Machina Cognoscens: Neural Machine Translation for Latin, a Case-Marked Free-Order Language

Gil Rosenthal
grosenthal@uchicago.edu
University of Chicago Master's Thesis

ABSTRACT

Neural methods have brought a revolution in automated Machine Translation processes, with most highly-spoken languages having robust training datasets and near-human performance. However, these methods have lacked the same effect in Case-Marked Free-Order languages. A free-order language is one that has no specific word order, i.e. the subject, verb, and object can be anywhere in the sentence without violating the rules of the grammar. Case-marked means that additional information about the word, such as the number and function, are encoded in morphological features of the word, such as case or conjugation. As a target language, we use Latin, which is a FOCM language with extremely poor machine translation tools existing. We have created a first-of-its-kind Parallel Translation Dataset consisting of roughly 100k pairs, and evaluated its performance in Neural Machine Translation, with novel methods of preprocessing to encode morphology, and new investigations into transfer learning. We achieve a best performance BLEU of 22.4 on the test dataset, which beats the current State of The Art Google Translate model by over 4.2 BLEU, and publish our pre-processing pipelines for further research usage.

1 INTRODUCTION

Machine Translation (MT) is a technology that allows for the automatic translation of text from one language to another. Throughout the past few decades, MT has evolved from rules-based, leveraging 1 to 1 dictionaries, to Statistics-based, learning phrase probabilities from corpora, and finally arriving to Neural Machine Translation, leveraging the flexibility of Deep Learning. A key factor in the evolution of these technologies has been the acquisition of large parallel text datasets, allowing for machines to learn correlations and rules on their own, without human translators writing heuristic algorithms by hand.

MT research is conducted primarily on widely-spoken languages, because they have the greatest potential impact for an automatic translator. The most common language pairs for MT research include English to/from other major languages such as Chinese, Spanish, and French. These languages have large global populations, and are often used to as the basis of business, trade, and international communication. In addition, the availability of large amounts of parallel text data for these language pairs makes them well-suited for training and evaluating MT systems.

This focus on common language pairs in MT research can be disadvantageous for less used languages, like Latin. These languages have smaller, if any, global populations and may not have as much, if any, parallel text data available for training and evaluating MT systems. As a result, there may be less research and development in MT for these languages, leading to lower quality translation systems.

Additionally, the lack of focus on smaller languages in NLP research may lead to a lack of understanding of the unique challenges and characteristics of these languages. For example, Latin is a case-marked free-order language, which poses obstacles to MT systems that may not be addressed in research focused on common language pairs. This paradigm is explained below.

While the classification of "Free-Order Case-Marking" may have only been coined recently [2], the linguistic phenomenon has existed for centuries, and has plagued MT systems for decades [1].

A Free-Order language is one that does not have a set word order, meaning that unlike many common languages which follow the Subject Verb Object/OSV/VOS scheme, these three constituents may appear anywhere in the sentence, with any number of other words intervening. It is a common joke among Classicists that to find the verb in a Cicero sentence, you only need to turn the page.

A Case-Marked language is one that expresses semantic and grammatical functions of words through rich morphologies. In Latin, Nouns often have 2 "roots", one for the Nominative and one for all others (usually based off of the Genitive), which then expands to 10 possible inflections. Verbs have 4 principle parts, which can expand to up to 249 different inflections [11]. However, all of these forms stem from the same "root" of meaning, and only differ in the context of how the word is expressed (tense, who is doing the action, voice, etc).

Combining these two properties of language results in a language that frustrates many MT techniques. Due to being Free-Order, the concept of using word order to identify the main constituents is inapplicable - it becomes indeterminable whether the three words in the middle of the sentence are SVO, OSV, or three random words that only add slight semantic meaning. Statistical methods lose the ability to exploit

their "look ahead/behind" method, because there is no guarantee that the adjacent words are related to the current word. Neural methods rely far more heavily on learning an adaptable attention function, as the model can no longer rely on only local words having an effect.

In this paper, we study the creation of a Machine Translation pipeline for Latin, a notable Case-Marked, Free-Order language. In addition to these challenges, Latin-English is classified as a Very Low Resource translation pair. While many modern NMT systems are trained on hundreds of millions to billions of translation pairs, at the start of this project there were none for Latin-English. First, we create a first-of-its-kind dataset consisting of roughly 100 thousand parallel sentences. Then, we evaluate existing and novel methods of NMT on these translation pairs. We achieve a best result of a 22.4 BLEU on our test dataset, generating coherent translations from the vast majority of our data, while still struggling with larger sentences and those with more confusing word orders.

2 RELATED WORK

2.1 Latin Translation

Historically, Latin has been the target for infrequent study among Machine Translation researchers, starting with heuristic morphology parsers, then rule-based jigsaw translators, and finally modern NLP pipelines with NMT.

2.1.1 Whitaker's Words. The first attempt at utilizing computers to aid in translation was Whitaker's Words [20], a Latin-English dictionary and translation tool developed by William Whitaker. The system allows users to quickly look up the meanings of Latin words and phrases. One of the key features of Whitaker's Words is its comprehensive coverage of Latin vocabulary, including entries for over 39,000 Latin words, covering both common and obscure terms. Additionally, Whitaker's Words can automatically parse an inflected Latin word into its root form and corresponding morphological information. Tremendous effort went into the inflection parser and internal dictionary, resulting in the most robust and faithful-to-grammatical-rules parser in existence. We leverage this program in our processing pipeline to enrich our inputs with morphological information.

2.1.2 BlitzLatin. BlitzLatin, developed by the same William Whitaker along with Cambridge University classicist John White, was the first attempt at creating a true Latin to English Machine Translator, beyond simply referencing dictionary entries. Employing algorithms modeled after chess AI, and leveraging the Whitaker's Words tool, BlitzLatin includes hundreds of individual heuristics to pick out correct meanings for words [20] [21]. However, limited both by the manual work required to determine the heuristics along with processing power of early 2000s, it did not produce highly successful translations. Work has continued since its original development, and translation quality has improved (unfortunately they do not provide us with a BLEU), but it still remains primarily based upon manual inputs of rules and mappings. While we are unable to use this program directly, it heavily influenced prior research into Latin-English machine translation and provides inspiration for our work.

2.1.3 CLTK. While not intended specifically for Machine Translation, the CLTK (Classical Languages Toolkit) [12] is an open-source Python Library developed and maintained by the Open Philology Project, offering tools and resources for natural language processing (NLP) tasks on classical languages. The toolkit includes a range of tools for text analysis, including tools for stemming, lemmatization, and tokenization. It also provides a robust API for integrating its own processes with new sources, which we utilize to integrate Whitaker's Words into our NLP pipeline.

2.1.4 NMT Approaches. Recently, a larger amount of interest in the field of Digital Classics has caused research into NMT on Latin to pick up, albeit not into English. In *Machine Translation of 16th Century Letters from Latin to German* by researchers at the University of Zurich, they create a first-of-its kind parallel dataset between Latin and German, and then perform experiments on creating novel MT training methods [6]. They generate their parallel data using bitext mining from online sources, leveraging Facebook's LASER framework of multilingual sentence embeddings. However, a not-insignificant amount of their mined data comes from recent, and some might say, unfaithful sources - such as a Latin Wikipedia Article about Grand Theft Auto III. While these articles are likely still written in valid Latin, they introduce words not found in old Latin vocabulary, and can possibly befuddle the training process. Beyond just training their models, they also innovate by re-training on back-translations, adding additional normalizations to the text (all *j*s turn into *i*s, removal of accents), and pre-training with Italian to German data first. They employ the Sockeye MT toolkit to perform these experiments, and achieve a best-case BLEU of 19.5. Building on their work, we use the Sockeye MT framework in our experiments to build our models, but in contrast we avoid Wikipedia bitext mining and only utilize professionally-translated data in order to prevent over-complication and contamination of our dataset.

Additionally, in *Latin-Spanish Neural Machine Translation: from the Bible to Saint Augustine*, researchers experiment in creating NMT systems from Latin to Spanish [15]. They use the Bible Vulgate, along with the letters of St Augustine as a dataset. Similarly, they base their experiments on the Transformer architecture, but instead utilize the OpenNMT toolkit. They use a shared vocabulary between the source

and target, and employ Sentencepiece for tokenization. They achieve a best-case BLEU of 10.1, due to the relative lack of training data. They also remark on the potential for Unsupervised Machine Translation processes to aid in the training, in which the models are first trained on monolingual corpora. We build on their work by incorporating the Vulgate bible into our dataset, but avoid the utilization of a shared vocabulary and OpenNMT.

In *A Latin-Catalan Parallel Corpus for Statistical MT*, researchers created a corpus of parallel Latin and Catalan texts, and then trained Statistical MT models on them [8]. They achieve a best case BLEU of 11.6, again due to the relative scarcity of data. However, they introduce the idea of leveraging previous tools (again, Whitaker's Words) to enrich their data with morphological features. Their best case BLEU is achieved on a dataset where each word has been split into a lemma and suffix, providing a boost of 0.7 over the baseline.

2.2 Factored/Morphological Translation Models

Factored Translation models are a type of machine translation (MT) approach that decompose the translation process into multiple stages, each of which focuses on a specific aspect of the translation [14]. For example, a Factored Translation model might include separate modules for lexical translation, syntactic translation, and semantic translation. In our context, we are interested in Factored Translation Models by means of decomposing a single word into its corresponding features - part of speech, case marking, dependencies - which we are able to figure out heuristically.

In the paper "Sublemma-Based Neural Machine Translation," [16] the authors propose a new approach to machine translation that uses sublemmas, which are sub-units of lemmas, as the basic units of translation. They break down words to their root forms, for which they then learn embeddings, and then concatenate further features based on the word's morphology and dependencies, encoding domain knowledge into the embedding vector itself. These features are trained embeddings, although the authors do not explain how the embeddings are trained. However, encoding features such as Part of Speech and word type should aid the Attention mechanism in determining relationships between Subjects, Verbs, and Objects, regardless of order. We do not have a method of directly adding/editing embeddings used in pre-trained models, but we intend on building on this work by pre-processing terms into multiple tokens encoding both the root and morphological features of each word, and allowing the model to learn separate embeddings for the morphology-specific tokens.

Another paper, "Incorporating source syntax into transformer-based neural machine translation" [4], proposes incorporating constituency parsing into the translation system by means of a mixed encoder. The encoder takes in a linearized constituency parse, which is represented by the original source sentence with syntactical annotations interwoven - "you have not been elected" turns into "ROOT (S (NP you) (VP have not (VP been (VP elected))) .))" This allows them to improve performance without altering the internal architecture at all, while still being able to incorporate source syntax. Similar to the prior work, we build on this paper by pre-processing the morphology directly in the text, allowing the Transformer to learn relationships directly, without additional attention or embedding mechanisms.

2.3 Low Resource Machine Translation

Within Low Resource MT, there are a few techniques that are primarily used to aid training of models, namely back translation [9] and transfer learning.

Many researchers use back-translation to improve their NMT performance [18]. Back-translation operates under the assumption that an NMT model has the same capacity to learn information as a normal Language Model. One first trains a model translating from target language to source language, and then uses that model to generate its own translations on novel target-language text, resulting in a dataset consisting of gold-standard target translations and silver-standard source translations. The synthetic data is then added to the source-target training set in order to bootstrap the decoder's training. In "Iterative Back-Translation for Neural Machine Translation", researchers use this system to create increasingly accurate synthetic data by updating the training samples as the model improves, thereby improving the model's performance by training on better data. We will utilize back-translation in an attempt to improve decoder coherence and fluency, providing it more English to learn with.

Additionally, transfer learning on similar language pairs is a common method to pre-train NMT models. For example, as done in the Latin → German paper, the model is pretrained on a larger parallel dataset consisting of Italian to German sentences. As Italian is still a commonly spoken language utilized in the EuroParl EU corpus, there is a far larger corpus of parallel translation pairs between the two languages. Italian is a direct descendent from Latin, and despite changing much over time, shares many morphological and grammatical features. Training on Italian to German allows the model to learn an decoder representation of German, while also learning an encoder representation of a language extremely close to Latin, allowing for more efficient training with the low resource dataset. We intend to experiment with the reuse

of multiple pre-trained <Language> → English models, to determine the importance of the similarity of the source languages to the performance of a transfer-learned NMT model.

3 METHODOLOGY

3.1 Data

Being a Very Low Resource translation pair, prior to this project, there existed no public parallel dataset consisting of Latin-English translations. However, there exist plenty of full-length translations of historical Latin texts, albeit not granularly-aligned. We construct a first-of-its kind dataset of 100k sentence-aligned translation pairs to train our model on, and for others to use in their research.[1] This dataset can be found at Huggingface. To construct this dataset, we source our translation pairs from three primary sources - Perseus, Loeb, and Vulgate. We partition our dataset by first shuffling it, and then randomly taking roughly 99k as training, roughly 1000 as valididation, and another 1000 as test.

3.1.1 On Alignment. For all of the data sources (except the Vulgate), manual alignment was necessary to fragment the scraped volumes into sentence-by-sentence pairs. This was a manual task and required great amounts of labor by an expert in Latin translation. First, parallel sources were scraped at the page level, after removing all footnotes and line numbers. Then, as some sentences bled between pages, pages were concatenated, and then the Latin was re-split into individual sentences by common punctuation marks. Finally, English sentences were manually aligned with the Latin sentences. Of note, this manual labor was not being done in the translation - the translations have already been created, peer-reviewed, and are generally regarded as Gold Standards in the classical community. The only additional error that may be introduced in alignment is that two sentences may not be perfectly aligned, and either the Latin or English may be missing/have added a few extra words.

In Quality Assurance trials, 100 translation pairs were randomly sampled, with the context of the pairs directly before and after them, and 99 of them were found to be perfectly aligned (in reference to the given translation), with only 1 having an extra English word that corresponded to the pair directly following.

3.1.2 Perseus. The Perseus Digital Library [3] is a digital repository of Ancient Greek and Latin texts. For many of the ancient works, they also catalog English translations. However, the translations are not aligned beyond the level of paragraph or section. After scraping this source, additional alignment was performed to break the parallel translations into sentences. Due to the messiness of the source data, we source only our translations of Julius Caesar's *De Bello Gallico* from Perseus, representing roughly 3% of our dataset.

3.1.3 Loeb Digital Classical Library. The Loeb Library has been the Gold Standard of classics translations for a century. The library includes over 520 volumes of Latin, Greek, and English texts, including works of literature, history, science, and more. Recently, all of their translations have been converted to a digital format and have been uploaded to their website. All Loeb volumes feature parallel translations, with the left page containing the source material and the right side containing the translation, lined up roughly by paragraph. Gaining access to the translations through scholarly credentials, we were able to scrape parallel translations from the Loeb volumes. We were able to automatically segment the Latin into sentences, but for consistency and accuracy sake the alignment with English is done manually as described above. We source the vast majority of our non-Vulgate translation pairs, roughly 65k, from Loeb.

3.1.4 Vulgate. The Vulgate is a Latin translation of the Bible, commissioned by the Pope in the 4th century AD. It is one of the most important and influential translations of the Bible, and it became the standard version used in the Western Church for many centuries. Church scholars have translated specifically the Vulgate edition of the Bible into English, and have published a line-by-line aligned translation onto the internet. We are able to scrape this and use it as a data source, resulting in roughly 32k translation pairs.

3.2 Distribution

The breakdown between Author and representation in the Training set is seen in Figure 1. Note that while the Vulgate was officially written by the author Jerome, the version on the site may have additional edits, so we attribute them solely under the category "Vulgate".

3.3 Preprocessing

To address the challenges of translating a Free Order Case Marked language, in addition to using the source text directly we introduce two novel preprocessing methods.

3.3.1 Stem/Case Split. Given that there are 350 unique morphological forms in Latin, we attempt to reduce the complexity by splitting each word into two parts: its root, and its case-marked ending. By performing this split, instead of having to learn every inflection of every word separately, the model should be able to learn the definition of a word based on its root, and then the context semantics (tense, person, etc) based on the ending. For example, the Latin word *Putavit* gets split into two tokens: [$putav, CASE\_it$] - intending for

[1] The English targets, as described in 4.7, are not the originals, due to Copyright restrictions

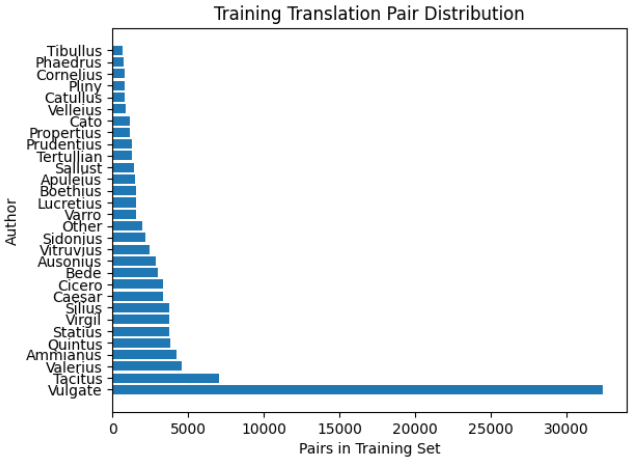

Figure 1: Data Distribution by Author

the model to learn *putav* = *to think* and *CASE_it* = 3rd person singular, past tense. We prepend each case form with the string CASE_ so that we can add special tokens to the tokenizer/model separating it from the string of characters "it". Indeclinable tokens are not split in any way, leaving just the base word. This word segmentation is powered by interfacing Whitaker's Words, which was described in the Related Work section.

3.3.2 Stem and Morphology. The Stem/Case split is the most intuitive way to encode the morphological features, but it fails to address case overloading. Case overloading is the phenomenon where a given word ending can represent more than one morphology, depending on the word it is attached to. For example, the ending *-is* can represent the 3rd person plural ablative or dative cases for a first declension noun, the 1st person genitive for a third declension noun, or a second person active imperative case for a third conjugation verb.

However, we know fairly well what class/part of a speech the root is going to be, so we can encode exactly what the ending means instead of the ending itself. For a verb, this corresponds to the features *[Is_Verb, Conjugation, Person, Number, Tense, Voice, Mood]*, for a Noun, *[Is_Noun, Declension, Number, Case, Gender]*, and similar information for other parts of speech. Encoding this information directly into the token embedding itself is an interesting and promising idea, however this was impossible using the NMT framework we were using without directly modifying the underlying code.

In an attempt to mimic this functionality, we represent this as a string and add it in place of the "case" information described above. However, contrary to the Stem/Case split, indeclinable words are still augmented with their context. As an example, the preposition *ob*, taking the Accusative case, gets annotated into the two tokens *[ob, PREPACC]*. *Putavit*, the example given above, is represented as *[putav, V11PERFACTIVEIND3S]*. In addition to preventing the endings from being overloaded, this has the added bonus of Part of Speech Annotation. Given that there is no set word order, having knowledge of what words in a sentence are verbs, nouns, etc., intuitively should aid in translation performance.

Adding morphology has a few caveats. First, these are seemingly nonsensical strings of characters. In a naive methodology, the sub-word tokenizer will break these down, resulting in a possible loss of actual utility, and perhaps confusing the model even more as the signal/noise ratio is reduced. We remedy this by adding all possible morphological forms as new tokens to the model's embedding before training, resulting in the avoidance of subword tokenization, along with the model learning embeddings for each morphology.

Additionally, each morphology string is possibly too rich with information - it encodes what "class" of noun/verb a word may be, which relates to its morphological inflection, but does not provide any additional information when we have parsed the morphology already. For example, "femina" is a feminine singular nominative noun, but it's also 1st declension 1st type, so it gets encoded as "N11SNF", while "dies", which is a feminine singular nominative noun, but 3rd declension 2nd type, is "N32SNF". So, even though in a grammatical sense they serve the same role, their morphology tokens are distinct and are learned separately. We experiment with removing these distinguishing characteristics and simplifying the morphology strings.

3.3.3 General Preprocessing. All 3 experimental datasets (untouched, stem/case, stem/morphology) underwent the same pre-processing after their generation. First, in the Latin source, all punctuation was removed under the assumption that each pair was a sentence. This may be an over-zealous preprocessing step, but any punctuation found in the text in fact was added by editors far after the source was written [10]. This also solved technical issues with the segmentation of words for the two special preprocessing methods, related to the Whitaker's Words tokenizer. Afterwards, all datasets were tokenized with a SentencePiece scheme with a vocabulary size of 65k words, or 67k with special morphological tokens, or 65,351 with simplified morphological tokens. Pairs were trimmed to a maximum length of 128 tokens.

4 RESULTS

4.1 Google Translate Baseline

As described in the Literature Review, there are no published Latin to English models whose results we can compare against. BlitzLatin is the only available specialized translator that we could theoretically compare against, but we lack the licensing to run it. As such, we will be using Google

Translate as our baseline to compare model performance against. Before 2021, Google Translate utilized Phrase-Based Machine Translation methods for Latin, leading to poor quality translations, causing educators worldwide to discourage students from using it.

A Reddit post [7] provides a good explanation for the poor performance of the old model. However, in fall 2021, Google updated the Latin translation models to their Neural Machine Translation framework, substantially improving performance. It still has many issues, as is discussed in the subreddit and Stack Exchange forum, but is generally regarded as able to output coherent text with some understanding of nuance.

We benchmarked both directions of English-Latin and Latin-English Google Translate by utilizing the Google Cloud Batch Translation service, which is presumably the same service underlying the commercial Google Translate application. We used each direction of the test set as the source text, with the other as the reference for BLEU calculation. For the Latin to English direction, Google Translate achieves a BLEU of 18.12. For the English to Latin direction, Google Translate achieves a BLEU of 15.62. We treat these both as the previous "State of The Art", as they are the only publicly available models for either direction.

4.2 Architecture

All models were based on the Transformer architecture. Initial experiments were performed using standard PyTorch Seq2Seq training, and subsequently the Sockeye NMT toolkit [5]. For each of the 3 experimental datasets, we train 3 models of varying size. The "small" model consisted of a 2 layer model, with 4 attention heads, and a model embedding size of 384. All models were trained for 40,000 Gradient Updates, which is roughly 10 epochs, with an initial Learning Rate of 0.005 and an Inverse Square Root Decay Learning Rate schedule. Validation checkpoints were taken every 4000 gradient updates, computing the validation perplexity and BLEU, along with reporting the current training perplexity value.

Subsequently, experiments were based on the MarianMT [13] framework, using pretrained models published by Helsinki-NLP on Huggingface. After hyperparameter tuning, models were finetuned for 5 epochs with a learning rate of 0.001. Validation was performed each epoch, with an optimized loss being Cross Entropy, and validation loss being the BLEU.

All experiments were performed on the Argonne Polaris supercomputer, using NVIDIA A100 GPUs.

4.3 Training from Scratch

4.3.1 Sockeye Framework. Initial experiments focused on training the model from scratch, using the limited dataset to learn both a Latin Encoder and an English Decoder. Utilizing the Sockeye NMT framework, which performed no subword tokenization, and therefore resulted in a significant presence of <unk> tokens, we achieved a best case BLEU of 10.8 utilizing morphological strings, 9.9 for stem + case split, and 8.9 for the raw text.

Qualitative inspection of the translations revealed that for small sentences, that follow a word order resembling SVO, the model's output was relatively satisfactory. Compare the model's output "Blessed are all that love thee: and they that rejoice over thy peace." with the ground truth "Blessed are all they that love thee, and that rejoice in thy peace,". However, with sentences that do not follow a SVO/OVS/SOV word order (which are grammatically correct in Latin), performance diminishes and use of the <unk> token greatly increases, along with confusions of subjects/objects: compare "When it is a vine which is to be cut off as a <unk>" with the ground truth "When it is two years old, cut off the branch below the basket".

Additionally, the consequences of training an English model solely on archaic translations become evident when observing vocabulary choices - "thy" and "thee" are extremely common in the outputs, and some words that are no longer utilized in colloquial English due to offensive connotations are also found. Having the Bible compose roughly half of the training dataset becomes clear, as well - there is a lot of "the Lord" and "Jerusalem" output, especially in contexts with new or unseen words.

4.3.2 PyTorch Transformer. While Sockeye is a great framework for training simple NMT models, it is unfortunately not easy to extend or modify in any substantial manner. In an attempt to explore some novel/unsupervised techniques for our research, we then attempted to replicate our results in a vanilla PyTorch setting, which we could then modify easily with novel experimental techniques.

Unfortunately, we were unable to replicate our results in a PyTorch environment. Despite utilizing online tutorials and published code, we were unable to get our model to be coherent with sentences longer than 2 words at all. Models seemed to develop a dictionary understanding of words mapping to each other - they could translate "Troiae" as "of Troy" and "qui" as "who", which are both correct, but were unable to combine words and ideas with any coherence. Despite our efforts to investigate the reason, we were unable to determine why the vanilla Transformer, with seemingly identical architectures, failed to produce coherent results. Consequently, we decided to move on from this approach. The highest BLEU score it attained was 2.1, which is considerably unsatisfactory.

4.4 Transfer Learning

With training from scratch proving difficult, we decided to pivot to utilizing Transfer Learning to leverage a pretrained English Decoder. Following the work of [6], for our first experiment we chose an Italian-English model as our base model. As we are translating into English, a pre-trained English Decoder was a clear choice. Our rationale behind using an Italian Encoder is that Italian is a direct descendent of Latin, and while a lot of the morphology has been simplified, the semantic meaning of many of the roots has remained.

4.4.1 Initial Transfer Learning Experiments. We utilized the Helsinki-NLP IT-EN [19] model as our base for finetuning. Interestingly enough, our Latin text, when tokenized, only uses 7258 unique tokens out of the model's vocabulary of 80,378, or only about 9%. Many of the most common tokens from our tokenized set are individual characters, or common case endings. Before finetuning, this model had a BLEU of 0.37, and essentially just repeated the Latin back as its "translation." This model, finetuned on our data, produces a test BLEU of 21.29, already beating Google Translate by 3 BLEU points. Further, the translations are coherent English, which we hypothesize is due to the pretrained English decoder. Qualitatively, it appears that the model is far better at handling Proper Nouns. However, it still struggles with ambiguity of subjects and objects, and frequently confuses between them.

We then experimented with Helsinki-NLP's Helsinki-NLP Romance-IT [19], intended to translate from any Romance language (which includes Latin) to English. We hypothesized that a synthesis of all Romance languages, including Latin (although the training data was highly suspicious) would outperform just Italian in a fine-tuning setting. In this case, our text utilized 7489 unique tokens out of 65,001, or about 11.5%. Before fine-tuning, this model achieved a BLEU of 6.77, which is higher than the Italian model, but still worse than our Sockeye attempts. With the exact same finetuning setup, this model achieved a test BLEU of 21.65, or 0.36 BLEU points higher than the Italian-only. After the initial training, we discovered that this model, in order to disambiguate between source languages, required a prefix to each sentence noting the language - for Latin, it was "»la«". With no finetuning and this prefix, we get a BLEU of 6.56, which is actually lower than if we did not specify that we were using Latin at all. After adding this prefix, though, we achieved a test BLEU of 22.32, beating the Italian-only model by over 1 full BLEU point.

4.4.2 Non-Romance Finetuning Sources. Based on these findings, we had a reasonable degree of certainty that employing a pre-trained encoder, which has been exposed to the source language, is advantageous during the fine-tuning process. However, we aimed to further examine the extent to which the performance could be attributed to the encoder itself or to a coherent English decoder. So, we decided to take three other models, Arabic-English, German-English, and Chinese-English, and compare fine-tuned performance.

Utilizing the Arabic to English model, our source text uses 1253 unique tokens out of the 62,833, or only 2% of the tokens. This makes sense, as the majority of Arabic text does not include Latin characters. The vast majority of the tokens are either individual characters or pairs of characters, acting as subword units. With no finetuning, this model achieves a BLEU of 0.32, again just copying back the original Latin. With an identical finetuning setup to prior experiments, we achieve a test set BLEU of 19.53, which is lower than the Italian/Romance models but not by an exceedingly large margin.

Utilizing the Chinese to English model, our source text uses 983 unique tokens out of the 65,000, or only 1.5%. Similarly, the majority of Chinese text does not include Latin characters. With no finetuning, this model achieves a BLEU of 0.17, this time outputting a strange mix of English and Latin. The majority of English output was repetitive "I'm sorry, but I don't know." After finetuning, we achieve a test BLEU of 18.54, which is again lower than the Italian/Romance models, but still even beats the Google Translate model.

Utilizing the German to English model, our source text uses 4355 unique tokens out of the 58,100, or 7.5% of the tokens. This is significantly higher utilization than either of the Arabic/Chinese models, but still less than both the Italian/Romance models, given that German is a non-Romantic language and likely has less language similarity. With no finetuning, we achieve a BLEU of 0.41, again essentially just copying over the input text as the "translation." Fine-tuned, we achieve a BLEU of 20.33, which is within 1 point of the Italian model despite having a totally different source grammar and vocabulary.

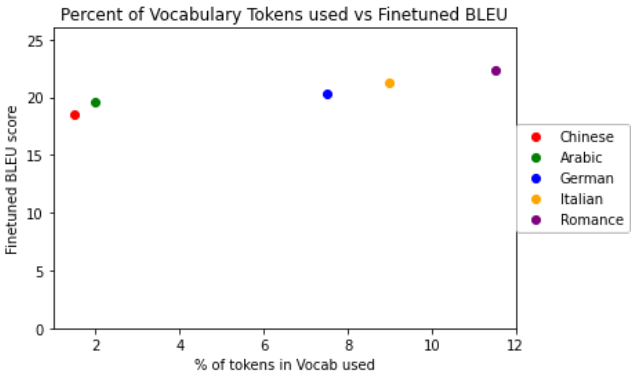

Figure 2: BLEU vs Source Vocabulary Usage

In Figure 2, we see a weak positive relationship between percent of tokens used within a vocabulary, and the fine-tuned BLEU of the model. Theoretically, a model with perfect coverage (with a vocabulary trained solely on our dataset) would perform the best, but in doing so we would lose the advantage of using a pre-trained model. This relationship, and the relatively decent performance of even the fine-tuned Chinese model, suggests that a large portion of the performance of the pretrained models comes from using a pretrained English Decoder that is able to output coherent and sensible translations, once finetuned, regardless of the original input encoding.

4.5 Preprocessing Experiments

The results of transfer learning were quite strong, significantly improving upon the Google Translate baseline. As they were far more powerful than training other models from scratch, we decided to base the remainder of our experiments on these transfer models in an attempt to further increase performance. We chose the Romance language based model, and experimented using both the Stem/Case split and Morphological preprocessing.

4.5.1 Stem/Case Split. We first fine-tune the Romance language model on our basic Stem/Case split dataset, with an identical setup as our previous experiments. We achieve a test set BLEU of 21.00 with this model, which is better than Google Translate but worse than training either the Romance or Italian model on just the raw text. Looking at the translations, we see a similar if not more exacerbated problem of confusion between subject/object and constituency parsing. This can likely be attributed to the relative lack of exposure to the CASE_ tokens, resulting in inaccurate embeddings and possibly confusing the model more than enlightening. Overall, though, translation quality is relatively high.

4.5.2 Morphological. We first fine-tune the Romance language model directly on the Morphological dataset, with no special tokens. Unfortunately, we achieve quite poor results - a test BLEU of 15.55. This, with high certainty, is due to the SentencePiece tokenization of the morphological strings - they are long and are liable to be broken up into however many subtokens.

As such, we then attempt to add all of the possible morphological strings as entries into the model's vocabulary. As described in section 3.3.2, this adds roughly 2000 new entries to the vocabulary, as each type of noun/verb has their own set of morphology strings. Adding these to the vocabulary, we now achieve a test set BLEU of 21.37, roughly equivalent to the results of the finetuned Italian-English model on the normal dataset, as the morphological information is being used as actual tokens rather than confusing strings of letters.

Finally, we attempt to simplify the morphological strings. We reduce a string such as *V11PERFACTIVEIND3S* to the string *Verb_Perfect_Active_Indicative_Third_Singular*, which we then add to the vocabulary as a single token - note the loss of the two numbers up front, specifying the type of verb it is. Performing this simplification results in 350 morphological tokens being added to the vocabulary. Running the same fine tuning, we achieve a test set BLEU of 21.31, which is slightly below that of the not-simplified tokens. Perhaps the individual information about what type of word can be useful, but these differences also seem to be roughly within the margin of error.

Contrary to our initial results with Sockeye, pre-processing did not aid in performance in the transfer learning setup, even with the case and morphological markers specifically marked as special tokens. There are a few possible reasons for this. First, the dataset simply could be too small to incorporate the addition of new tokens, and to learn a meaningful embedding for them. Especially with nearly 2000 different morphological strings, a training size of 100,000 might not be enough to learn their semantics in a useful way. Second, textually pre-processing may not be the correct method when incorporating these features in a transfer learning setting. Possibly adding in a secondary embedding representing morphology and subsequently an additional layer to the model to incorporate these features could be a better method of doing it, allowing the model to attend to the features in a more direct way. These embeddings could possibly be initialized by a skip-gram model trained solely on morphologies, without stems. In either scenario, this avenue of research merits further investigation.

4.6 Backtranslation

In an attempt to augment our training dataset for transfer learning, we perform backtranslation on monolingual data. The intention behind backtranslation is that by augmenting our data with gold-standard English, translated into silver-standard Latin, we will at the very least improve our English decoder's fluency, and hopefully also improve the Latin Encoder's robustness to partially corrupted language.

First, we train English to Latin models for all three datasets. We build off of the corresponding Helsinki-NLP English to Romance language models for each. As Latin is an extremely morphologically rich language, as we have discussed before, we do not expect the translation scores to be as high in this direction. For English to Latin normal direction, we achieve a BLEU of 13.15, which is unfortunately lower than Google Translate. However, as English to Latin synthesis is not the intent of our project or the vast majority of scholarly work, this is not a critical research failure. With more work on

ensuring correct morphology is generated, we could possibly learn a better model. However, this is solely for data augmentation in our scenario, so it is less important.

For the stem/case split, we achieved a BLEU of 20.06. However, it is not feasible to directly compare this metric with the outcomes from Google Translate or the standard results. This is because the stem/case dataset essentially represents a distinct language from conventional Latin, given that each original word is divided into its stem and case components.

For the morphological processing, with no special tokens we achieved a BLEU of 16.99. Adding special tokens, we achieve a BLEU of 22.41. As with stem/case, this BLEU cannot be interpreted in comparison with English to Latin given differing target vocabularies. Given that we saw no improvement using simplified Morphological tokens, we decided not to backtranslate using them as well.

With our English to Latin/Latin variant models trained, we then perform Backtranslation on roughly 141,000 monolingual English sentences. The composition of what monolingual data we would use to backtranslate was a difficult problem, because we did not want to bring in modern day words or technology that would simply reduce the applicability to translating ancient works. We first chose an Anthology of English works from the years spanning roughly 1500-1700 AD, and then added a selection of historical texts about Greek, Roman, and Near Eastern civilizations. All texts were then filtered to exclude any line under 50 characters or that included a non-ASCII character, and then re-organized from line-by-line (of each book) to sentence-by-sentence.

With our backtranslated dataset created, we then re-train the Latin to English models from scratch with the combined dataset. We do not perform any re-sampling or balancing, as the datasets differ in size only slightly.

For the normal Latin to English model, after including the backtranslated data, we see a test BLEU of 22.43, only about a 0.1 BLEU increase over the original model. However, on a qualitative observation, the backtranslated model seems to do better at distinguishing between objects and indirect objects and subjects, and additionally has a slightly more modern vocabulary, likely due to the inclusion of modern historical texts.

After including the backtranslated data, the Morphological split, with no special tokens, increases in performance from its original fine tuning model, to 16.74. However, with special tokens, we have a BLEU of 21.28, which is 0.09 lower than its not-backtranslated baseline.

For the split model, we again see a slight decrease in test BLEU from 21.00 to 20.73.

The fact that all models saw a decrease or essentially no change after performing backtranslation is disappointing, but it possibly speaks to the effect of transfer learning from a pretrained model. As previously described, the main intention of backtranslation is to increase the fluency and coherence of the decoder by providing it with a gold standard of English, with then a silver standard (reverse translated) Latin source. It is possible that the English decoder is nearly at convergence when we import from the pretrained model. It is also possible that our reverse translation methods are not good enough to generate viable synthetic data, in fact hurting the model. Perhaps that the monolingual data being backtranslated is either out of domain or corrupted to the point where the English isn't coherent enough to aid in learning. All of these warrant further investigation.

4.7 Modernized Translations

Given that the majority of translations that we train our model off of were created in the early 1900s, there is an outsized representation of outdated vernacular - verb forms like "dost" and "goeth", and pronouns such as "thee" and "thou." Modern English no longer utilizes these forms, so we experiment with updating the English part of the translation pairs to Modern English, and tgeb transfer learning on the modernized pairs to see if performance and readability will improve. Additionally, as the Latin itself is in the Public Domain due to all authors dying millenia ago, our novel translation pairs will not be restricted by any Copyright, so we can release them publicly for free use.

To update the translations, we utilize the OpenAI GPT-3.5 Turbo model. We use the following prompt: "Translate an old dataset from the 1800s to modern English while preserving the original meaning and exact same sentence structure. Retain extended adjectives, dependent clauses, and punctuation. Output the translation preceded by the text "Modern Translation: ". If a given translation is not a complete sentence, repeat the input sentence. Original: {Original Text}". This prompt was settled on after multiple iterations of Prompt Engineering. The primary focus was to maintain sentence structure, which is derived directly from Latin, while treating the goal of updating the vocabulary as a secondary objective.

Over 99% of the output translations conformed to our standard, while there were a few that required manual correction, with the model apologizing that the input sentence was not complete and that it required further context: "Modern Translation: If you could please provide context or further information, it would be helpful for me to understand the meaning behind your statement. If not, I am afraid I cannot provide a proper response as it is not a complete sentence."

After modernizing the translations, we first re-ran our BLEU comparison on Google Translate, to see if it aligned better with Modern English rather than archaic English. It achieved a BLEU of 15.00, over 3 points worse than on the original baseline. This is discussed below.

We then re-ran all of our prior experiments on the modernized dataset. The Base model achieved a best BLEU of 19.85, again over 2.5 points lower than the prior experiment. Stem/Case-Split achieved a BLEU of only 19.04, once again significantly lower than the original dataset. Morphological split achieved a BLEU of 19.20, while Simplified Morphological Split achieved a BLEU of 19.16.

In all cases, we see a significant decrease in performance. There are a few possible reasons for this. Firstly, despite our attempts at formalizing behavior through the prompt, GPT-3.5 is a non-deterministic and unconstrained model. Given the nature of GPT models, unless we read through all modernization translations, we have no method of ensuring it has followed the instructions in every case. Further, even if it does follow all of our specifications, it is possible that its decisions to modernize certain words can, in fact, change the semantics of a sentence. Changing a translation of *audax* from "bold" to "aggressive" may not appear to be a large difference in human understanding, but it in fact will confuse the model's learning between different vocabulary and result in worse translation quality. It is also possible that our models were overfit on archaic English, so that just outputting a "thee" and a "thou" at random has a high likelihood of matching the vocabulary of a given translation and falsely increases the BLEU. However, this line of reasoning is likely to be more tenuous given that Google Translate, which likely has a very good Modern English decoder model, also suffered in the updated translations.

Further constraining generation by applying specific filters, such as Sentence Transformers Similarity [17] between the source and output, checking actual sentence structure, and creating a more restricted list of possible "updates" could result in better modernized translation performance. Sourcing from more modern translations to begin with would also aid in this task, but unfortunately at the moment none seem to be freely available.

5 CONCLUSION AND FURTHER WORK

We have shown that for a low-resource, case-marked, free-ordered language like Latin, the best performance will be achieved by performing transfer learning with a model that has already learned a robust and coherent decoder into the target language. Contrary to our preliminary results training both languages from scratch, utilizing morphological features such as a stem/case split or a specific morphology token does not improve performance significantly, if at all. Back-translation provided modest gains in translation performance, likely by increasing the fluency of the generated English translations by providing the decoder with more samples. Experiments in using modernized translations do not aid in translation performance, likely due to methodological flaws in the modernization process.

Most importantly, we established a new State of the Art in Latin to English machine translation. We outperform Google Translate by 4.2 BLEU, and provide the trained model for public usage and development. Further, we have created a first-of-its kind parallel dataset for further research usage.

In future work, performing novel techniques such as Unsupervised NMT on monolingual Latin data would likely improve performance. Other representations of source syntactical information, such as a dedicated embedding for morphology, or a replacement for Positional Encoding representing Parts of Speech/Constituency parsing could also aid in incorporating domain knowledge into the model . Additionally, maximizing the size of the training dataset will always aid in rendering the model more robust and fluent. Further, deeper research (beyond a simple backtranslation model) into the opposite direction, from English to Latin, could aid in our understanding of the ability of NMT models to learn to generate sentences in morphologically rich languages. A similar process as our pre-processing methods could be employed to combine the semantic sense of a word with its relevant context information to generate an inflected form. Additionally, inclusion of Latin into multilingual NMT systems could aid in the discovery of parallels/similarities with other languages, while also extending the impact of the translator.

ACKNOWLEDGMENTS

Thanks to Professor Ettinger for providing continual guidance and aid in my Thesis. Additional thanks to Professors Stevens and Foster for teaching a wonderful class Fall 2022, and for allowing us to use the Argonne resources to perform experiments, and to Professor Tharsen, for guiding my project in a Digital Humanities direction.

REFERENCES

[1] Alexandra Birch, Miles Osborne, and Philipp Koehn. 2008. Predicting success in machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. 745–754.

[2] Arianna Bisazza, Ahmet Üstün, and Stephan Sportel. 2021. On the Difficulty of Translating Free-Order Case-Marking Languages. *Transactions of the Association for Computational Linguistics* 9 (11 2021), 1233–1248. https://doi.org/10.1162/tacl_a_00424 arXiv:https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00424/1972440/tacl_a_00424.pdf

[3] Gregory Crane. 2022. The Perseus Digital Library and the future of libraries. *International Journal on Digital Libraries* (19 Aug 2022). https://doi.org/10.1007/s00799-022-00333-2

[4] Anna Currey and Kenneth Heafield. 2019. Incorporating Source Syntax into Transformer-Based Neural Machine Translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*. Association for Computational Linguistics, Florence, Italy, 24–33. https://doi.org/10.18653/v1/W19-5203

Dataset	Base Model	Training	Test BLEU
Base	Google Translate	N/A	18.12
Base	Sockeye	Sockeye	8.9
Stem/Case	Sockeye	Sockeye	9.9
Morphology	Sockeye	Sockeye	10.8
Base	PyTorch	PyTorch	2.1
Base	Italian-English	None	0.37
Base	Italian-English	Fine-Tuning	21.29
Base	Romance-English	None	6.77
Base	Romance-English	Fine-Tuning	21.65
Base	Romance-English	None + >>la<<	6.56
Base	Romance-English	Fine-Tuning + >>la<<	22.32
Base	Arabic-English	None	0.32
Base	Arabic-English	Fine-Tuning	19.53
Base	Chinese-English	None	0.17
Base	Chinese-English	Fine-Tuning	18.54
Base	German-English	None	0.41
Base	German-English	Fine-Tuning	20.33
Stem/Case	Romance-English	Fine-Tuning	21.00
Morphological	Romance-English	Fine-Tuning	15.55
Morphological	Romance-English	Fine-Tuning + Special Tokens	21.37
Morphological	Romance-English	Fine-Tuning + Simplified Tokens	21.31
Base	Romance-English	Backtranslation	**22.43 (Best)**
Stem/Case	Romance-English	Backtranslation	20.73
Morphological	Romance-English	Backtranslation	16.74
Morphological	Romance-English	Backtranslation + Special Tokens	21.28
Base (Modern)	Romance-English	Fine-Tuning	19.85
Stem/Case (Modern)	Romance-English	Fine-Tuning	19.04
Morphological (Modern)	Romance-English	Special Tokens	19.20
Morphological (Modern)	Romance-English	Simplified Tokens	19.16

Figure 3: Latin to English Experiment Results

[5] Tobias Domhan, Michael Denkowski, David Vilar, Xing Niu, Felix Hieber, and Kenneth Heafield. 2020. The Sockeye 2 Neural Machine Translation Toolkit at AMTA 2020. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*. Association for Machine Translation in the Americas, Virtual, 110–115. https://www.aclweb.org/anthology/2020.amta-research.10

[6] Lukas Fischer, Patricia Scheurer, Raphael Schwitter, and Martin Volk. 2022. Machine Translation of 16th Century Letters from Latin to German. In *Second Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2022)*. LREC, 43–50. https://doi.org/10.5167/uzh-218848

[7] Gilgamesh. 2017. Why is google translate so bad for Latin? A longish answer. https://www.reddit.com/r/latin/comments/6akqdi/why_is_google_translate_so_bad_for_latin_a/

[8] Jesús González-Rubio, Jorge Civera, Alfons Juan, and Francisco Casacuberta. 2010. Saturnalia: A Latin-Catalan Parallel Corpus for Statistical MT. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA), Valletta, Malta. http://www.lrec-conf.org/proceedings/lrec2010/pdf/541_Paper.pdf

[9] Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. 2022. Survey of Low-Resource Machine Translation. *Computational Linguistics* 48, 3 (Sept. 2022), 673–732. https://doi.org/10.1162/coli_a_00446

[10] Arthur W. Hodgman. 1924. Latin Equivalents of Punctuation Marks. *The Classical Journal* 19, 7 (1924), 403–417. http://www.jstor.org/stable/3288661

[11] Jonas Ilmavirta. 2016. How many distinct forms does a typical Latin verb have? https://latin.stackexchange.com/questions/921/how-many-distinct-forms-does-a-typical-latin-verb-have

[12] Kyle P. Johnson, Patrick J. Burns, John Stewart, Todd Cook, Clément Besnier, and William J. B. Mattingly. 2021. The Classical Language Toolkit: An NLP Framework for Pre-Modern Languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 20–29. https://doi.org/10.18653/v1/2021.acl-demo.3

[13] 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*. Association for Computational Linguistics, Melbourne, Australia, 116–121. http://www.aclweb.org/anthology/P18-4020

[14] Philipp Koehn and Hieu Hoang. 2007. Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics, Prague, Czech Republic, 868–876. https://aclanthology.org/D07-1091

[15] Eva Martínez Garcia and Álvaro García Tejedor. 2020. Latin-Spanish Neural Machine Translation: from the Bible to Saint Augustine. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*. European Language Resources Association (ELRA), Marseille, France, 94–99. https://aclanthology.org/2020.lt4hala-1.14

[16] Thien Nguyen, Huu Nguyen, and Phuoc Tran. 2021. Sublemma-Based Neural Machine Translation. *Complexity* 2021 (October 2021), 1–9. https://doi.org/10.1155/2021/5935958

[17] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084 [cs.CL]

[18] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 86–96. https://doi.org/10.18653/v1/P16-1009

[19] 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 (EAMT)*. Lisbon, Portugal.

[20] W. A. Whitaker and J.F. White. 2002. Blitz Latin. Experiments with automatic translation of Latin. *Journal of Classics Teaching Review* 32, 32 (2002), 2–8.

[21] John F. White. 2015. Blitz Latin Revisited. *Journal of Classics Teaching* 16, 32 (2015), 43–49. https://doi.org/10.1017/S2058631015000203

Received 5 March 2023