04.05.22

OPENCAMPUS.sh

Transformers for Natural Language Processing and Beyond

INTRODUCTION TO TRANSFORMER MODELS

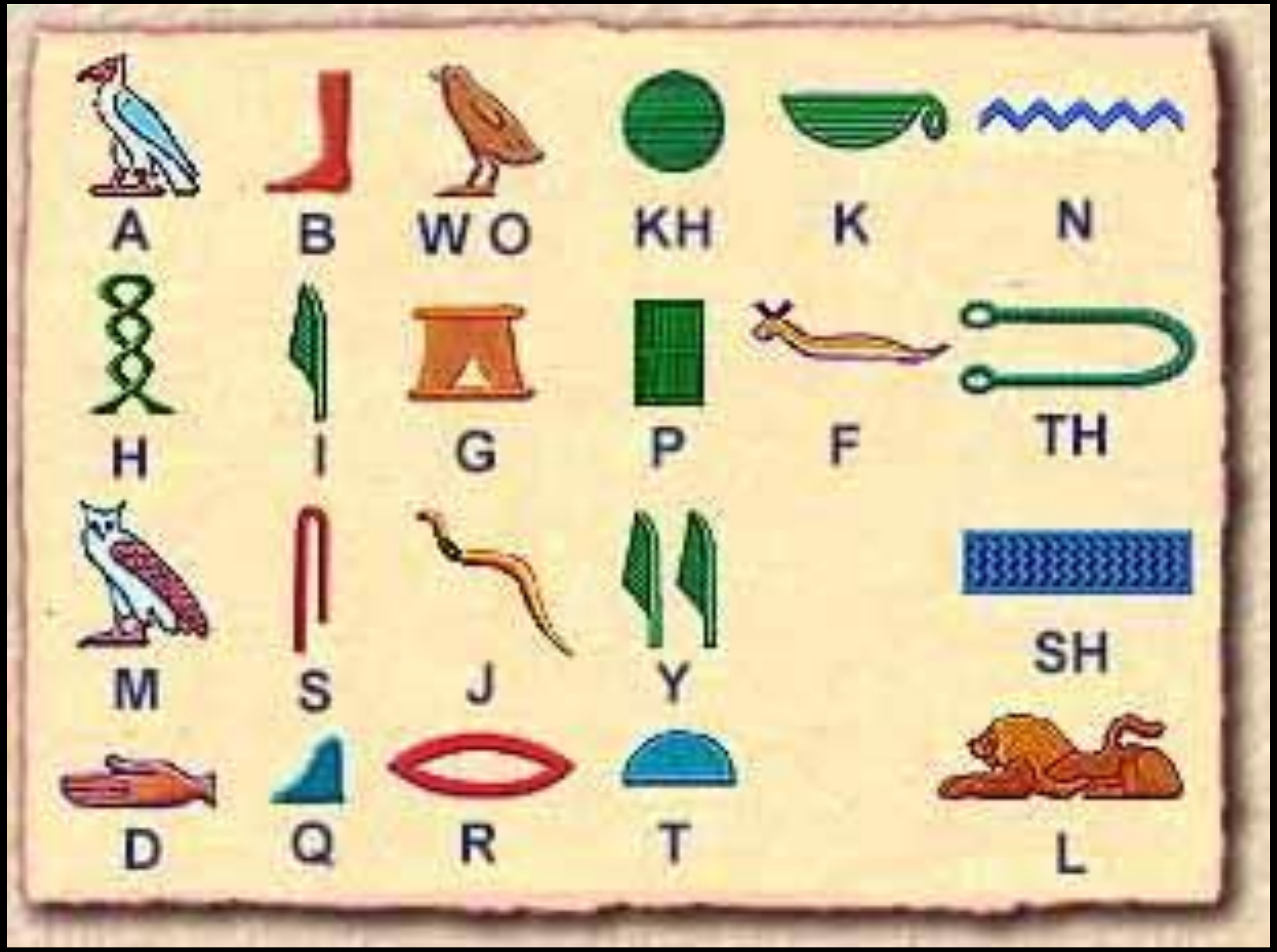
- Quiz
- Breakout Discussion
- From Pre- to Post-Processing in Transformers
- Projects

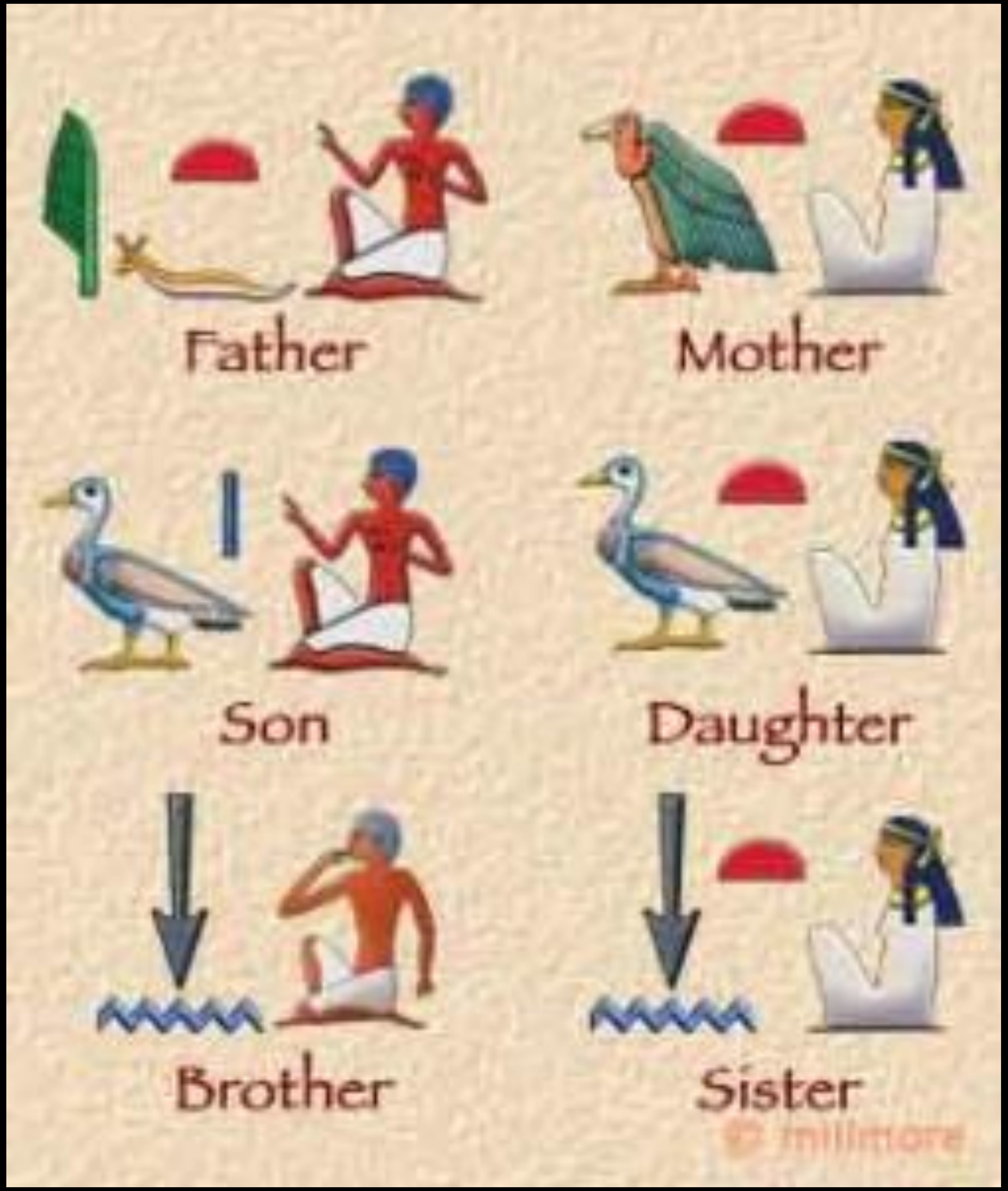
QUIZ

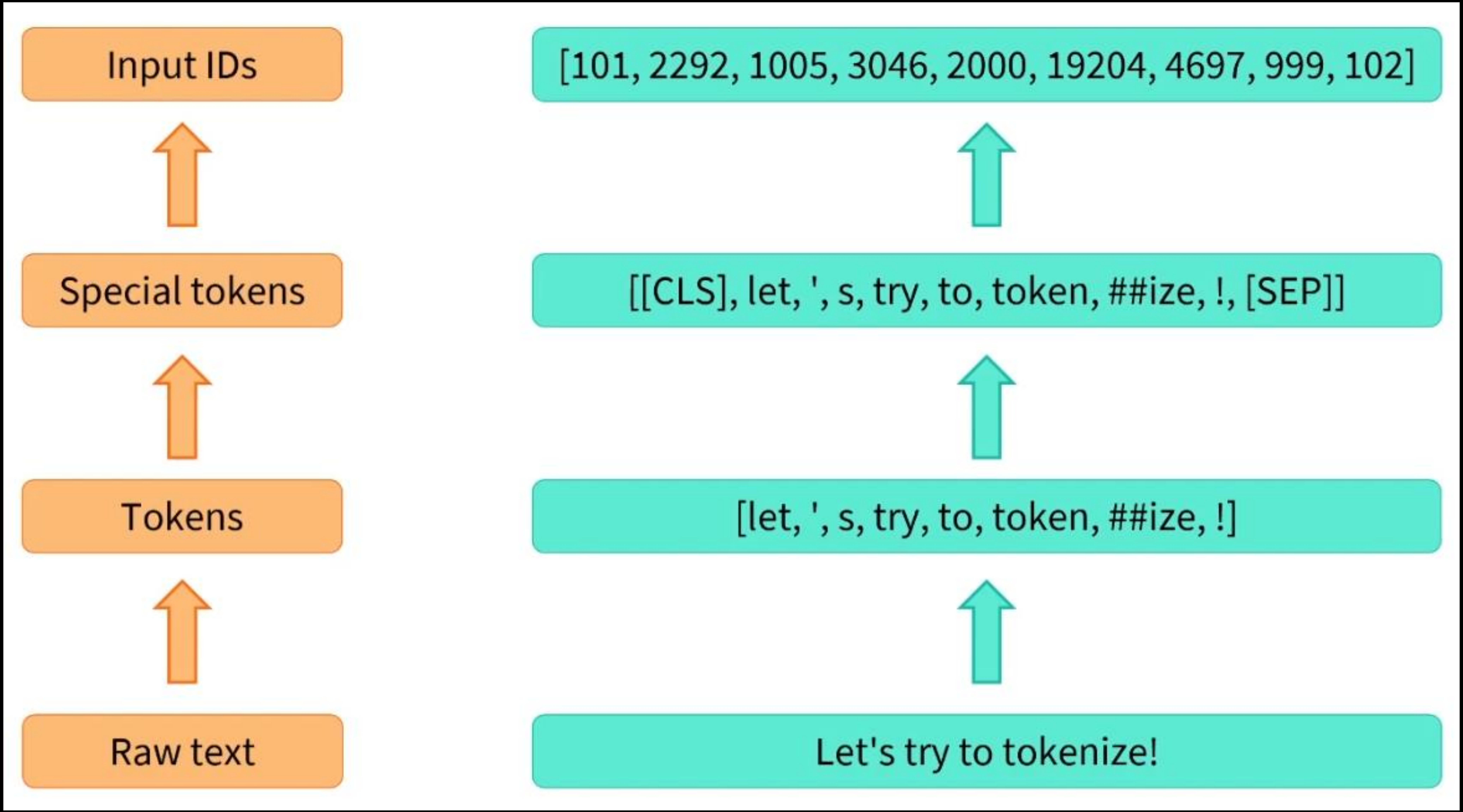
https://forms.office.com/r/KXcAaRYCHf

BREAKOUT DISCUSSION

- How does the tokenizer used by a model influence its capability?
- What might be an approach to tokenize ancient hieroglyphics?


A
B
W O
KH
K
N
H
I
G
P
F
TH
M
S
J
Y
SH
D
Q
R
T
L


Father
Mother
Son
Daughter
Brother
Sister
© millmore

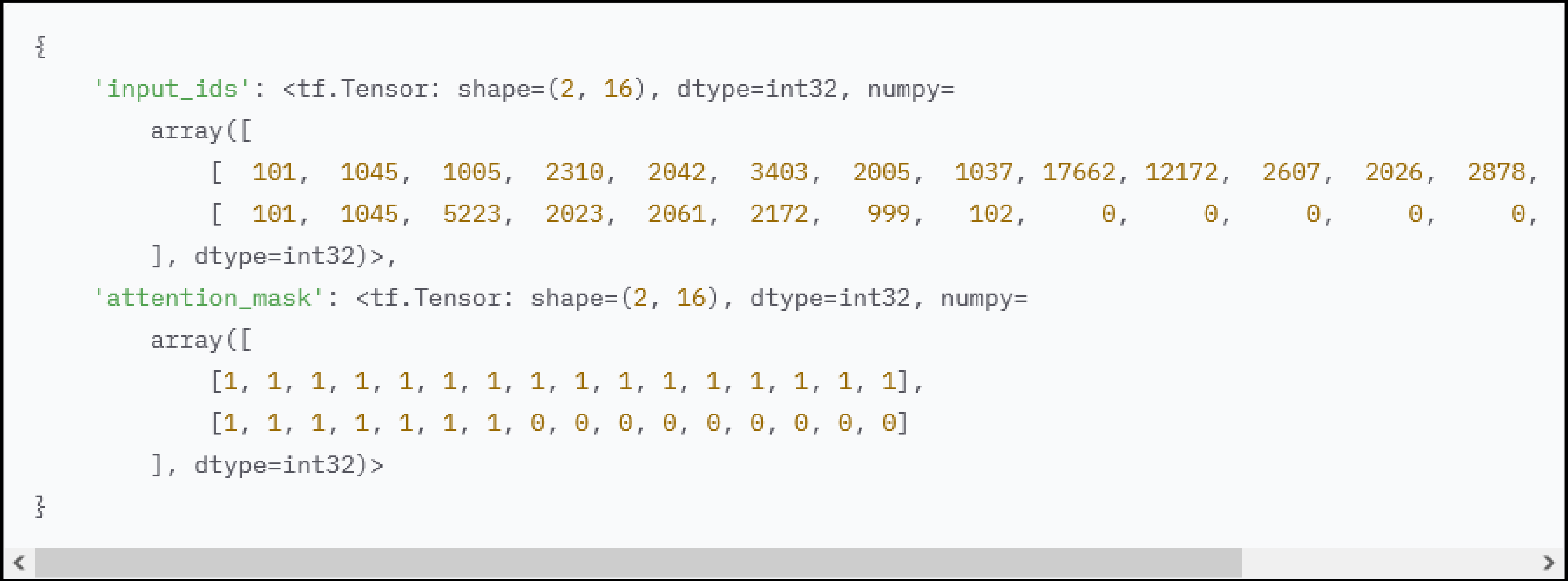

Input IDs
[101, 2292, 1005, 3046, 2000, 19204, 4697, 999, 102]
Special tokens
[[CLS], let, ', s, try, to, token, ##ize, !, [SEP]]
Tokens
[let, ', s, try, to, token, ##ize, !]
Raw text
Let's try to tokenize!

TOKENIZATION

- Byte-Pair Encoding (BPE; e.g., GPT-2)
  Chetna Khanna – Medium. (n.d.). Retrieved May 2, 2022, from https://chetnakhanna.medium.com/
- Unigram (e.g., T5, XLNet)
- WordPiece (e.g., BERT)

Tokenizers: How machines read. (2020, January 28). FloydHub Blog. https://blog.floydhub.com/tokenization-nlp/

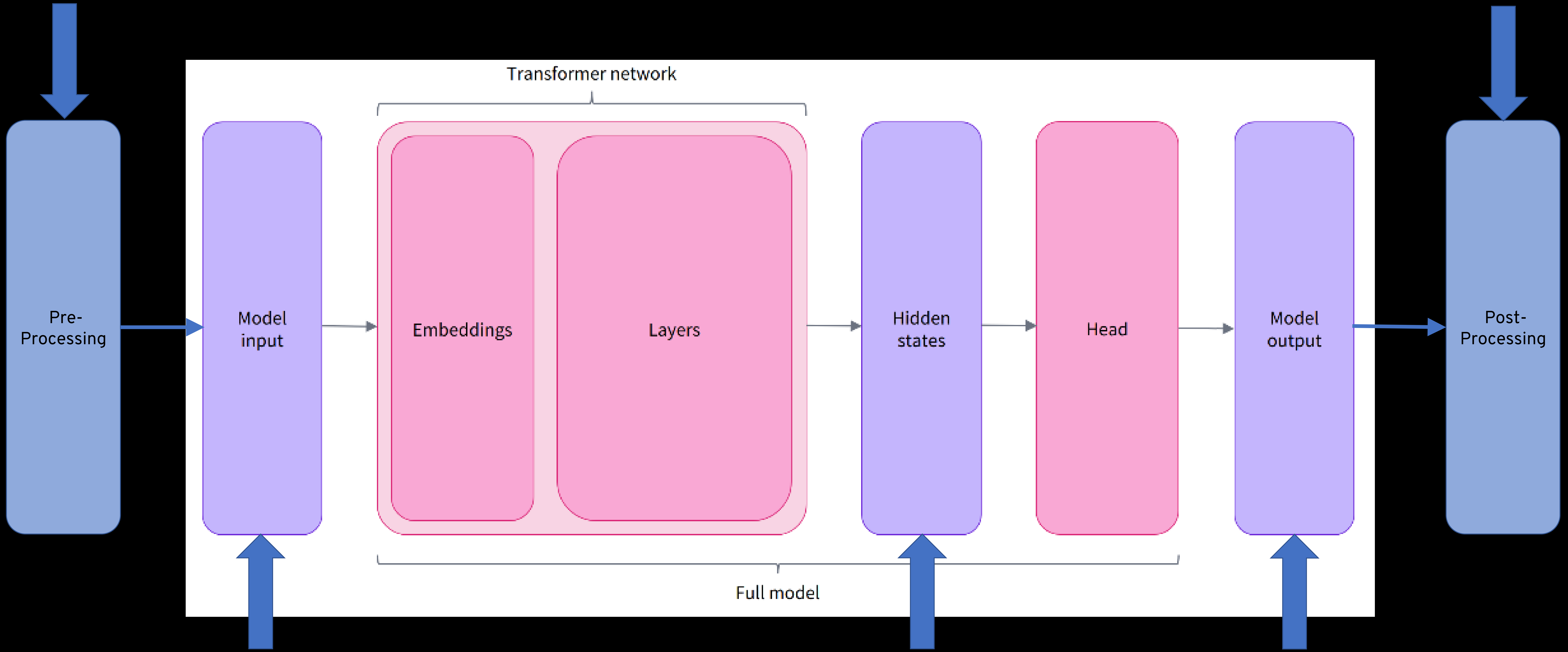
ATTENTION MASK



```
{
    'input_ids': <tf.Tensor: shape=(2, 16), dtype=int32, numpy=
        array([
            [  101,  1045,  1005,  2310,  2042,  3403,  2005,  1037, 17662, 12172,  2607,  2026,  2878,
            [  101,  1045,  5223,  2023,  2061,  2172,   999,   102,     0,     0,     0,     0,     0,
        ], dtype=int32)>,
    'attention_mask': <tf.Tensor: shape=(2, 16), dtype=int32, numpy=
        array([
            [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
            [1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        ], dtype=int32)>
}
```

- Splitting
- Mapping to integers
- Adding model dependent tokens/integers

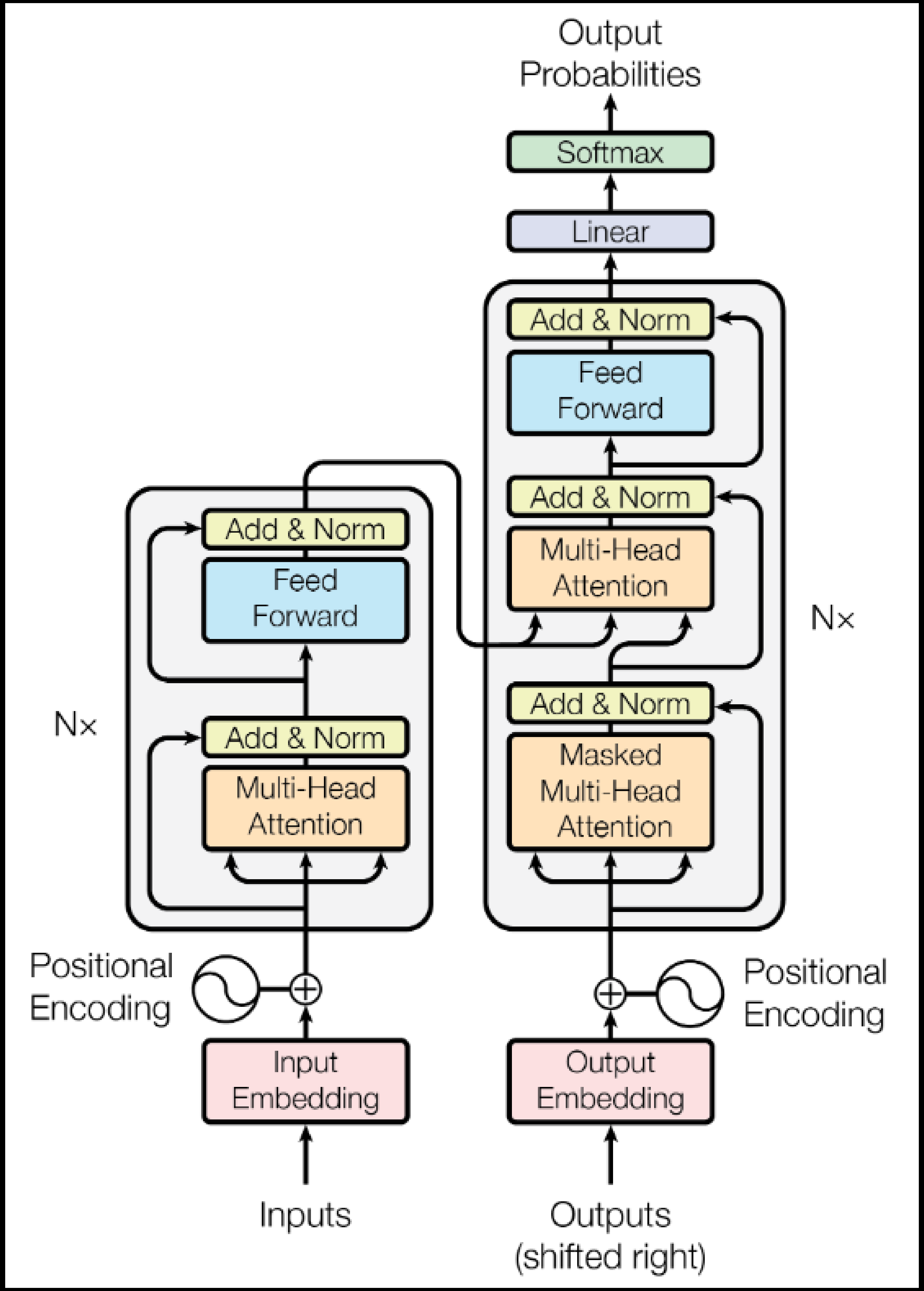
- Logits to probs
- Probs to classes
- (Classes to tokens/text)



Tokenized subwords (integers)

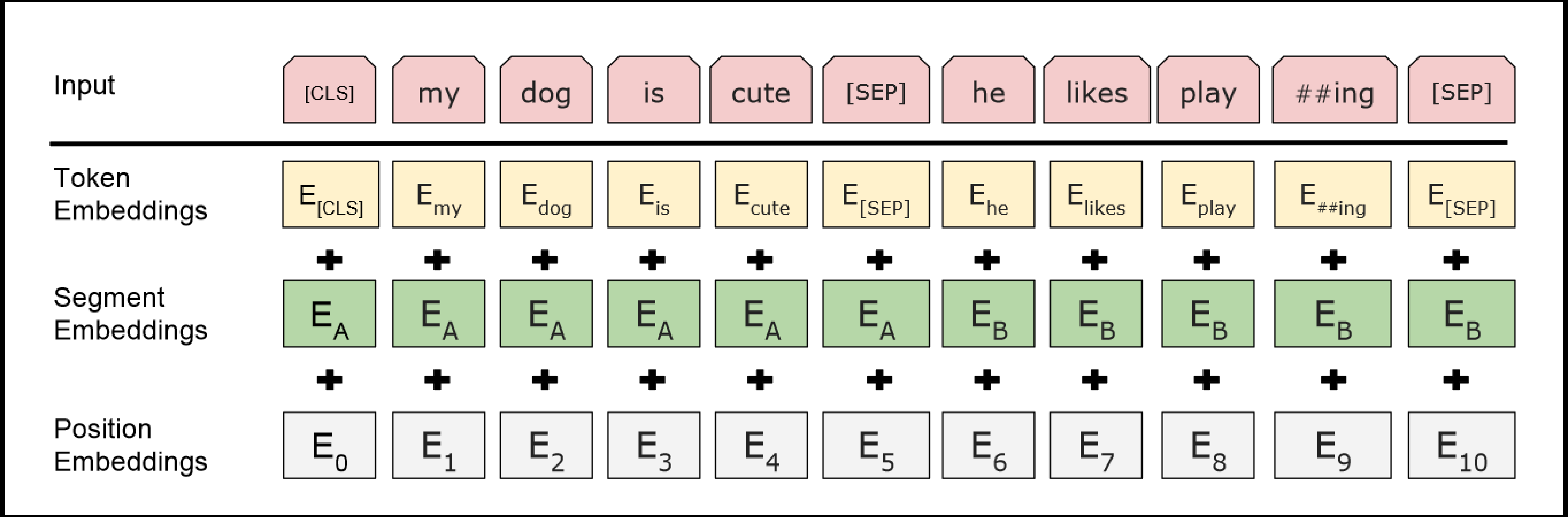
Contextualized embeddings for each token (high dimensional vectors of floats)

Logits (one float for each trained class)



Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention Is All You Need. *ArXiv:1706.03762 [Cs]*. Retrieved from http://arxiv.org/abs/1706.03762

BERT EMBEDDINGS

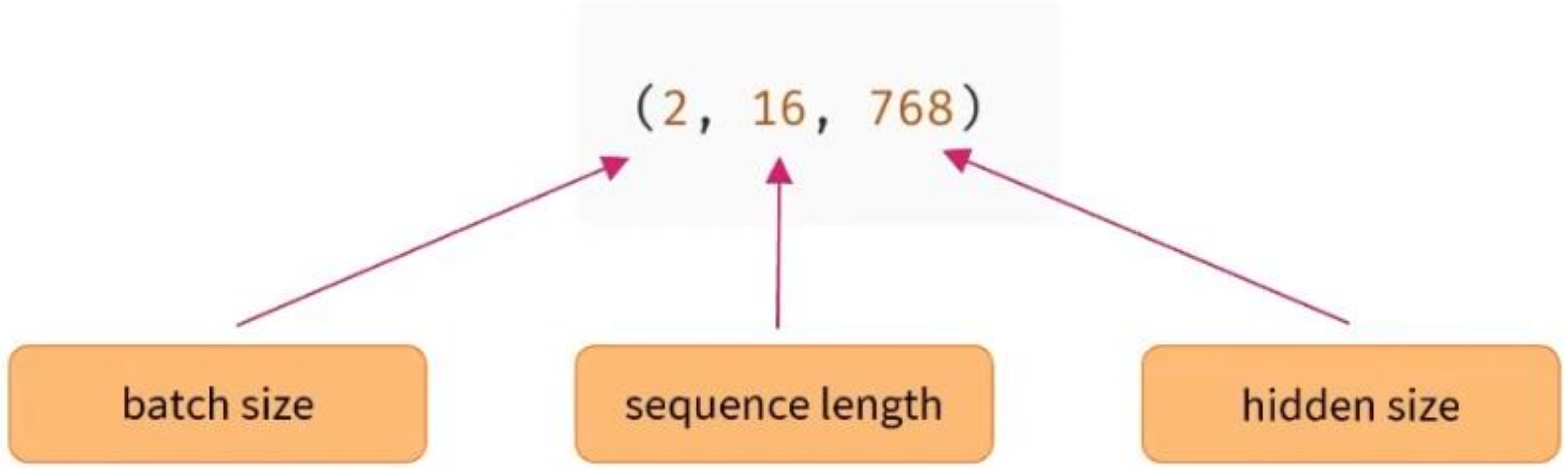


Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv:1810.04805 [Cs]*. Retrieved from http://arxiv.org/abs/1810.04805

```python
from transformers import TFAutoModel

checkpoint = "distilbert-base-uncased-finetuned-sst-2-english"
model = TFAutoModel.from_pretrained(checkpoint)
outputs = model(inputs)
outputs.last_hidden_state.shape
```



```
(2, 16, 768)
```

PROJECTS

- Chris/Dariush: Classification of student emails to predict if an argument is correctly included
- Friedrich/Nicolas/Dustin/Wang: Detection of Transposable Elements in Genome Sequences
- Prosper/Julien: Generating Marketing Content for NFTs
- Dieter/Desmond: Time Series Prediction for Electric Motors

PROJECT MILESTONES

- 11.05. Form project groups
- 18.05. Literature review
- 25.05. Dataset characteristics
- 01.06. Baseline model
- 08.06. Model & model evaluation (Joint Coding)
- 15.06. Project presentations

TODOS UNTIL NEXT WEEK

- Complete chapter 3 (Fine-Tuning a Pretrained Model) of the Hugging Face course
- Next week everyone should have a clear idea for a project