2.5-putting-all-together

Giriş

Önceki bölümlerde bütün aşamalar manuel olarak gerçekleştirildi. Tokenizer'ların nasıl çalıştığından, padding, truncation ve attention mask kavramlarından bahsedildi. Fakat HuggingFace Transformers API bütün bunları high-level bir fonksiyon ile halledebilir.

Örneğin `tokenizer()` direkt olarak kullanıldığında, modele verilecek olan inputlar kolayca elde edilebilir.

```
from transformers import AutoTokenizer

checkpoint = "distilbert-base-uncased-finetuned-sst-2-english"
tokenizer = AutoTokenizer.from_pretrained(checkpoint)

sequence = "I've been waiting for a HuggingFace course my whole life."

model_inputs = tokenizer(sequence)
```

Burada, `model_inputs` değişkeni, model için gerekli olan her şeyi barındırır. Örneğin bu değişken, DistilBERT modeli için input ID'ler ve attention mask içerir. Aşağıda görüleceği gibi, bu metod oldukça güçlüdür. Öncelikle, tek bir cümleyi tokenlerine ayırabilir:

```
sequence = "I've been waiting for a HuggingFace course my whole life."

model_inputs = tokenizer(sequence)
```

Kod üzerinde herhangi bir değişiklik yapılmadan, mulitple sequence ile de çalışabilir:

```
sequences = ["I've been waiting for a HuggingFace course my whole life.", "So have I!"]

model_inputs = tokenizer(sequences)
```

İstenilen amaca yönelik olarak otomatik olarak padding yapabilir:

```
# Will pad the sequences up to the maximum sequence length
model_inputs = tokenizer(sequences, padding="longest")

# Will pad the sequences up to the model max length
# (512 for BERT or DistilBERT)
model_inputs = tokenizer(sequences, padding="max_length")

# Will pad the sequences up to the specified max length
model_inputs = tokenizer(sequences, padding="max_length", max_length=8)
```

Aynı zamanda cümleleri kırpabilir:

```
sequences = ["I've been waiting for a HuggingFace course my whole life.", "So have
 I!"]

# Will truncate the sequences that are longer than the model max length
# (512 for BERT or DistilBERT)
model_inputs = tokenizer(sequences, truncation=True)

# Will truncate the sequences that are longer than the specified max length
model_inputs = tokenizer(sequences, max_length=8, truncation=True)
```

`tokenizer` objesi, farklı framework'lerin (tensorflow, pytorch…) tensorleri arasında dönüşüm de gerçekleştirebilir. Aşağıda bir örnek verilmişir:

```
sequences = ["I've been waiting for a HuggingFace course my whole life.", "So have
 I!"]

# Returns PyTorch tensors
model_inputs = tokenizer(sequences, padding=True, return_tensors="pt")

# Returns TensorFlow tensors
model_inputs = tokenizer(sequences, padding=True, return_tensors="tf")

# Returns NumPy arrays
model_inputs = tokenizer(sequences, padding=True, return_tensors="np")
```

Special tokens

Bir cümlenin manuel olarak (`tokenizer.tokenize()`) elde edilen id'ler ile otomatik olarak elde edilen id'leri temsil eden liste arasında küçük farklar bulunmaktadır. Örneğin:

```
sequence = "I've been waiting for a HuggingFace course my whole life."

model_inputs = tokenizer(sequence)
```

```
print(model_inputs["input_ids"])

tokens = tokenizer.tokenize(sequence)
ids = tokenizer.convert_tokens_to_ids(tokens)
print(ids)
```

Aşağıdaki çıktıyı üretir:

```
[101, 1045, 1005, 2310, 2042, 3403, 2005, 1037, 17662, 12172, 2607, 2026, 2878, 2166,
 1012, 102]
[1045, 1005, 2310, 2042, 3403, 2005, 1037, 17662, 12172, 2607, 2026, 2878, 2166, 1012]
```

Başa ve sona bir adet token eklendiği fark edilebilir. Yukarıdaki id'ler bir decode edildiği zaman aşağıdaki cümleler elde edilir:

```
"[CLS] i've been waiting for a huggingface course my whole life. [SEP]"
"i've been waiting for a huggingface course my whole life."
```

Tokenizer, otomatik olarak bazı özel token'lar eklemiştir. Bunun nedeni modelin bu tokenler ile eğitilmiş olmasıdır. Tokenizer objesi, bu modele verilecek cümlelerin uygun forma getirilmesi görevini üstlenir. Bu özel token'lar ve pozisyonları modelden modele değişiklik gösterebilir ancak tokenizer bunu otomatik olarak ayarlayacaktır.

Wrapping up: From tokenizer to model

Tokenizer objesinin her adımını inceledik. Aşağıda karşılaşılabilecek her durum için referans kullanım verilmiştir:

```
import torch
from transformers import AutoTokenizer, AutoModelForSequenceClassification

checkpoint = "distilbert-base-uncased-finetuned-sst-2-english"
tokenizer = AutoTokenizer.from_pretrained(checkpoint)
model = AutoModelForSequenceClassification.from_pretrained(checkpoint)
sequences = ["I've been waiting for a HuggingFace course my whole life.", "So have I!"]

tokens = tokenizer(sequences, padding=True, truncation=True, return_tensors="pt")
output = model(**tokens)
```