, and the subset of languages other than English is even smaller:

1. the designated English training dataset is roughly 3 times as large as the next largest subset
2. the test and validation subsets for English is still roughly 10% larger than the next largest subsets

The proportions reported above are considered in terms of the subsets' raw size on disk.

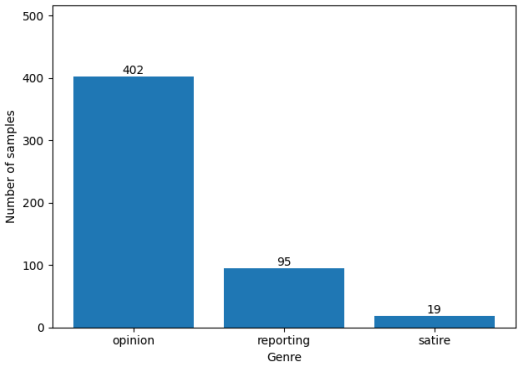

Figure 1: Class distribution of samples in the original English dataset

As we can see in Figure 1, the class distribution of samples is, unfortunately, very unbalanced, consisting mostly of opinion pieces, with very few satire pieces.

This issue is exacerbated by the fact that not only are there more articles in the class of opinion pieces, but there are also significantly more words per article on average in that class, as can be seen in Figure 2. The disparity is quantified in about 30% more opinion pieces than satire pieces, and about 15% more opinion pieces than reporting articles.

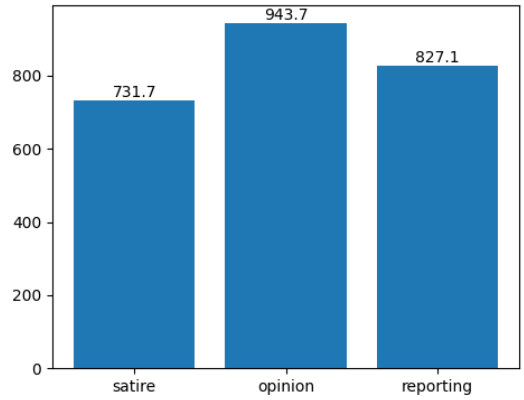

Figure 2: Class distribution in terms of average words per article in the original English dataset

It is also noteworthy to mention that we merged the individual *title* and *body* fields of each sample, as we found no reason to keep them seperate.

2.2 Execution Environment

We initially tried using free Google Colab or Kaggle computing instances for running our notebooks, but they were too slow or buggy. Then, we learned of a company offering affordable preconfigured containerized environments with pretty serious hardware for the money (*vast.ai*). An instance with an RTX 5090 attached was sufficient for our needs, and that is what we used in the end.

There was also the option of exposing ports of the container, so that we could access the Jupyter Notebook interface hosted by our instance remotely. (Vast.ai Inc., 2024)

2.3 Resources and Implementation

We published our source code inside a GitHub repository [3], which includes all the notebooks used for exploratory data analysis, translation, and classification, as well as the documentation and graphs. We implemented the pipeline using **Python 3.12** in the GPU-accelerated environment mentioned in Subsection 2.2 (vast.ai-RTX 5090). The primary libraries utilized include:

- **Hugging Face Transformers** and **Accelerate** for model fine-tuning and inference.
- **PyTorch** for backend tensor operations.
- **Scikit-learn** for computing precision, recall, and F1 metrics.

[3] [https://github.com/VladWero08/mt-pattern-preserve]

- **Pandas** and **Seaborn** for data manipulation and visualization.

2.4 Translation Model

For the translations, we have used a model from the **M2M-100** family models, developed by Meta AI, made available on Hugging Face on the original English dataset, and chose 5 high-resource target languages to make our translations into: French, German, Spanish, Polish, and Russian.

2.4.1 Model Architecture

The `facebook/m2m100_418M` model is a multilingual encoder-decoder (sequence-to-sequence), which was notable for being the first many-to-many machine translation model capable of translating directly between 100 languages without relying on English as an intermediate language. (Fan et al., 2020)

The 418 million parameter version is the smallest variant, optimized for a balance between performance and computational efficiency. In terms of model dimension, it is 1024, with a vocabulary size of 128000 tokens, to accommodate subword units from all 100 languages.

Unlike traditional translation models which are "English-centric" (i.e., in order to translate from French to Chinese, we start with French → English, and then English → Chinese), *M2M-100* uses *Language Indicators* for direct translation:

1. **Source Language ID:** A special token (*__en__*, *__fr__*, *etc.*) is prepended to the input to "communicate" to the encoder the language it is processing.

2. **Target Language ID:** The decoder is forced to start generating tokens with a specific target language token.

2.4.2 Setup & Inference

Because the size of the context window of this particular *M2M-100* model is relatively small, sitting at 1024 tokens, and the articles in our dataset are rather large (in fact oftentimes larger than the context window itself), we had to split each sample into chunks that could then be processed individually by the model and stitched together into a unified translation at the end.

At first, we tried small chunks of 512, 256, and 128 tokens. The smallest worked: chunks got translated, but the source text cut off mid-idea. Because we were already losing context by using a smaller window, the text meaning was split between chunks. Based on this finding, we switched to a sentence tokenizer to form chunks.

We fed each sentence to the model, truncating any sentence that is larger than 512 tokens, as we thought that the translation of a sentence would probably amount to a similar number of tokens. We haven't verified this hypothesis, since a sentence hitting this limit would be extremely large (word count in the hundreds).

We stuck with the default configuration of the model mostly, making sure to change the number of "paths" used in the *Beam Search* to 5, so that we could get potentially better results than simply going the greedy route (a single path).

Overall, translating from the original English dataset to another language took about 1 hour per language, which was reasonable.

2.4.3 Evaluation

Before classifying articles, we assessed translation quality using reference-free COMET (Rei et al., 2020), the `Unbabel/wmt20-comet-qe-da` model. The scores were averaged for each set of translated articles, and then a sigmoid function was applied over the mean. It resulted in Polish having the best score, followed very closely by Russian. We were intrigued to find out if this quality score of the translation will influence which language preserves the patterns of English the best.

Language	COMET
French	0.3829
German	0.4156
Spanish	0.4213
Polish	**0.4514**
Russian	0.4504

Table 1: COMET Scores

2.5 Classification Model

To evaluate whether the stylistic nuances of the original text survive the translation process, we employ a text classification approach using **XLM-RoBERTa** (Cross-lingual Language Model - RoBERTa). (Conneau et al., 2019)

2.5.1 Model Architecture

For our experiment, we utilize the `xlm-roberta-large` version, which consists of approximately **560 million parameters**. This architecture shares a vocabulary and vector space

across languages, allowing us to utilize the exact same architecture for English, French, German, and other target languages.

2.5.2 Data Balancing

Given the significant class imbalance (where 'Satire' is underrepresented), we employ two strategies to prevent the model from biasing toward the majority class:

- **Oversampling:** During dataset creation, we explicitly duplicate samples labeled as 'Satire' 3 times to increase their frequency.
- **Class Weighting:** We calculate the inverse frequency of each class and pass these as weights to the loss function, penalizing the model more heavily for misclassifying minority classes.

Other oversampling methods were also in our sight, such as back-translation with multiple intermediate languages, to generate more samples for the satire class, and maybe even for the reporting one. However, this would also imply more computing power, so it remains an idea for future work.

2.5.3 Experimental Setup

To assess pattern preservation, we implement an **Independent Fine-Tuning** pipeline. We train a separate, fresh instance of the classifier for each language dataset. This ensures that the performance on the translated text is not influenced by previous exposure to the original English data.

2.5.4 Training Process

The experimental workflow proceeds as follows for each language:

1. **Initialization:** We load a pre-trained *XLM-RoBERTa Large* model, resetting all fine-tuned weights.
2. **Fine-Tuning:** We fine-tune the model on the training split of the current target language.
3. **Evaluation:** We evaluate the model on the validation split of that same language to record the performance metrics.

The training process was highly efficient due to the relatively compact size of the dataset. Fine-tuning for 5 epochs took approximately 1 minute and 5 seconds per language.

2.5.5 Hyperparameters

We fine-tune the model for 5 epochs using a batch size of 16, the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.99, \epsilon =$ 1e-08, weight decay of 1e-4 and a learning rate of 2e-5. We utilize a Weighted Cross-Entropy Loss function to optimize the model.

2.6 Results

2.6.1 Evaluation Metrics

We evaluate performance using **Macro-averaged Precision, Recall, and F1-scores**. By comparing the Confusion Matrices of the original English text against the translated versions, we can visually quantify the specific misclassification patterns introduced by the translation process.

2.6.2 Quantitative Analysis

Table 2 summarizes the evaluation metrics across the original English text and the five translated versions.

Language	Precision	Recall	Macro-F1
English	0.7395	0.7700	0.7407
French	0.8023	0.7063	0.7172
German	0.7684	0.7607	0.7257
Spanish	0.7984	0.6671	0.6510
Polish	**0.8175**	**0.8174**	**0.8135**
Russian	0.8285	0.7999	0.7820

Table 2: Macro-Averaged Metrics

The results present a nuanced picture of pattern preservation. The Spanish translation resulted in the highest information loss, with the F1-score dropping to 0.65. Conversely, the Polish and Russian models outperformed the English baseline. This counter-intuitive result suggests that for these languages, the M2M100 translation may have simplified the vocabulary or sentence structures in a way that made the distinct genres (Satire vs. Reporting) easier for the classifier to separate linearly, even if the "human" nuance was lost.

Another worth mentioning observation is that the two highest scoring languages, Polish and Russian, correspond to the highest translation quality scored assessed by COMET previously in 2.4.3. Which means that the quality of the translation influences the preservation of the source language.

2.6.3 Visual Validation

To validate the reliability of our baseline, we analyzed the Confusion Matrix for the English dataset

(Figure 3).

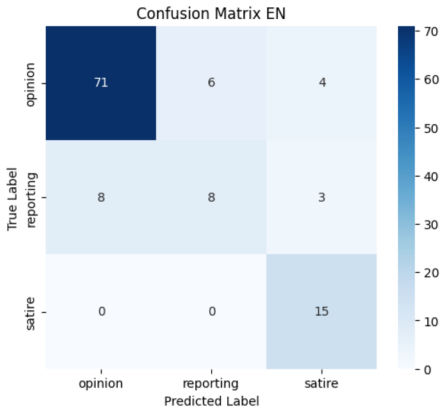

Figure 3: Confusion Matrix for English Classification

The matrix confirms that our strategies for handling class imbalance (Oversampling and Class Weighting) were successful. Notably, the model achieved a **100% recall rate for the satire class** (15 correctly classified, 0 missed), but it confused some opinion and reporting pieces as satire (7 in total). Maybe a smoother weight classing or less oversampling would improve this issue.

However, misclassifications were primarily confined to the boundary between opinion and reporting, which share more semantic similarities than the distinct genre of satire.

Compared to English, the top-performing translated language Polish shows identical recall on satire articles but lower misclassification rates, yielding higher accuracy for both opinion and reporting pieces, yet still struggles to differentiate between the latter two. (4).

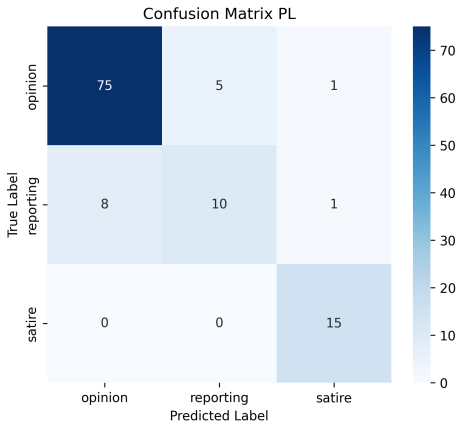

Figure 4: Confusion Matrix for Polish Classification

3 Limitations

As described in Subsection 2.1, the dataset we chose is severely unbalanced, skewing heavily towards the opinion class. This has probably introduced some amount of bias in the classifier models, resulting in all of our trained models more frequently inferring that any given article belongs to this class, than to any other. As such, the scores obtained by each such model on the generated translations might not be as representative of the degree of pattern preservation as we would have hoped.

Moreover, since our approach (detailed in Subsection 2.4.2) revolved around translating sentences individually, the quality of the final translations has probably suffered, because there was no way for the model to correlate subtler patterns spanning multiple sentences. The main reason for this is that we simply had no access to superior hardware which would be able to run larger models and doing so reasonably quickly. The model we did have access to simply couldn't accommodate such large samples. Another consequence of this limitation which we have observed during inference is the tendency of the MT model to initially produce a high fidelity translation, only to then summarize the original content when translating towards the end, as the model reaches the full capacity of the context window. This phenomenon, we've learned, is called *Low Scalability to Long Text*.

One final limitation we have encountered is related to another experiment, which is left as a possible direction for future work. The experiment, explained in much more detail in Section 4, involves running older, non-transformer based MT architectures, such as SMT or convolutional NMT. It seems exceedingly difficult to find *off-the-shelf* models or methods like those we have enumerated which do not require extensive tweaking, additional implementation or customization.

4 Conclusions and Future Work

In retrospect, we could have chosen a dataset which is much more balanced, so that the bias of the classifier would be reduced.

Initially, there were 2 experiments which we intended to do, but due to time and technical constraints, we were only able to complete the first one, which is explained in detail in this paper.

The other experiment involves comparing different MT model architectures: Phrase-Based Statistical MT, Rule-Based MT, Neural MT with CNNs,

RNNs, and Transformer architectures, with the hypothesis that newer models, more precisely the Transformer-based one, achieve better language preservation. Essentially, we would choose a single target language, translate the English dataset to that language using each model, train and run a classifier for every model, and finally compare the performance of the classifiers, as we have already done in our first experiment for the different target languages.

Another interesting future initiative is measuring the quality of language pattern preservation from source languages other than English, to various other languages. We could define a meta metric consisting of a single number that would reflect the overall degree of preservation from one base language to others, and then compare base languages based on it.

Keeping the limitations we have stated in mind, our findings point out that modern MT models have greatly improved in terms of preserving language patterns.

Working on this project was enjoyable, and has definitely shown us that there are a lot of creative directions in which experimental research can be taken, which we did not think about before.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation.

Antonios Patsesakis, Alberto Barrón-Cedeño, Firoj Alam, Giovanni Da San Martino, and Preslav Nakov. 2023. SemEval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multi-lingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2258–2277, Toronto, Canada. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online. Association for Computational Linguistics.

Zhixing Tan, Shuo Wang, Zonghan Yang, Gang Chen, Xuancheng Huang, Maosong Sun, and Yang Liu. 2020. Neural machine translation: A review of methods, resources, and tools. *AI Open*, 1:5–21.

Vast.ai Inc. 2024. Vast.ai: Gpu rental marketplace. `https://vast.ai/`. Cloud GPU Infrastructure.