Lt2326 project
Neural Machine Translation from Ancient Greek to English with Transformer Architecture

Ourania Kolovou

October 2024

1 Background

Machine Translation (MT) is an essential part of Natural Language Processing (NLP) that focuses on translating words, phrases, and complete texts between languages. Neural Machine Translation (NMT) has gained popularity due to its promising results and relatively simple design. It requires more computer resources and data than previous approaches such as Statistical Machine Translation (SMT). High-resource language pairings, such as English-Chinese, have produced outstanding BLEU ratings, particularly using models like Google's Transformer [1]. Low-resource couples, such as Ancient Greek-English, continue to face issues since Ancient Greek is complicated, with diverse syntax, a large vocabulary, and numerous dialects throughout historical periods. The language also offers challenges because of idioms and uncommon phrases in modern English.

The current project is based on my thesis on Machine Translation from Ancient Greek to English: Experiments with OpenNMT [1]. This thesis aimed to build a Neural Machine Translation model for an Ancient Greek-English low-resource language pair and capture the rich and complex source language morphology, syntax, vocabulary, and dialects. There have been a lot of experiments mostly with RNN (LSTM) architectures. The OpenNMT framework [2] was used for building the models. The model that performed best was an RNN-based model with a one-layer encoder-decoder. Despite the modest scores in metrics, including a BLEU score of 8 and METEOR of 0.35, the model exposes limitations in capturing morphosyntactic, semantic, and pragmatic details.

In this project, I will use some of the thesis data to build a machine translation model with Transformers architecture. The question is if the Transformers architecture will have better performance than the RNN architecture and if can capture better the morphosyntactic, semantic, and pragmatic details of the source language.

2 Data Resources

The data were collected for the thesis purposes from Digital Perseus Library [2] and Opus [3] \Tatoeba [4]. The whole dataset used for the thesis consists of 55266 a parallel corpus of Ancient Greek and its translation in English. All the data were already aligned in sentences or paragraphs, but those from Perseus Digital Library were in XML format, resulting in very long sentences up to 2000 tokens per line. The data were cleaned to achieve the appropriate text format for the training process.

[1] https://gupea.ub.gu.se/handle/2077/81765

[2] https://www.perseus.tufts.edu/hopper/

For this project, 9068 sentence pairs were selected up to 20 tokens per line. The data were shuffled and split into training (80%), testing (10%), and validation (10%) sets. In the tables below it can be seen more information about the data used for the training process.

Corpus	Sentences	Tokens
Ancient Greek	9068	116082
English	9068	171055

Table 1: Overview of the corpus

Datasets	Sentences	Tokens A.Greek	Tokens English	Percentage
Training	7253	92719	136993	80%
Testing	908	11718	17207	10%
Validation	908	11645	16855	10%

Table 2: Training, Testing, and Validation sets

3 Methods

3.1 Data pre-processing

The data in the XML format were converted into plain text using Python programming for the thesis purposes. The code used can be found on my Github repository [3]. The process involves five steps: In the first step, the XML data is parsed with "BeautifulSoup" library, cleaned, and saved in XML format. In the second step, the data is reformated, the "< div >" tags are adjusted, and the extra spaces are removed. In the third step, the text is extracted from XML, all HTML tags are removed and the data are saved as plain text. In the fourth step, the text is cleaned further by removing leading spaces, reducing blank spaces, and eliminating empty lines. In the final step, the cleaned text is loaded into a pandas DataFrame to ensure the proper order of sentences and filenames.

3.2 Tokenization

Subword tokenization was performed in the parallel corpora. It is an approach popularized by Sennrich [5], which extracts subword units instead of treating words as separate entities. This technique improves the representation of complex words, named entities, and loanwords. Below is an example of the next before and after subword tokenization.

(1) from ('My, cat, killed, a, squirrel.') to ('My, c@@, at, killed, a, squ@@, ir@@, rel@@, .')

3.3 Model

I trained a Transformer encoder/decoder model where the hyperparameters were almost identical to the base model described in Vaswani [6]. There were some differences though such as the number of training steps: the model was trained for 4000 steps to prevent overfitting in this small dataset, with a batch size of 4096 tokens on a single GPU. The model was built and trained using the OpenNMT framework [2]. OpenNMT is an open-source NMT application with various architectures and configurations. The training process took approximately 40 minutes.

[3] https://github.com/RaniaKol/Thesis

Hyperparameters	Values
Architecture	Transformer
Number of Layers	6
Encoder Hidden Size	512
Decoder Hidden Size	512
Batch Size	4096
Batch Type	Tokens
Vocabulary Size	16000
Learning Rate	2
Dropout Rate	0.1
Optimizer	Adam
Heads	8
Transformer_ff	2048
Tokenization	BPE

Table 3: Hyperparameters of the Model

3.4 Evaluation

The model was then evaluated against the testing data with the BLEU (Bilingual Evaluation Understudy) [7] and METEOR (Metric for Evaluation of Translation with Explicit Ordering) [8] scores.

3.4.1 BLEU

BLEU (Bilingual Evaluation Understudy) assesses and compares a candidate translation to a reference translation by calculating the overlap of n-grams (word sequences). It counts the number of n-grams that match between the candidate and reference, despite where they appear in the text. More matches imply a more accurate translation. The BLEU score ranges from 0 to 100, a higher score indicates that the generated translation is closer to the reference.

3.4.2 METEOR

METEOR (Metric for Evaluation of Translation with Explicit Ordering) is a translation assessment technique that compares a candidate translation's accuracy and fluency against the reference's. It assesses word and phrase similarity, considering word order and grammatical structure. The METEOR score is generated using precise matches between terms in the candidate and reference translations, resulting in a deeper evaluation than strictly n-gram-based approaches. The METEOR score ranges from 0 to 1; the higher the score, the better the quality of the translation.

4 Results and Discussion

A vocabulary size of 16,000 subwords was utilized. The BPE method combines the most frequently appearing pairs of bytes in a text corpus until a predefined vocabulary size is achieved. This technique will continue until the required vocabulary size is reached. Smaller vocabulary size provided a significant issue since it resulted in an increased frequency of unknown terms in the generated output.

Since I don't have a baseline model I will compare the project's model results with the thesis RNN model. The RNN architecture can be seen below:

Hyperparameters	Values
Architecture	RNN
Number of Layers	1
Encoder Hidden Size	300
Decoder Hidden Size	300
Batch Size	3000
Batch Type	Tokens
Vocabulary Size	32,000
Learning Rate	0.001
Dropout	0.0
Optimizer	Adam
Tokenization	BPE

Table 4: Hyperparameters of the RNN Model

In the Table 5 there are the BLUE and METEOR scores for the two models.

Model Name	BLEU Score	METEOR Score
Transformer Model	8.6	0.26
RNN Model	8	0.35

Table 5: BLEU (0-100) and METEOR (0-1) scores (higher scores indicate better performance)

While the RNN model has been trained with 44212 sentence pairs seems to have a lower BLEU score than the Transformer model which was trained only with 7253 sentences. However, the METEOR score follows the opposite trend. This happened because the sentences that the RNN model was trained on were very long, with some sentences being up to 2000 tokens, while for this project the sentences that were selected were up to 20 tokens per line. So, shorter sentences have better results.

4.1 Error Analysis

In order do the error analysis I will use the generated sentences from the RNN and Transformer architecture for comparison regarding the morphology, syntax, semantics, and pragmatics errors.

ID	Source	Reference	RNN Model	Transformer Model
1	ὑπὸ κύνα καὶ πρὸ κυνὸς ἐργώδεες αἱ φαρμακεῖαι.	About the time of the dog-days, and before it, the administration of purgatives is unsuitable.	When a dog and a dog should be purged by a dog, the medicines are to be purged of a dry and complicated diet.	About the time of the dog-days, and before it, the administration of purgatives is unsuitable.
2	ὁ δὲ Κλέων τέως μὲν ἡσύχαζεν, ἔπειτα ἠναγκάσθη ποιῆσαι ὅπερ ὁ Βρασίδας προσεδέχετο.	After remaining quiet for some time, Cleon was at length obliged to do as Brasidas expected.	When they had gone through the island to Paches, commanding officer for a long time, he was constrained to appeal.	Cleon, however, remained quiet for a while, and then was forced to do what Brasidas had received.
3	μάτην δὲ σέβονταί με, διδάσκοντες διδασκαλίας ἐντάλματα ἀνθρώπων.	And in vain do they worship me, Teaching as doctrine rules made by men.	But in vain do they worship me, teaching me by teaching the teaching of men.	But in vain do they worship me, Teaching as doctrines the commandments of men.

Table 6: Example translation results from RNN and Transformer Models

In the Table 6 above there are three sentence examples from the RNN and Transformer models as well as the source and the reference sentences. The sentences have been chosen randomly. Below there is a brief analysis on each sentence.

Morphosyntactic errors:
The RNN model has significant morphosyntactic errors in all three phrases. In the first and third sentences, the phrases display repetition, complicating the structure, and having no meaning. The second sentence has an unclear structure and lacks clarity and consistency. In comparison, the Transformer model has a clearer structure in all sentences, keeping the original meaning.

Semantic errors:
The semantic errors in the RNN model appear throughout the three sentences. In the first sentence, the translation incorrectly implies that the medications should be removed from a diet, affecting the original meaning. The second sentence introduces some other components that do not exist in the reference one. However, the Transformer model has a better structure and keeps the original meaning of the sentence.

Pragmatic errors:
From a pragmatic perspective, the RNN model fails to preserve the original context over the three statements. The health practices associated with dog days are lost in the first phrase, leaving the message ambiguous. The second generated phrase is irrelevant to the reference one. However, the Transformer model retains the pragmatic context, transmitting the intended meanings of course with some errors too.

In general, from the error analysis, the Transformer model seemed to have better quality translation than the RNN even though it was trained with 7253 sentence pairs. In the future, I would like to be able to train the Transformer model with all of the data achieving even better performance.

References

[1] Lanxin Zhao, Wanrong Gao, and Jianbin Fang. High-performance english–chinese machine translation based on gpu-enabled deep neural networks with domain corpus. Applied Sciences, 11(22), 2021.

[2] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In Mohit Bansal and Heng Ji, editors, Proceedings of ACL 2017, System Demonstrations, pages 67–72, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[3] Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

[4] Jörg Tiedemann. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, and Matteo Negri, editors, Proceedings of the Fifth Conference on Machine Translation, pages 1174–1182, Online, November 2020. Association for Computational Linguistics.

[5] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith, editors, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in Neural Information Processing Systems, 30, 2017.

[7] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[8] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss, editors, Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.