Victoria Zabel

Assignment 5: Learning Portfolio

What is an NLP Model?
NLP model stands for natural language processing and covers aspects of the machine learning and linguistics field. The overall goal is to understand the context of words. Common NLP tasks range from classifying words in a sentence, generating text context given a certain input, as well as generating sentences from text (such as summarizing). Additionally, NLP models can be used for speech recognition or describing images.

What are the challenges of NLP Models?
The main challenge of these types of models are how to present or formulate the language in a way that a machine could learn from, as the process would greatly differ to how humans perceive language inputs.

What is a pre-trained NLP model?
A pre-trained NLP model are deep learning models that has previously been trained on large datasets to perform NLP tasks like the ones mentioned above. These models receive various inputs of the datasets in order to recognize the context or relationships of the given input. Being pre-trained, these models have the advantage of saving time. One example of an NLP model are “transformers”.

How do I load them?
NLP models can be loaded by using the Transformers library by Hugging Face in Python. This can be done with a basic line of code such as:

```
!pip install transformers
from transformers import pipeline
```

This line allows us to create prompts and phrases allowing the model to learn. Using the pipeline function has three phases:
Pre-Processing: first, text is converted into a format that is understandable for the model.
Processing: here, the preprocessed inputs are fed to the model.
Post-Processing: the model’s predictions are processed here in a coherent and understandable way.

Commonly known Transformers are GoogleAI or GPT-4.

What is tokenization?
As previously mentioned, NLP models perceive text inputs differently than humans do, meaning with numbers rather than the actual individual words. As such, prompts that are used as inputs need to be converted from text to numerical values, which can be achieved by tokenization. One example of this is splitting a line of text into parts according to either punctuation or spacing. This can be done using the split() function:

```
# text = I wanna go on vacation split by spacing
tokenized = "I wanna go on vacation".split()
print(tokenized)

# = ['I', 'wanna', 'go', 'on', ‘vacation']
```

For tokenizers with large vocabularies, words will be assigned an individual ID in order to be identifiable by the model. Sometimes unknown tokens ([UNK] / “”) will be returned, which is something that should be avoided. This can be avoided using a different type of tokenization, namely “character based” tokenizing, in which each letter or character of text is given an ID. This decreases the size of the vocabulary and the number of unknown tokens.
Sub-word tokenization is the third type, which combines elements of the previous two. Here, words and phrases are categorized into words and rare words. In rare words, the suffix will be tokenized by itself. Meaning that beautifully would be tokenized as “beautiful” and “lly”.

What does fine-tuning mean?
The transformers model allows us to “fine-tune” pre-trained models by using its Trainer class. Fine-tuning is used for models that have already been trained for certain tasks and purposes. In this case, the models will be trained and tweaked additionally to perform another but similar task. By importing transformer’s TrainingArguments for example, we can fine tune the desired model by training and later evaluating it.

What types of NLP Models are there?
There are three main types of NLP models: encoder, decoder, and sequence to sequence models. Encoder models are mainly used for sentence classification, named entity recognition, as well as question answering. Known encoder models include ALBERT, BERT, or RoBERTa. These types of models use only the encoder level of Transformers and try to mask certain words in a sentence, for example, and training the model to reconstruct the input.

On the other hand decoder models, such as CTRL, GPT-4, and Transformer XL, are used for text generating and use the decoder of a Transformer. Here, the next word of a sentence is predicted.

Sequence to sequence models, also known as encoder-decoder models, cover tasks such as summarizing text, translating, or generative question answering, meaning that they are able to generate new sentences or text according to inputs. Examples of these models include BART, mBART, or T5.

What possibilities do I have with the Transformers package?
The Transformers package allows us to solve a variety of NLP tasks as well as creating or using existing models. Using the previously mentioned pipelines of the transformer package, such as ner (named entity recognition), summarization, text generation, or question answering, we are able to fulfill NLP tasks.