Fall 2020 Introduction to Natural Language Processing
Project Final Report

A. Project Team members (list team members here)

1. Ethan Mendel
2. Sam Perlmutter

B. Project Topic Introduction (2 points) : e.g. NLG or MT (5-10 sentences)

What are you attempting to do? Why is it worth doing?

We did our project on Machine Translation, more specifically translating Hebrew to English. We are attempting to simultaneously build a translation model and learn how different training sessions produce different models. This project is worth doing to help us better understand how models train and how differently sized training data and time affect the overall effectiveness of the models.

C. Prior work (4 points) (10-30 sentences)**:** Each group member must contribute at least 1 source (citations, resources, links) based on prior work in this topic area.

Sam Perlmutter: I found multiple datasets of English-Hebrew sentence pairs to use a training set – http://casmacat.eu/corpus/global-voices.html | http://opus.nlpl.eu/Wikipedia.php. I also read a paper about machine translation between Arabic and Hebrew for insight on datasets and semtitic languages – https://groups.csail.mit.edu/sls/publications/2016/Belinkov_SeMaT02_2016.pdf

Ethan Mendel: I read through googletrans API to help build a hebrew corpus as a starting point and a checker for our English/Hebrew machine translation - https://pypi.org/project/googletrans/. I also found a couple guides on how to create different language models - https://medium.com/@ageitgey/build-your-own-google-translate-quality-machine-translation-system-d7dc274bd476, https://machinelearningmastery.com/develop-neural-machine-translation-system-keras/, https://towardsdatascience.com/neural-machine-translation-with-python-c2f0a34f7dd, https://github.com/prateekjoshi565/machine_translation/blob/master/german_to_english.ipynb

Next: Indicate which of the ideas was selected to continue for the project.

We have decided to develop a Machine Translation system by using a simple Seq-2-Seq model through the Keras library.

D. Data sources (1 points) (10-20 sentences)**:** Each group member must propose at least 1 data source for the chosen topic and document it with at least 1 source.

Sam Perlmutter: http://casmacat.eu/corpus/global-voices.html | http://opus.nlpl.eu/Wikipedia.php

Ethan Mendel: https://tatoeba.org/eng

Next: Indicate which of the data sources was selected to continue for the project.

We are going to use the sentence pairs form Tatoeba because the Wiki sentences are 1) not always correct translations, and 2) contain some non-hebrew translations.

Tatoeba gave us 126,335 sentence pairs that we aggregated together.

E. Approach (5 points) (10-40 sentences):

Explain what is your solution and how you did it. Which techniques did you use?
How did you measure performance?
Also explain team member contributions here: which team member contributed to which part of the problem (make a table here to be concise),
include any relevant and specific info. e.g. software packages and data used

We downloaded the English sentences, Hebrew sentences, and sentence links files from Tatoeba. In preprocessing, we read through the sentence links files and searched through the English and Hebrew sentence files to link sentences together. When we found a link, we removed punctuation and converted the English sentences to lower case. We removed punctuation because English is a SVO language, and Hebrew can either be SVO or VSO (additional details on the syntax discussion). This means that punctuation might not always correspond, and could throw the model off. We converted the English sentences to lower case because Hebrew does not have casing.

After the initial preprocessing of the data we split each sentence by spaces, got the max sentence lengths, and used Keras to create both English and Hebrew tokenizers. We then split the data into testing and training sets, encoded them, and trained our Seq-2-Seq model using Keras.models.Sequential.fit.

We measured performance using the nltk library’s BLEU and METEOR scores because they provide good metrics for evaluating the effectiveness of translation models. BLEU score evaluates the quality of the text translated and METEOR score is an average of recall and precision scores where recall is more heavily weighted.

Part of Project	Preprocessing	Training	Evaluation
Done by	Ethan & Sam	Ethan	Sam

F. Results (Very important!!) (15 points, unlimited sentences)

Include relevant observations, measurements, and statistics.

Include graphs, equations, pictures, etc. as appropriate

Model 1 - 50,000 pairs; 6 epochs

Average BLEU score: 1.8345392759940913e-159

Average METEOR score: 0.26234439726940284

Model 2 - 50,000 pairs; 30 epochs

Average BLEU score: 1.8345392759940913e-159

Average METEOR score: 0.3059985264735084

Model 3 - 50,000 pairs; 90 epochs

Average BLEU score: 5.8705256831810946e-158

Average METEOR score: 0.269963344988355

Overview

None of our models were particularly effective. In general, every prediction was translated as “I is,” or “I is to,” or some other variation of short words that don’t form any meaningful or comprehensive sentences. We believe the issue was that we did not have enough data for the models to learn enough aspects of the languages. We also underestimated the amount of computing power required to train the models, coupled with both of our lack of experience in this field, caused us to spend more time getting our models to train instead of actually training and improving our code.

G. Summary (10-20 sentences) (3 points)
Try to draw together the Introduction, Prior Work, Approach and Results sections.
(They may appear to be disjoint sections to an unfamiliar reader).
Restate important results and your conclusions.

We built a translation model to translate from Hebrew to English. After reading through a few examples online, we decided to make a few Seq-2-Seq models (using a Keras, a deep learning library) and compare how different parameters changed the translations.

We got 126,335 Hebrew-English sentence pairs and started training. We used the first 50,000 sentence pairs to train three models over six, thirty, and ninety epochs (iterations), saving each model that had the least loss.

Overall, the results were very underwhelming.

H. Retrospective

Looking back at our approach, we have identified a few mistakes we made along the way.

We spent a lot of time preprocessing trying to link sentences together. At the time of preprocessing, we had not found the pre-linked sentence files Tatoeba has elsewhere on their site.

Additionally we realized each model we trained, a new, random test/train split was created. This has the possibility of leading to vastly different models and even vocabularies. Looking back we should have made the split, saved them separately, and used these files for training consecutive models.

Lastly, we would have liked to use more sentence pairs to see how morer data could help our models learn the language better. With more time and our now *relative* experience, we think we could have built better models if we were to do this project again.