The Bible is All You Need: Translating the Bible using Transformers

Yorai Shaoul, Dana Rosenfarb, Zachary Metzman, Doron Hazan

Abstract

Hebrew has changed drastically from its use in the Old Testment. In fact, it has changed so much that even native Hebrew speakers have trouble understanding the ancient or Biblical language. Given the immense interest in ancient languages that is shared by linguists, religious figures, and students, we wish to facilitate the understanding of old languages by modern day readers via deep learning methods. To this end, we made use of state-of-the-art translation methods to obtain a deep neural model that translates Biblical Hebrew to Modern Hebrew. In this work we present our collected dataset of a Modern Hebrew Translation of the Bible and two Biblical to Modern Hebrew machine translation models. Additionally, we provide a quantitative and qualitative evaluation of our methods. We report BLEU scores of 82% and 98%, showing the efficacy of our methods in the task of Hebrew translation, and provide a discussion suggesting that adaptation of pretrained language models to new tasks might be superior to end-to-end learning.

The code for this project is available online at https://drive.google.com/drive/folders/1sK_ECSV_buoGGQ-xtkKV5m7iXCcvLeiT?usp=sharing.

1 Introduction

The Bible is widely considered to be the most read book existing.[1] It is a collection of religious texts or scriptures sacred to Jews, Samaritans, Christians and others. Believers might generally consider the Bible to be a product of divine inspiration, and the Bible is considered as the source containing the fundamentals of the Jewish, Christian and various other religions. The straight link to religion, popularity and reputation of the Bible led it to have a profound influence on innumerable humans throughout history.[2]

Figure 1: The Hebrew Alphabet and English Pronunciation. (Source: Jewish Museum London.)

Over the years, the Hebrew language has evolved from its ancient roots to a modern and (somewhat) widely used language. As a result, older Hebrew texts (i.e., the Old Testament) take on a different structure and style from their modern translations. Therefore, the Old Testament or the Bible written in Biblical Hebrew is very hard to read and perhaps even uninterpretable by many Hebrew speakers nowadays. Since the Bible and its content have immense influence on so many people, its translation has immense ramifications. Namely, the translation of the Bible has to be precise and interpretable.

However, we found no evidence for previous literature that aimed to establish mapping between Bilblical and Modern Hebrew. In this paper, we used existing Natural Language Processing approaches to try complete this task. We looked at various sources that offer Bible translations. Most

[1] According to the 2015 edition of Guinness World Records

[2] The Hebrew Bible shares most of its content with its ancient Greek translation, the Septuagint, which is the base for the Christian Old Testament. The Christian New Testament is a collection of writings by early Christians, believed to be mostly Jewish disciples of Christ, written in first-century Koine Greek. In this paper we use the Biblical Hebrew Old Testment

of the authors' native language is Hebrew and have experience reading it to make an independent judgement of the quality of the translation offered by the different sources. We concluded that the ultimate source for our project is the public Annotated Bible (מקרא מבואר). This source is publicly available on Wikitext and includes translations that we believe are widely acceptable within the Hebrew-speaking community.

In addition, our work was also motivated by providing potential contributions to these following needs:

- A need to provide a translation framework between ancient and modern languages. This was rarely done in the past.
- A need to shed more light about the Bible and about Biblical Hebrew. By translating the Bible to an interpretable, modern language, we will elucidate the Bible, and make it more accessible to those who desire to understand it.
- A need to deepen our understanding of the differences between Biblical and Modern Hebrew. With the appropriate experiments, we can detect fundametnal similarities and differences between the two, such as vocabulary similarity, transormation of words (ancient to modern), etc.
- A need to expand the amount of languages the Bible is translated to (extrapolation). Our models provide framework to translate the Bible to other languages that perhaps do not provide existing translations to the Bible.

Overall, in this work we provide a comprehensive analysis on using three translation models on Biblical to Modern Hebrew, conduct experiments and provide insights that we hope would shed more light on the task of translating the Bible and Biblical Hebrew.

1.1 Issues with Translation

Some of meaning of particular words in the Bible are not actually known with total certainty. Commentators such as Rashi and Ramban interpret the Bible and provide translations. As a result, different biblical translators may translate the Bible not just with different words, but with also fairly different meanings.

2 Related Work

To our knolwedge this task has not been researched before. Therefore, we look to general translation literature and literature related to translated of similar languages, partially building on work reviewed by (Zampieri et al., 2020) overviewing the challenges in translation between dialects. In addition, Harrat et al. (2019) provides a detailed overview of translations between dialetcs of arabic. However, this review is fairly old and does not cite any method that make use of Sequence-to-Sequence models. Laith H. Baniata (2018) uses a more modern approach to translation of arabic dialects by using an encoder-decoder method. The paper used a multi-task learning model to share one decoder among language pairs. There has also been some older literature related to tranlsation between Shaksperian english and modern english (Xu et al., 2012).

While there has not been much literature related to translation between biblical and modern Hebrew, there is a fair amount of literature related to applying NLP methods to Hebrew text. This literature provides general information that is helpful when working with Hebrew text. Tsarfaty et al. (2019) notes some of the challenges with common NLP libraries such as ScaPy in relation to Hebrew.

2.1 History

The Hebrew language was used widely in Israel starting in the 10th century BCE (1000 BC to 901 BC) to 4th century CE. During this period, the language evolved and was even lost for 1400 years. However, the Hebrew used today and the Hebrew from the 10th century BCE are very different. We briefly tell the evolution story of the language.

Archaeological findings of Yossi Garfinkel suggest that Hebrew was used over 3,000 years ago. After many exiles, the Jews stopped using Hebrew language as the spoken langauge. Particuallarly, the Neo-Babylonian Empire conquering of the ancient Kingdom of Judah in the 6th century BCE, resulted in some loss of spoken Hebrew and the use of Aramaic. Once Cyrus the Great leaders of the Greeks conquered Babylon, Jews began to speak Hebrew once again but alongside Aramaic. In fact, one of the most important book to the Jews, the Gemara, is written in Aramaic. At some point, the common language in Israel was Greek. As Jews were dispersed further from Israel, languages used by Jews began to change even more. During the Middle Ages, one of the greatest Jewish Scholars,

Maimonides, actually wrote commentery mostly in Arabic. In Europe, Ashkenazi Jews mainly spoke Yiddish, which actually uses Hebrew characters, and only read Hebrew for religious practices.

In the 19th century, Haskalah (Enlightenment) and Zionist movement began to gain popularity as Jews longed to return to their historical home land. As a result, Eliezer Ben-Yehuda began to revive the Hebrew language to create a usable modern language. Since Biblical and Ancient Hebrew are so old, they do not even contain words for modern items. He belived that the revival of the Hebrew language could unite all Jews in Israel. Using grammar and methods from many other languages, but specifically other semenatic languages, Ben-Yehuda would create what is known as Modern Hebrew. It is the only language that has been successfully revived dead language in history. Today, aproximately 9 million people speak Hebrew around the world. Of these 9 million, approximately 5 million are native Hebrew speakers.[3]

2.2 Hebrew Language 101

Since Hebrew is spoken by few and differs from English substantially, we provide an overview of the basic characteristics of Hebrew to help the reader better understand the problem at hand. The most noticeable difference between Hebrew and English is the alphabet. The Hebrew alphabet is unique to the language[4] and contains 22 letters (consonants, no vowles as explained later on), five of which use different forms when placed at the end of a word. Some consonants can be modified by the addition of a dot to represent a vowel, as an aid to children and those learning Hebrew as a foreign language. Modern Hebrew text, both handed-written and printed, consists of consonants, spaces and western punctuation. Hand-written Hebrew does not join the letters, unlike Arabic and most handwritten English. The Hebrew alphabet is depicted in Figure 1. In addition, Hebrew is written horizontally from right to left, like Arabic. Books open to that, for English native speakers, would be the last page. These differences create many challenges when working with deep learning methods that are primarily built for the Latin alphabet.

Furthermore, Hebrew has *nikud*, 15 diacritical signs used to distinguish between alternative pronunciations of letters and to represent vowels - the Hebrew *nikud* system is depicted in Figure 2. Different combinations of Hebrew letters and Nikkud can create a very diverse writing system. It is worth mentioning that in the past, there was more of a difference when pronouncing certain letters and certain *nikud* charcters. As time passed by, ceratin differences (such as between Alef and Ayin, or between Qamets and Patach) have disappeared, and today they are pronounced almost the same.

Hebrew verb grammar is similar to English in that it has past, present and future tenses, conditionals, imperatives and infinitives. It has the active and passive voice and differentiates between transitive and intransitive. There are some minor differences, however, than can lead to incorrect English verb use. For example, Hebrew does not use the copula to be in the present tense as in English, so beginners may correctly say sentences such as "I happy today!"

Moreover, Hebrew is a much more inflected language than English. For example, Hebrew has more verb endings, nouns and pronouns vary in form according to the preposition that precedes them, and adjectives must agree in number and gender with the nouns they modify. Hebrew has the masculine and feminine genders.

Unlike English, many Hebrew "stop words" are connected to the root word. For example, 'and' is written by appending the letter *vav* (ו) in front of a word, 'from' is written by appending the letter *mem* (מ) in front of a word, 'to' is written by appending the letter *lamid* (ל) in front of a word, 'in' is written by appending the letter *bet* (ב) in front of a word, etc.[5] These features make it difficult for neural networks to learn the langugae and its vocabulary for proper translation.

2.3 Biblical Hebrew Versus Modern Hebrew

The core vocabulary of Modern Hebrew comes from Biblical Hebrew. A large majority of simple nouns and verbs – at least those that existed in the days of the Bible – are the same. But there are also many words in the Hebrew Bible that are not used in Modern Hebrew, and some that are used differently. Moreover, the grammatical structures are different, and so are the tenses, the words' order and the use of *nikud*.

[3] Accoring to the March 18, 2013 Israel Hayom Newsletter

[4] Other languages that were once spoken by Jews such as Aramaic and Yiddish uses the same alphabet. Yiddish is actually based on the German Language. Aramaic is almost a hybrid dialect.

[5] Adding to the challenge of the task at hand, there are no widely used or maintained Hebrew tokenizers, stemmers, or stop-word cleaners.

Mark	Name	Sound	Hebrew	Trans.	Class	Type
אָ	Qamets	"a" as in aqua	קָמַץ	a	Long	A-Type
אַ	Patach	"a" as in aqua	פַּתַח	a	Short	
אָ	Chateph Patach	"a" as in aqua	חָטֵף פַּתַח	a	Reduced	
אָה	Qamets Hey	"a" as in aqua	קָמַץ הֵא	ah	Long	
אֵ	Tsere	"ei" as in eight "e" as in they	צֵרֵי	ei / e	Long	E-Type
אֶ	Segol	"e" as in red	סֶגוֹל	e	Short	
אֶ	Chateph Segol	"e" as in red	חָטֵף סֶגוֹל	e	Reduced	
אֵי	Tsere Yod	"ei" as in eight	צֵרֵי יוֹד	ei	Long	
אֶי	Segol Yod	"ey" as in obey	סֶגוֹל יוֹד	ey	Long	
אִ	Chireq	"ee" as in green	חִירֶק	i	Short	I-Type
אִי	Chireq Yod	"ee" as in green	חִירֶק יוֹד	i	Long	
אֹ	Cholem	"o" as in yellow	חוֹלָם	o	Long	O-Type
אָ	Chateph Qamets	"o" as in yellow	חָטֵף קָמַץ	o	Reduced	
אָ	Qamets Chatuph	"o" as in yellow	קָמַץ חָטוּף	o	Short	
אוֹ	Cholem Vav	"o" as in yellow	חוֹלָם וָו	o	Long	
אֻ	Qibbuts	"u" as in blue	קִבּוּץ	u	Short	U-Type
אוּ	Shureq	"u" as in blue	שׁוּרֶק	u	Long	
אְ	Sheva'	*Vocal*: short "e" *Silent*: no sound	שְׁוָא	e or '	(vocal) Short	

Figure 2: The Hebrew Nikkud and English Pronunciation. (Source: Hebrew4Christian.)

3 Computational Resources

Our models were trained using the free version of Google Colab. Google Colab provides GPUs based on supply and demand, so there is no guarantee that models are trained on the same GPU. At times one can be assigned a Nvidia P100, T4, P4, or K80. Normally, a the worse performing GPUs are assigned for training, since we do not have the payed version. In addition, sessions can be terminated at any point and are limited to 12 hours per day, blocking users if excessive use is detected. As a result, we made the best of our resources. We wanted to complete more experiments with additional models, but were unfortunately prohibited. In addition, we attempted to use pretrained models and prebuilt models so that we do not exhaust our limited computation time. To implement our models we make using of PyTorch and HuggingFace.

4 Dataset

4.1 Collection and Cleaning

Despite the accurate translation, there are many challenges with preprocessing and cleaning the text to use pre-existing models. Since there are nearly no common Hebrew text cleaner Python libraries we carried out our dataset construction and cleaning ourselves by solving the following issues:

1. We need to remove non-translation characters such as editors commentary, numbers, etc.
2. We need to handle strange characters such as multiple attached spaces and replace unknown characters with better representations such as replacing "\u2009" with whitespace.
3. In some cases, we need to reverse the order of sentences since Hebrew goes from right to left.
4. We need to remove *nikud*: transforming יִשְׂרָאֵל to ישראל. *Nikud* is also an area for which we experiment because we are curious to see if any of our models can represent and learn nikud in addition to semantics and structure.

4.2 Data Statistics

Our dataset, which includes Biblical Hebrew sentences and their Modern translations, includes a total of 9222 examples that were collected from 50 Bible sections. It is summarized in Table 1. As previously discussed in Section 2.2, one major difference between Biblical Hebrew and Modern Hebrew is the use of *nikud*, which is the vowel symbol system used to specify pronuciations of words. For example, the word "Israel" is traditionally written as יִשְׂרָאֵל in contrast to today's writing which strips the *nikud* to obtain ישראל . Therefore, we provide two datasets, one of plain Hebrew and the other of Hebrew with *nikud*.

An expected difference between the two datasets is the number of unique characters each contains. This difference is caused by the *nikud* symbols as they are counted as unique characters in addition to conventional letters. For example, יִשְׂרָאֵל has 10 characters while ישראל has only 5. We note that this property leads to single words being represented in multiple ways (via variation in their *nikud*), and results in a far greater number of unique words appearing in the *nikud* dataset. For example, the number of unique words in the no-*nikud* Train Set Source is approximately 40% of the *nikud* counterpart.

We also observe that the fraction of source words that are only seen once in the datasets is greater in the *nikud* dataset. On average, 50% of the words in the plain dataset are seen once while 60% of the *nikud* dataset are singular. The fact that the one-appearance word fraction is approximately equal between the Target datasets sheds light on the lesser use of *nikud* in modern Hebrew. Today it is more common to be using the same *nikud* symbols re-

	Test	Train	Val	Test *nikud*	Train *nikud*	Val *nikud*
Source Max Sentences Length	43	41	32	43	41	32
Target Max Sentences Length	42	41	36	42	41	36
Source Sentences	922	7379	921	922	7379	921
Target Sentences	922	7379	921	922	7379	921
Source Unique Word Count	4380 (66%)	13762 (40%)	4363 (64 %)	7765 (81%)	32915 (50%)	7800 (78%)
Target Unique Word Count	4960 (65%)	15872 (40%)	4976 (64 %)	5270 (67%)	17554 (41%)	5276 (65%)
Source Word Count	12301	100855	12664	12464	102309	12835
Target Word Count	13692	111740	14107	13970	114063	14375
Source Character Count	58588	480209	60376	111155	910549	114587
Target Character Count	67444	550744	69495	113611	926910	116970

Table 1: Statistics for our Biblical Hebrew to Modern Hebrew translation dataset. We report statistics for both the clean dataset (no *nikud* and the raw dataset (with *nikud)*. In parentheses next to Unique Word Counts, we add the fration of these words that appear only once within their set. We note that the *nikud* sets genrally have a higher single-word fractions and more unique words.

gardless of the context of words. This insight might suggest that the *nikud* source dataset contains more contextual information in the words themselves and might be better for learning. Section 4.3 further examines the similarity between the source and the target sentences.

4.3 Base Line and Translation Similarity

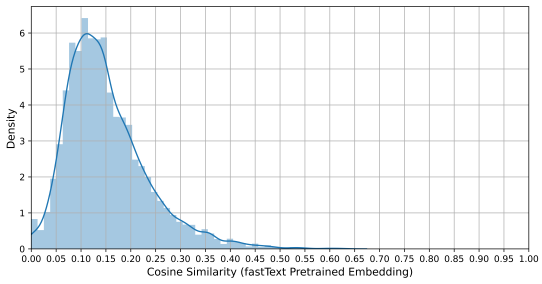
To provide a quantitative sense of the differences of Modern and Biblical Hebrew, we provide two measures of similarity for the source and target: cosine similarity between Hebrew embeddings from pretrained fastText model for Hebrew and the BLEU score for non-translated sources and targets. The latter acts as a baseline BLEU score for our translation task—we want to perform better than this value.

There is a pretrained fastText 300 (Mikolov et al., 2018) Hebrew word embedder from Facebook that is implemented in Pytorch. We use this pretrained model to create feature representations for our source and target sentences. With these feature representations, we can compare the two sentences with a number of metrics. Specifically, we use Cosine Similarity defined as

$$\text{CosineSimilarity}(u,v) = 1 - \frac{u \cdot v}{||u||_2||v||_2} \quad (1)$$

where u adn v are vector embeddings. Figure 3 depicts the distribution of the cosine similarity measures for each source and target sentences. We see that the embeddings are not drastically different from each other as would be the case for two different languages. The mean cosine similarity is 0.1516 with a minimum of just about zero. One should note that the source translates similar texts so that the Biblical Hebrew is more understandable

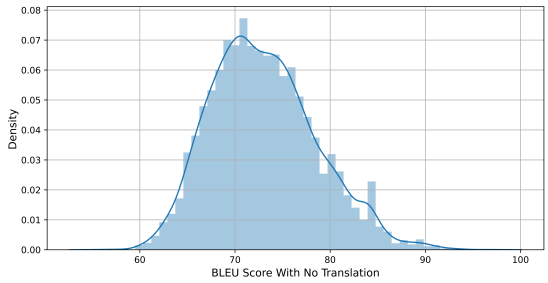
Figure 3: Distribution of Cosine Similarity of sentences with pretrained Hebrew fastText Embedding



to those who speak Modern Hebrew. If a sentence in the Bible does not need translating because the Modern Hebrew translations is the same, translators may still translate the sentence with a different wording. The translator may be reluctant to translate a sentence exact as written in biblical text even if that is a modern translation.

Even though the cosine similarities of the embeddings are fairly low, the BLEU scores are higher, indicating that this translation task may not be straightfoward not because of the difference in words between the sources and targets but because of nuanced word choice. Figure 4 depicts the distribution of BLEU scores for the source and the target sentences. The maximum BLUE score is 97.40, which is extemely high and caused by translations taking on the same form as their sources. The mean of the distribution of BLEU scores is 73.03. The pretrained Hebrew fastText model does poorly in capturing the biblical Hebrew embedding, and the BLEU scores shed light on similar word choices in the translations to the sources.

Figure 4: Distribution of BLEU Scores Between Source and Target



5 Models

There are a number of models used for experiments and comparison in the translation task described in this paper. These models are mainly encoder-decoder models, but we apply a number of variations. The first model we apply is a typical Seq2Seq model with attention. The second model is a fine-tuned pretrained Hebrew-to-Hebrew model with 12 encoders and 12 decoders from Facebook's multilingual model (Fan et al., 2020). In addition, we attempt to use a Seq2Seq model with a copy mechanism with little success.

5.1 End-To-End Transformer Model (ETET)

In sequence-to-sequence problems (for which a neural net transforms a given sequence of elements, such as the sequence of words in a sentence, into another sequence), the initial solutions are based on the use of RNNs in an encoder-decoder architecture. These architectures are limited when working with long sequences, as their ability to retain information from the first elements is lost when new elements are incorporated into the sequence. In the encoder, the hidden state for each step is associated with a specific word from the input sentence. Therefore, if the decoder only accesses the last hidden state of the encoder, it will lose relevant information about the first elements of the sequence. To deal this limitation, the attention mechanism is created.

The attention mechanism is based on the idea that instead of paying attention to the last state of the encoder, in each step of the decoder we look at all the states of the encoder, accessing information about all the elements of the input sequence, calculating a weighted sum of all the past encoder states. This allows the decoder to assign greater weight or importance to certain elements of the input for each element of the output.

But this approach continues to have an important limitation—each sequence must be treated one element at a time, which is very time consuming and computationally inefficient.

The transformer model extract features for each word, using a self-attention mechanism to figure out how important all the other words in the sentence are with respect to that original word. As no recurrent units are used for this process, it is very parallelizable and efficient. Our model is Transformers based, and is based on Vaswani et al. (2017) paper "Attention is all you need" and on Klein et al. (2017) (Harvard's NLP) implementation of it.

As seen in Figure 5, there is an encoder model on the left side and the decoder on the right one. The encoder maps an input sequence of symbol representations $(x_1, ..., x_n)$ to a sequence of continuous representations $z = (z_1, ..., z_n)$. Given z, the decoder can then generate an output sequence $(y_1, ..., y_n)$ of symbols, one element at a time. At each time step, the model consumes the previously generated symbols as an additional input to generate the next. The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder.

The encoder is composed of a stack of 6 identical layers. Each layer has two sub-layers. The first sub-layer is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. Residual connections are employed around each of the two sub-layers followed by layer normalization. The decoder is also composed of a stack of 6 identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to the encoder, we employ residual connections around each of the sub-layers, followed by layer normalization. Figure 6 presents the attention mask that shows the position each target word (row) is allowed to look at (column). Words are blocked for attending to future words during training.

As explained before, the attention maps a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the

Figure 5: Seq2Seq Attention Model (Source: Vaswani et al. (2017))

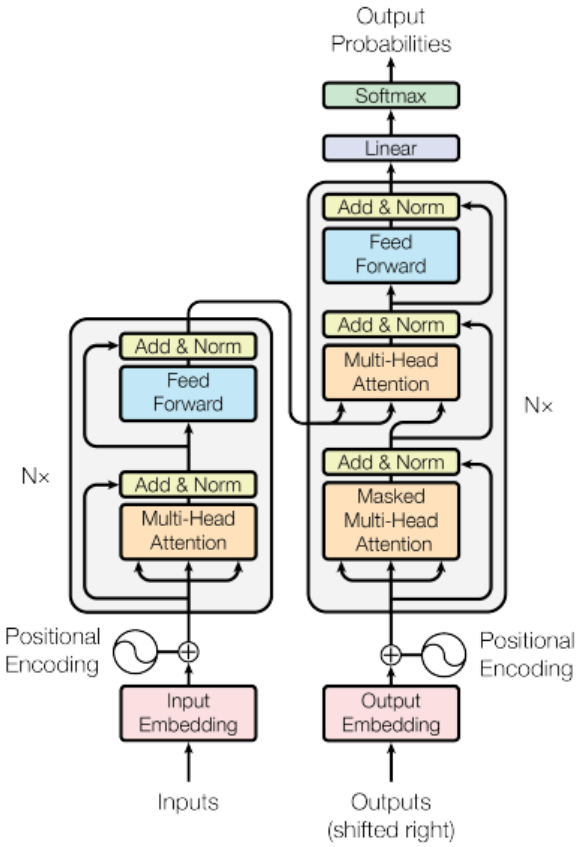


Figure 6: Attention Mask (Source: Vaswani et al. (2017))

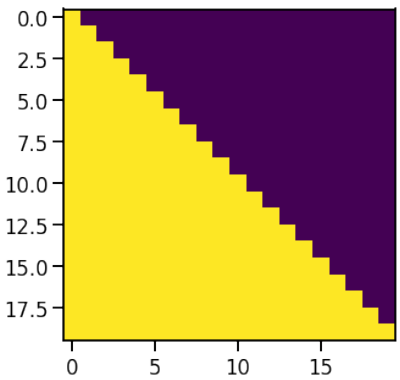
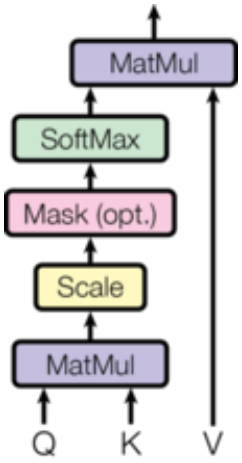


Figure 7: This Figure depicts the scaled dot-product attention. (Source: Vaswani et al. (2017))



values, where the weight of each value is computed by a compatibility function of the query with the corresponding key.

The particular attention used here is a scaled dot-product attention and is depicted in Figure 7. The input consists of queries and keys of dimension d_k, and values of dimension d_v. Then dot products of the query with all keys are computed, each is divided by $\sqrt{d_k}$, and a softmax function is applied to obtain the weights on the values. The use of multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. Due to the reduced dimension of each head, the total computational cost is similar to that of single-head attention with full dimensionality.

In addition to attention sub-layers, each of the layers in the encoder and decoder contains a fully connected feed-forward network, which is applied to each position separately and identically. This consists of two linear transformations with a ReLU activation in between.

Since tokenization for Hebrew is not ideal, a custom tokenizing function based on "Pytorch's NLP From Scratch" (Robertson, 2021) was created. Each word was represented as a one-hot vector, and in order to keep track of one-hot encoding a mapping of words to indexes was created and a word counter to use later to replace rare words.

Learned embeddings—the usual learned linear transformation and softmax function—are used to convert the input tokens and output tokens to vectors of the dimension of the model. In addition, they are used to convert the decoder output to pre-

dicted next-token probabilities.

Since the model contains no recurrence and no convolution, we must provide the model information about the relative or absolute position of the tokens in the sequence for the model to make use of the order of the sequence. To address that issue, "positional encodings" were added to the input embeddings which are located at the bottoms of the encoder and decoder stacks. There are many choices of positional encodings, we chose to use sine and cosine functions of different frequencies.

5.2 Fine-Tuned Multilingual Transformer Model (FTMT)

Recently, Facebook has designed a state-of-the-art multilingual translation model which focuses on non-english centric translations (Fan et al., 2020). The team created "a true Many-to-Many multilingual translation model that can translate directly between any pair of 100 languages" (Fan et al., 2020). This model uses attention and transformers with 12 encoder and 12 decoder layers for a total of 1.2 billion parameters. Since this model is designed for multilingual translaiton, it requires a special prefix language ID token for both the source and target text. Additional details on the model construction and implementation can be found in Fan et al. (2020).

With our aim creating a Hebrew-to-Hebrew task, we choose to use this model and take advantage of the state-of-the-art decoding and encoding models and pretraining. A significant factor that lead to our choice is the uniqueness of this pretrained Hebrew encoding *and* decoding model. Tokenization that supportt Hebrew inputs are uncommon and not avaliable for many other models. This pretrained model, which is provided by Facebook and implemented by HuggingFace, allowed us to achieve promising results with minimal domain-specific tranining.[6] This model is pretrained with 7.5 billion training sentences from 100 languages. Given we have under 10,000 Hebrew sentences, we fine-tune this model to be able to translate from Biblical Hebrew to Modern Hebrew.

To train this model, we use an AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 0.00001. A number of different learning rates were tested and ranked by the performance on our validation set. We found that with this learning rate and optimizer, the model acheived very promising performance on the validation set after just 4 epochs The loss function used is a negative log likelihood loss implemented in HuggingFace as `NllLossBackward`. In total, we train this model for only 10 epochs because of our limited computational resouorces described in Section 3.

[6] https://huggingface.co/transformers/model_doc/m2m_100.html

5.3 CopyNet

In simple terms, a Copy Mechanism is a mechanism on the decoder of a Seq2Seq model used for translation that copies words from the source sentence and uses them in the output sentence. Intuitively, we beleived that the CopyNet architecture could acheive promising perfromance since many of the words in the Modern Hebrew and Biblical Hebrew are the same (see Fig. 4 for an illustration). To implement this model, we followed the implementation described in Gu et al. (2016) and even attempted to modify a number of implementatations of CopyNet found throughout the web. While performance was able to improve from the mean BLEU of 0.000265, it never reached a score above 0.1. A number of different training procedured were used, but the model seemed to perform poorly. One reason for this was the inadequete GPU memory to hold the copy vocabulary size needed for Hebrew which has many varation of words as described in Section 2.2. Because of this lackluster performance, we decided on focusing on the first two methods that could be trained with our limited resources.

6 Experimental Evaluation

To determine the efficacy of our methods for translating between Bibblical Hebrew and Modern Hebrew we conducted three classes of experiments. In this section we discuss our experimntal evaluation of the training process of our two models on plain and *nikud* Hebrew inputs, present quantitative translation results, and finally qualitatively discuss the quality of translations of free-input text.

6.1 Experiments

The first and most basic experiment we conduct is an observation of the evolution of the training losses reported when training our models. When fitting the parameters of our models to fit the training samples, we interpret a decreasing trend in the losses as successful learning. We train and evaluate our models on the data specified in Table 1,

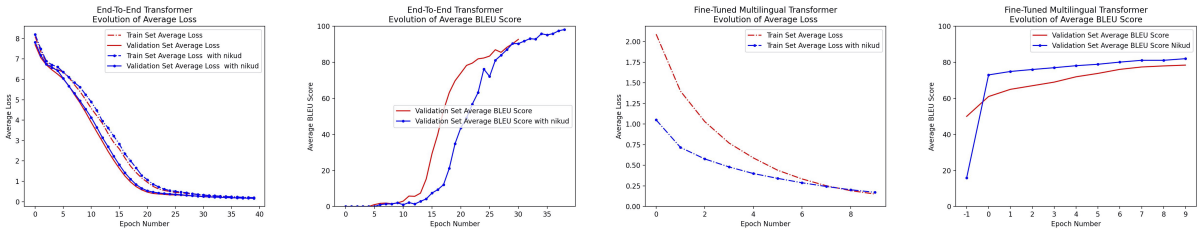


Figure 8: Experimental results. We illustrate our models' ability to learn Hebrew-to-Hebrew translations from labeled data and provide a quantitative evaluation of the quality of our translations. Our End-To-End Transformer (ETET) and Fine-Tuned Multilingual Transformer (FTMT) models both show a decreasing loss evolution during training and an increasing associated BLEU score. Given that the BLEU scores reported are generated from previously unseen test data, these properties suggest that our models can learn effective translations between Biblical and Modern Hebrew texts. Values at epoch -1 signifies the BLEU score achieved by FTMT before any fine-tuning took place.

which includes our Hebrew-to-Hebrew dataset and a variant of that dataset that includes *nikud*.

To complement the analysis of the learning in our models, we further wish to understand if the learning that took place has been successful for the task at hand (Hebrew-Hebrew translation). To this end, we use the BLEU score metric to quantify the quality of the outputted translations. We compare our models' translations of Biblical Hebrew sentences with reference translations to recover the BLEU scores. Finally, to verify that our learning pipeline can also generalize to novel inputs and is robust to adversarial queries, we test our models' performance qualitatively on free-text inputs. We evaluate the outputs of our models that are generated for inputs of Hebrew sentences from different modalities. By using from the Bible, from poetry, and simply arbitrary sentences, we wish to understand qualitatively if our models are robust.

The following sections detail the results of our evaluation. Section 6.1.1 discusses the training-loss evolution of our models, Section 6.1.2 presents our BLEU score results, and Section 6.1.3 covers our observed qualitative results from translating free-text inputs.

6.1.1 Trainig Translators

As illustrated in Fig. 8, we observe that both of our models have successfully learned to reduce their training loss when trained on Biblical Hebrew sentences alongside their Modern Hebrew translations as supervision. The loss evolution that appeard when training our End-To-End Transformer model (ETET) shows a very similar rate of loss change over time for the Train set and the Validation set when training on plain Hebrew and *nikud* Hebrew. It could be that ETET showcased a similar loss evolution for both variants of Hebrew because its randomly initialized state was sufficiently unbiased to treat the different datasets as realtively similar.

On the other hand, our Fine-Tuned Multilingual Transformer (FTMT) model did behave differently when trained on plain and *nikud* Hebrew inputs. Perhaps counter-intuitively, FTMT showed a higher initial loss when trained on plain Hebrew despite being pretrained on this type of language.

6.1.2 Translation Quality

Through evaluating the BLEU scores of the translated sentences with reference translations we conclude that both of our models have learned to successfully perform Biblical Hebrew translation. Our model ETET achieved BLEU scores of 92% and 98% for plain and *nikud* Hebrew repsectively. Such high scores could point to lack of diversity in our dataset. Given that our target translations are all from the same source it could be that our Test and Train sets have sufficiently similar statistical occurrence of sentences and translations to facilitate near-perfect translation.

We gain insights into the effects of pretraining on large amounts of data through the translation quality evaluation for the FTMT model. We first note that the initial performance score FTMT achieves without *any* Fine-Tuning is 50% for plain Hebrew and 15% for *nikud*. This an expected result given that FTMT was pretrained on plain Hebrew. We additionally see the main two benefits of pretraining: fast adaptation to new data and robustness to overfitting. The first is seen in the dramatic jump in the BLEU scores for both the *nikud* Hebrew trained model and the second is seen in the monotonic increase of the scores in the following epochs for both training modalities. It seems as if the pretrained

Input Sentence / Input Sentence English / Model Name						
	בְּרֵאשִׁית בָּרָא אֱלֹהִים אֵת הַשָּׁמַיִם וְאֵת הָאָרֶץ In the beginning God created the heaven and the earth.	בראשית ברא אלהים את השמים ואת הארץ In the beginning God created the heaven and the earth.	הָהָר הַגָּדוֹל הוּא הַהָר הַכִּי גָּבוֹהַ The biggest mountain is the tallest mountain.	ההר הגדול הוא ההר הכי גבוה The biggest mountain is the tallest mountain.	הִנָּךְ יָפָה רַעְיָתִי, הִנָּךְ יָפָה עֵינַיִךְ יוֹנִים. You are beautiful my wife, you are beautiful your eyes like doves.	הנך יפה רעיתי, הנך יפה עיניך יונים. You are beautiful my wife, you are beautiful your eyes like doves.
FTMT	הבהמות באותן העולות באותן העולות באותן העולות ... The kettle come the climbing in their climbing (repeats).	בהתחלה ברא אלוהים השמים את הארץ ואת At the beginning God created the heaven the earth and the	הכהן הכהן הכהן הכהן הכהן הכהן The Cohen (repeats).	הגדול ההר הוא הגבוהה ההר The big mountain is the tallest mountain.	יפאה היין שפיכת רעיונותי את יפיך היין שפיכה יפאה היין שפיכת את יפיך הוא שלך היין שפיכה Beutiful Wine Shedding Ideas The Wine Shedding...	יפה רעיתי יפה עיניך יונים My beautiful wife beautiful your eyes like pigeons.
FTMT Nikud	כְּסַכֵּם אֶת בָּרָא הוּא אֱלֹהִים וְאֵת הַשָּׁמַיִם וְאֵת הָאָרֶץ Agreement God created the heavens and the earth	בְּבַקָּשָׁה בָּהֶם אֱלֹהִים הָיוּ וְהַשֵּׁם Please in them there were gods and the name	הַהָר הַגָּדוֹל הוּא הַהָר הָרֵי גְּבוֹהּ The biggest mountain is the mountain mountains tall.	הַבַּעַל בְּעַל הַבֵּן בְּעַל הוּא The husband son husband is.	רַעְיָתִי אֶת יָפָה הִנָּךְ יָפָה עֵינַיִךְ יוֹנִים My wife you are beautiful your eyes like doves.	מְקוֹמָם חֲבֵרוֹת חֲבֵרוֹת חֲבֵרוֹת חֲבֵרוֹת חֲבֵרוֹת Their place companies companies...
ETET	No output.	לְחֹלְקִים אוֹתָהּ וְיַחֲלֹק מֵהַעוֹר הָאָרֶץ וְאֵת הָאָרֶץ וְאֵת הָאָרֶץ וְאֵת To parts he'll divide it from the skin of the land and the land and the land...	No output.	No output.	No output.	No output.
ETET Nikud	חָטָאתָ אֶת פַּר אֶת וְיָשָׁחַט מִשְׁפַּחְתּוֹ וְעַל עַצְמוֹ עַל וְיְכַפֵּר הָאָרֶץ אֶת יִגְיֹשׁ You have sinned for kettle and butcher its family and on himself will forgive the land will be given.	No output.	No output.	No output.	No output.	No output.

Figure 9: Three sample sentences (with *nikud* and without) translated with out two models. English translations are added next to the Hebrew texts. We observe that the Fine-Tuned model FTMT produces more robust translations when compared to the End-to-End model ETET, which is limited in its vocabulary and training. Unsurprisingly, free-text that include *nikud* are generally translated better by models that were trained on text with *nikud*.

parameters were positioned sufficiently well in the parameter space to allow fast refinements instead of a convergence to an unfavorable minima.

6.1.3 Free Text Translation

We report qualitative results on translating free-text inputs using all four variants of our models (ETET, ETET-*nikud*, FTMT, FTMT-*nikud*). Example sentences taken from Biblical and poetry sources alongside Modren Hebrew inputs are included in Fig. 9. Sensibly, the models that were trained on plain Hebrew struggle with recovering good translations for *nikud* inputs and vice versa. Given that *nikud* is encoded as extra unicode characters in words this result is expected.

Focusing on ETET, we observe that a large fraction of the inputs failed to produce an output. The reason for these failures is the small vocabulary size of ETET that is only extracted from our dataset and leads to novel input words being unable to be tokenized. This is a severe drawback of our End-To-End model.

On the other hand, we observed that FTMT (both trained on plain language and *nikud*) has successfully handled novel inputs and produced reasonable outputs. Such robustness can be attributed to the extensive pretraining this model has undergone and to the newly introduced Hebrew tokenization that was released with the M2M100 model. This result suggests that our FTMT model specialized to perform well on the translation task (achieving a BLEU score of 82%) without losing the ability to handle arbitrary inputs.

7 Discussion

Through evaluating our two models on the task of Ancient Hebrew to Modern Hebrew translation we have observed the important role large datasets play in the construction of effective language models. In our experiments we have shown that learning a language model end-to-end solely from a small dataste yields a model achieving impressive translation BLEU scores that fails to adapt to samples outside the dataset. Our ETET model overfit to the nature of the data in the dataset and suffered from a small vocabulary.

Furthemore, the role of pretraining has become increasingly clear in the development of our FTMT model. The adaptation of an established language model to a specific task was facilitated for the FTMT model's precomputed initial parameters. Since both our transformer models are very large, it is intuitive to believe that small amounts of training data would not be able to excite all of their parameters sufficiently to induce meaningful learning.

Reliance on past training is important.

8 Conclusion

In this work we have presented two methods to successfully perform Biblical Hebrew to Modern Hebrew translation, achieving BLEU scores of 82% and 98%. We have presented the nuances of this translation problem alongside the origins and uniqueness of the Hebrew language and provided insights into potential factors contributing to the successes and drawbacks of our methods.

In future works, we will like to further experiment with a Copy Mechansism to see if it can work. In addition, our corpus only includes the Torah, not the prophets or scrolls. With these additional texts, we can hopefully have more trianing data and improve the models performance. In addition, a website that allows for translation of biblical Hebrew in a child-friendly format will allow students all over Israel to use our reseach to improve thier learning.

References

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. *CoRR*, abs/2010.11125.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *CoRR*, abs/1603.06393.

Salima Harrat, Karima Meftouh, and Kamel Smaili. 2019. Machine translation for arabic dialects (survey). *Information Processing Management*, 56(2):262–273. Advance Arabic Natural Language Processing (ANLP) and its Applications.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation.

Seong-Bae Park Laith H. Baniata, Seyoung Park. 2018. A neural machine translation model for arabic dialects that utilizes multitask learning (mtl). *Computational Intelligence and Neuroscience*, 2018.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Sean Robertson. 2021. Nlp from scratch: Translation with a sequence to sequence network and attention.

Reut Tsarfaty, Amit Seker, Shoval Sadde, and Stav Klein. 2019. What's wrong with hebrew nlp? and how to make it right. *CoRR*, abs/1908.05453.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of COLING 2012*, pages 2899–2914, Mumbai, India. The COLING 2012 Organizing Committee.

Marcos Zampieri, Preslav Nakov, and Yves Scherrer. 2020. Natural language processing for similar languages, varieties, and dialects: A survey. *Natural Language Engineering*, 26(6):595–612.

Acknowledgments

Thank you for a wonderful semester!