**Fine Tuning AlephBERT**

*Background*

The goal of our project is to fine tune a BERT model in Hebrew so that it can be employed in sentence completion tasks. As stated in Wikipedia, BERT is a “transformer-based machine learning technique for natural language processing (NLP) pre-training developed by google.” BERT is trained on a very large corpus of text and is then fine-tuned for specific NLP tasks.

Researchers at Bar Ilan University approached the task of building a new model that would be trained on Hebrew texts. They called their model AlephBERT. We have taken their model and fine-tuned it for our task (sentence completion). We take one text corpus at a time, and using it, fine tune our model.

Our fine tuner program trains the model on the text, sentence by sentence. A word in the sentence is randomly chosen as a mask during the training and test stages. The mask serves as a label.

What results from each fine-tuning session is a pickle file that contains the weights of the new masked language model. We take this output file and use it as the input to a new fine-tuning session, where this input is improved using a new text corpus. The fine tuning takes place in the file called “fine\_tune\_model.ipny”. We run this program using the google colaboratory.

*Sentence Completion Task*

One way to make our lives easier is to make speech easier. Sometimes we are at a loss for the right word in a sentence, and we don’t want to spend time and energy trying to figure out the perfect word. We might also want our devices to complete our sentences for us in order to save us time and energy typing or speaking.

This is where sentence completion comes in. Given a sentence, we take one word in the sentence- which we call the mask- and replace it with the word that is very compatible with the rest of the sentence. For example, we have the sentence “it is very nice outside.” We select any word in the sentence, say we choose “nice”, and designate it as the mask. We then evaluate our model on the rest of the sentence, and it outputs the most likely words.

The BERT model accepts a sentence with a mask. It then breaks down the sentence into words. Each word is “tokenized.” This means that we derive the root of the word. For example, we might have the word “running,” which would be tokenized to the word “run.” Each word (or at least many words) in our language is given a corresponding key (usually in the form of a number). At this point we have a vector of keys. This vector is evaluated by our model, and it outputs the word that should be in place of the mask.

*User Interface*

We run the server on localhost, port 8080. We use the react library for our front end, and the file we run when this page is opened is called app.js. When the user opens the page, he sees two columns- one for English sentence completion and one for Hebrew. The user is asked to type a sentence. After he submits the sentence, he chooses a word from that sentence that serves as the mask. The sentence and mask are sent to the server.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, צג, שחור

התיאור נוצר באופן אוטומטי

The server, running a JavaScript program processes this submission. This program, named index.js, determines the language of the sentence. It then runs a python script. There are two such scripts- one for each of the English and Hebrew models (english-model and hebrew-model, respectively.) The sentence and mask are sent to the corresponding script. This python program runs the BERT model using our most updated pickle file- meaning, the most sophisticated (or accurate) model (as of now the most recent file is tune\_6\_10.pkl). The server then returns the ten most likely words, with their probability of being the correct word.

The client receives these words and probabilities, and they are presented in two columns, where each row contains a word and its corresponding probability. The words are ordered by their probabilities.

תמונה שמכילה טקסט, צילום מסך, שחור, מסך

התיאור נוצר באופן אוטומטי

The user can also choose to use the original model (the one that isn’t fine-tuned to our sentence completion task). We added this option so that the user can see the difference between the models in terms of their results. As you can see in the picture below, the “Use Original Model” button has been pressed.

תמונה שמכילה טקסט, צילום מסך, צג, שחור

התיאור נוצר באופן אוטומטי

*Setup*



If the app is running but nothing opened in the browser please follow the instructions on the readme file of the project at branch “production”.

To receive tune\_6\_10.pkl please contact us through our github account.