**NLP Final Project Status report**

Doron Hazan (doronh), Zachary Metzman (zmetzman), Dana Rosenfarb (danarose), Yorai Shaoul (yorai)

We propose creating a **machine translation model between Ancient Hebrew and Modern Hebrew**. 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, to bridge the gap between Ancient Hebrew and Modern Hebrew, we propose a machine translation model.

One important aspect of translating Ancient Hebrew to Modern Hebrew is to create a **reliable dataset** that would guarantee or maximize the true translation mapping. To do this task, we looked at various sources that offer bible translations. Most of the authors’ native language is Hebrew and have experience reading the bible 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).[[1]](#footnote-0) This source is publicly available on Wikitext and includes translations that we believe are widely acceptable within the Hebrew community. Despite the accurate translation, there are many challenges with preprocessing and cleaning the text to use pre-existing models. We will elaborate extensively on all these changes in the final report, but some issues we faced are: 1) removing non-translations such as commentary, random numbers, etc. 2) dealing with multiple attached spaces 3) replacing unknown characters with better representations such as replacing “\u2009” with whitespace 4) changing the order of sentences (Hebrew goes from right to left), which is experimental (We are curious to see the outcomes of reversed vs non-reversed sentences per model.) 5) removing nikud (Ancient Hebrew and Modern Hebrew include nikud, which is a vowel system not of letters but of diacritical signs used to distinguish between alternative pronunciations of letters. It looks like this: אִזְמַרְגָּד) which is also experimental. (We are curious to see if any of our models can represent and learn nikud in addition to semantics and structure.)

Since Modern Hebrew and Ancient Hebrew share many of the same words, it was suggested that we explore a **copy mechanism** and have been experimenting with CopyNet.[[2]](#footnote-1) We found an older implementation of the model, and we are modifying it to suit our needs. So far, there have been some issues with training. There is a vocabulary size parameter that should be in the tens of thousands, but with a vocab size of that magnitude it is difficult to find a setting that does not cause a CUDA memory error. With a smaller vocab size, say 2000, the validation BLEU scores are in the thousands place. In addition, while the BLEU on the validation set may be increasing slightly, the validation loss is increasing too. Given the time for this project and the resources available, a CopyNet implementation with experiments may not be viable. Instead, experiments involving **transfer learning of pretrained seq2seq models from PyTorch seems far more doable**. It may be interesting to see if certain pretrained translation models allow for faster convergence with the two types of Hebrew. If so, it may suggest that there are abstract language concepts that the model captures that are related to Hebrew despite its major differences. In addition, we are considering trying GPT-2.

Based on “Attention Is All You Need” by A.Vaswani et al.[[3]](#footnote-2) and a Harvard NLP article,[[4]](#footnote-3) **we have also implemented a transformer model**. In our model, we use stacked self-attention and pointwise-fully-connected layers for both the encoder and the decoder. The encoder maps an input sequence of symbol representations to a sequence of continuous representations . Given , the decoder can then generate an output sequence 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. Since tokenization for Hebrew is not available, we wrote a custom function based on Pytorch’s “NLP From Scratch: Translation With a Sequence to Sequence Network and Attention” article.[[5]](#footnote-4) **Our model is capable of outputting seemingly good translations with about 1 hour of training time.**

Since our work on setting up **meaningful** **validation mechanisms is still in progress**, we have so far been testing our models qualitatively, for the most part. Observing progressively lower loss values for our transformer model, we have verified its efficacy via comparing target translations to source translations and by translating free-text inputs and examining the translated output. Given the progressively lower loss values from the training cycles, we believe that our model has learned *something*. **Initial experiments show proper word-choice for Modern Hebrew translations and an overall similarity to the target translations**.

In the remainder of the semester, we intend to implement a central evaluation pipeline and rigorously evaluate our models as well as performing ablation studies to shed light on their abilities. We intend on using the BLEU metric to evaluate translation quality. Finally, we are interested in knowing whether different training modalities and different inputs to the model would affect the quality of our machine-translations.

As hinted at in the introduction, the Hebrew language is unique in its complexity. Apart from having evolved over the years, the way its vowels are used has also changed. For instance today, in contrast to older times, the contemporary Hebrew language does not include vowels. As a result, the Ancient Hebrew language might seem foreign to a modern reader.

**Our additional tests will dive deeper into the minutia of the language**. The second set of tests will evaluate the **significance of including vowels** in our training data. From a first glance, it is sensible to believe that any additional information could contribute to the training procedure. However, as discussed in the CopyNet section, every pair of words that uses the same letters but with different vowels is considered different. A single word or a collection of letters could have multiple versions that differ only in their vowel-structure. This could lead to our corpus being shallow--lacking sufficient token repetition.

Another area of interest is **incorporating Modern Hebrew translations from more sources into our training dataset**. We may obtain more recent books (and their “translations” to simplified language) and add them to our training corpuses. This could lead to better embeddings of ancient words that are still in use and reinforce the retention of that language.

1. https://bit.ly/3vE0ijo [↑](#footnote-ref-0)
2. https://arxiv.org/pdf/1603.06393.pdf [↑](#footnote-ref-1)
3. https://arxiv.org/pdf/1706.03762.pdf [↑](#footnote-ref-2)
4. http://nlp.seas.harvard.edu/2018/04/03/attention.html [↑](#footnote-ref-3)
5. https://pytorch.org/tutorials/intermediate/seq2seq\_translation\_tutorial.html [↑](#footnote-ref-4)