BERT
(Bidirectional Transformers for Language Understanding)

BERT trains language model in both directions

Word prediction using context from only one side

Left-to-right prediction	Right-to-left prediction
Alaska	York
Alaska is	New York
Alaska is about	than New York
Alaska is about twelve	larger than New York
Alaska is about twelve times	times larger than New York
Alaska is about twelve times larger	twelve times larger than New York
Alaska is about twelve times larger than	about twelve times larger than New York
Alaska is about twelve times larger than New	is about twelve times larger than New York
Alaska is about twelve times larger than New York	Alaska is about twelve times larger than New York

Word prediction using context from both sides (e.g. BERT)

Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

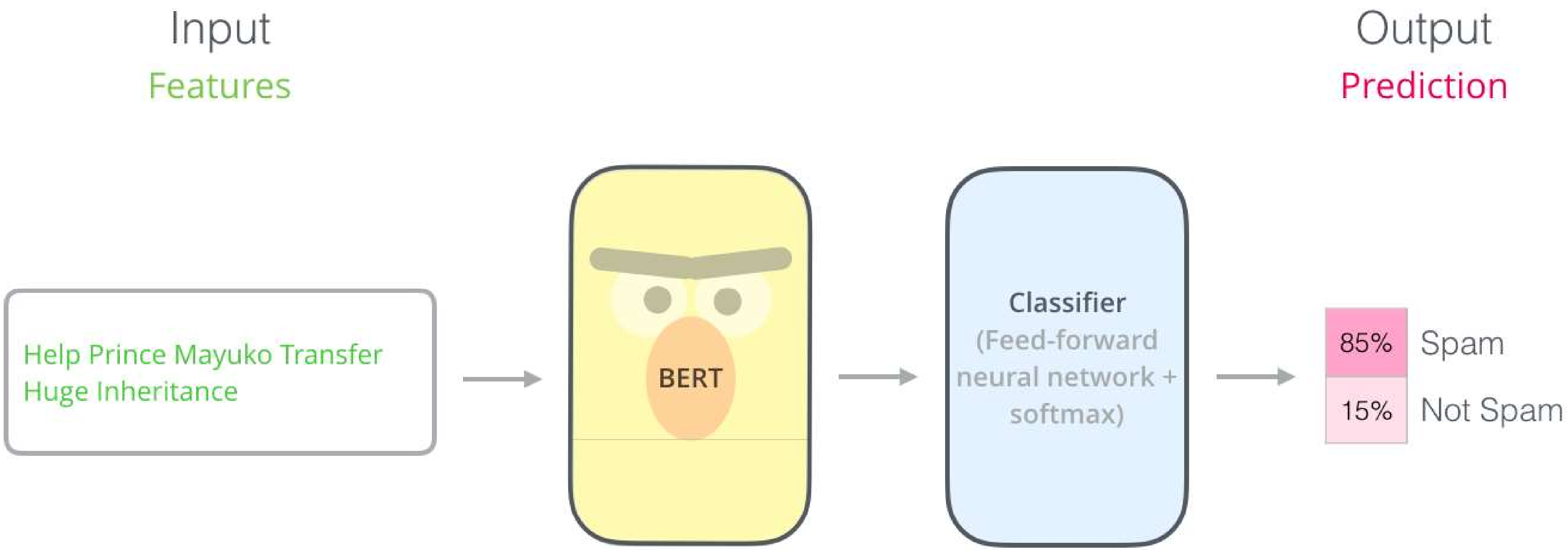
Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

BERT trains Language Model in both directions

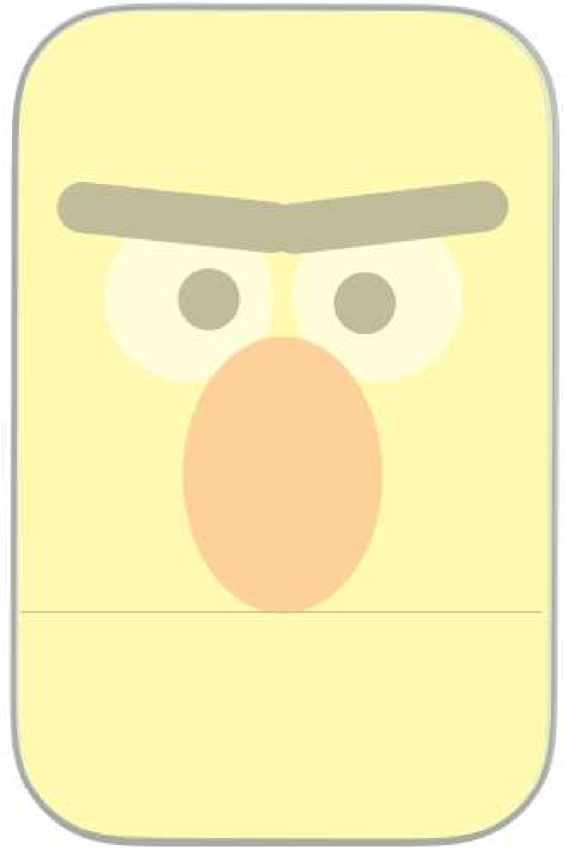
BERT: Sentence Classification



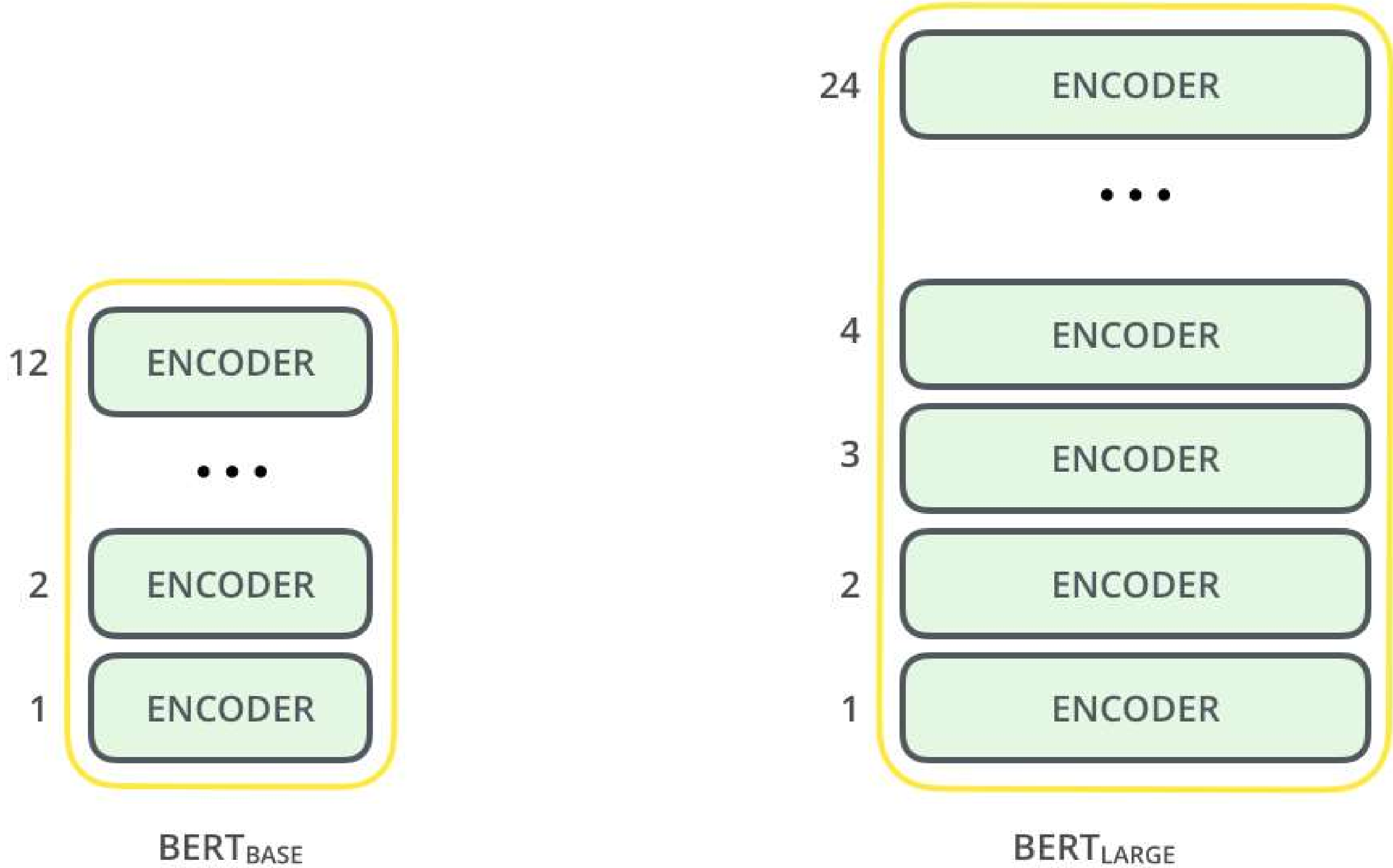
BERT: Sentence Classification

Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached…	Not Spam

BERT: Sentence Classification

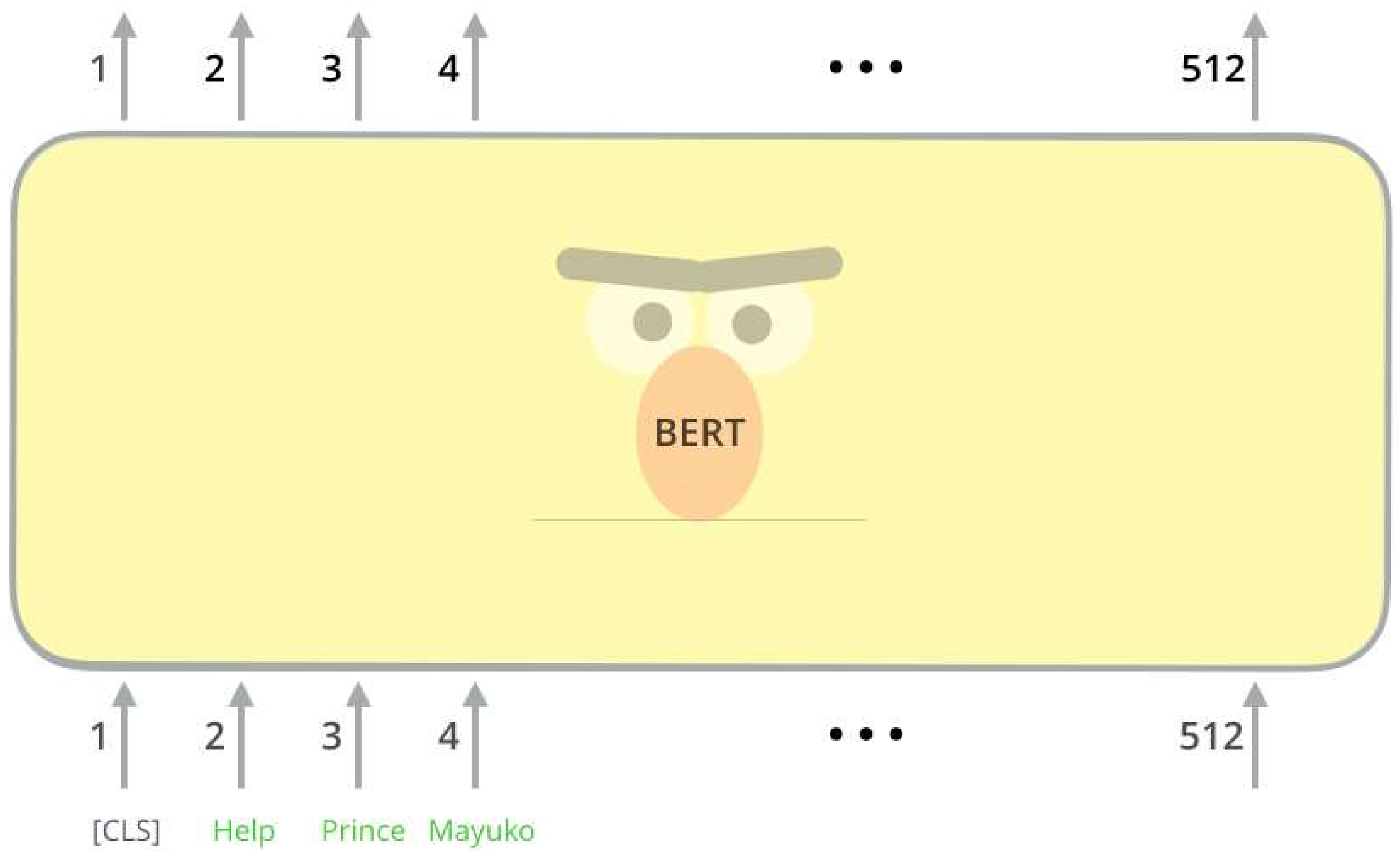


BERT: Only Encoders are used

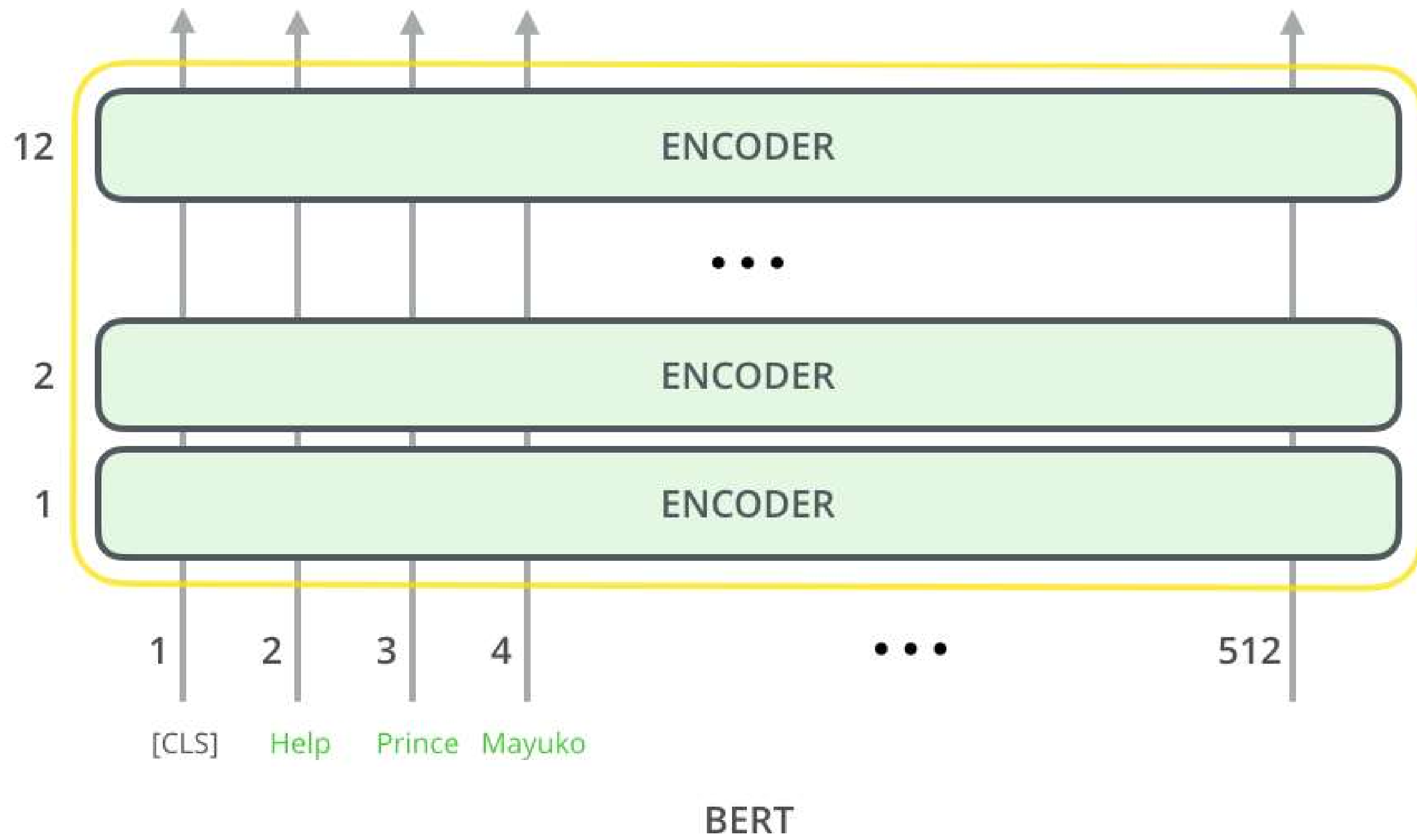


These also have larger feedforward-networks (768 and 1024 hidden units respectively), and more attention heads (12 and 16 respectively)

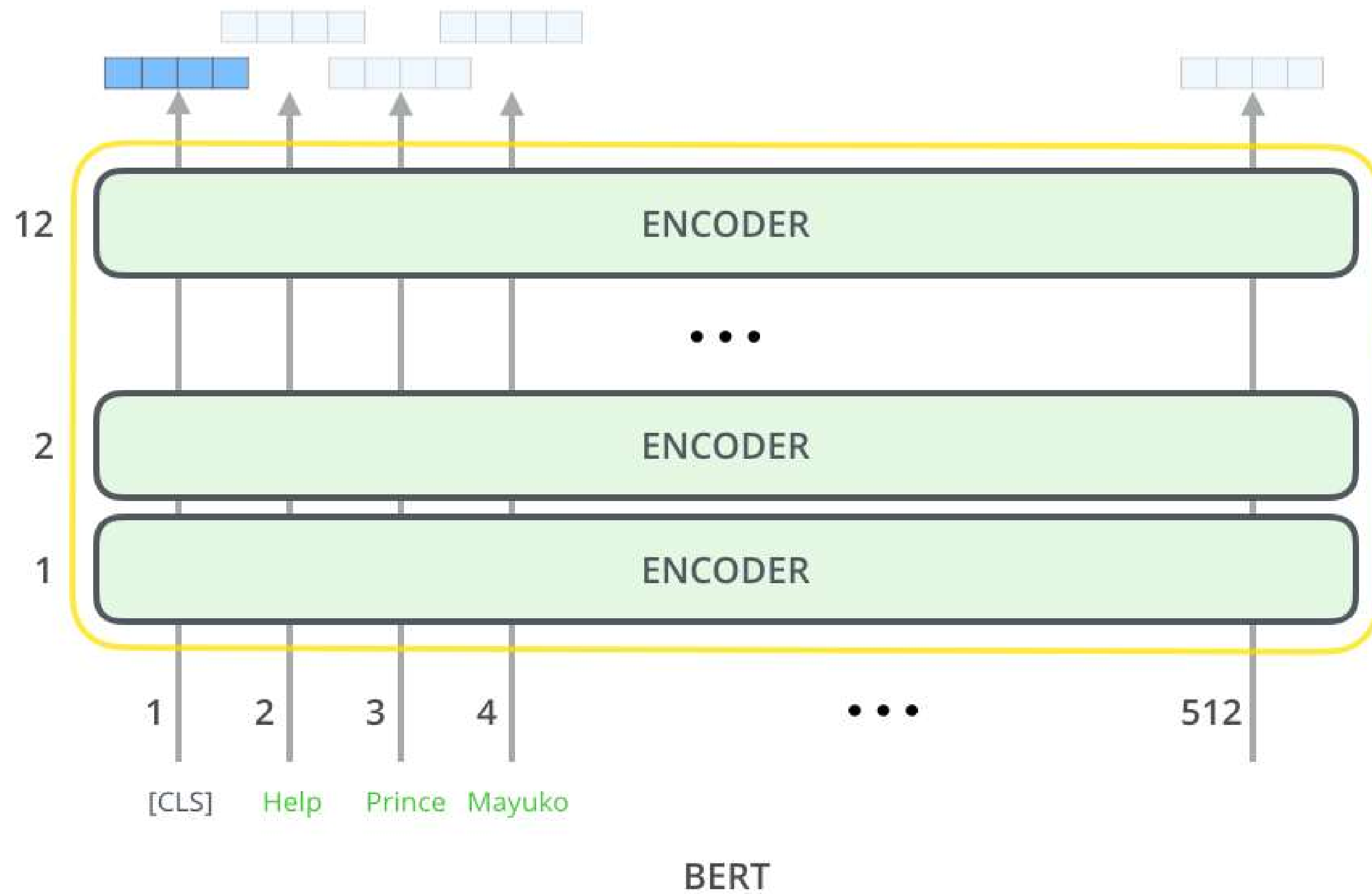
BERT: A special [CLS] token is used for classification



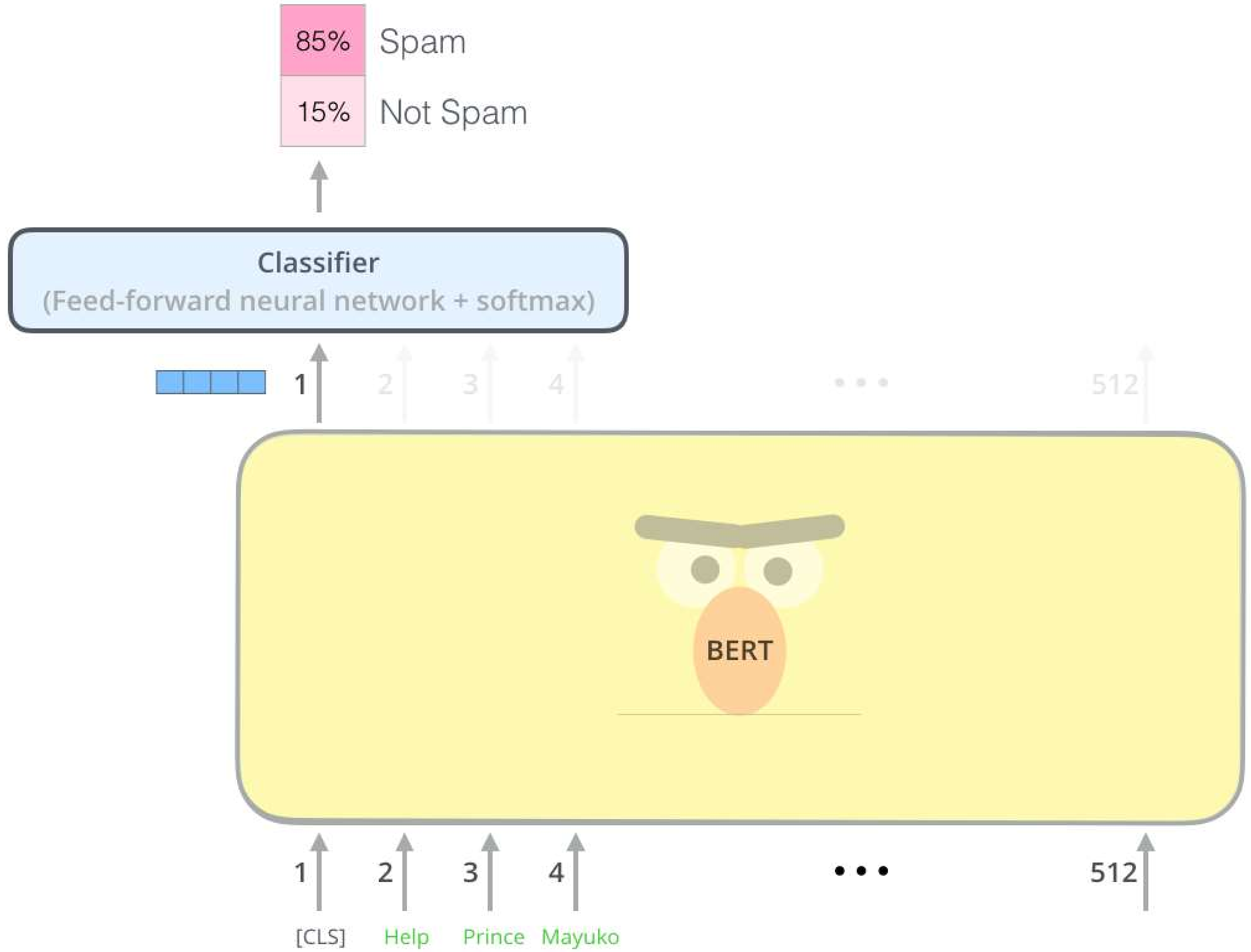
BERT: Only Encoders are used



BERT: Only Encoders are used



BERT: A special [CLS] token is used for classification



BERT: For Language Modeling

- **Use [MASK] token for 15% for text**
- In this 15% tokens, 80% will be replaced with '[MASK]', 10% will be replaced with a random word from vocabulary, 10% will not be replaced.

All 15% are not MASKED

- 80% were replaced by the ‘<MASK>’ token

Example: “My dog is ***<MASK>****”*

- 10% were replaced by a random token

Example: “My dog is **apple***”*

- 10% were left intact

Example: “My dog is **hairy***”*

Why did they not use a ‘<MASK>’ replacement token all around?

- If the model had been trained on only predicting ‘<MASK>’ tokens and then never saw this token during fine-tuning, it would have thought that there was no need to predict anything and this would have hampered performance
- By sometimes asking it to predict a word in a position that did not have a ‘<MASK>’ token, the model needed to learn a contextual representation of *all* the words in the input sentence, just in case it was asked to predict them afterwards.

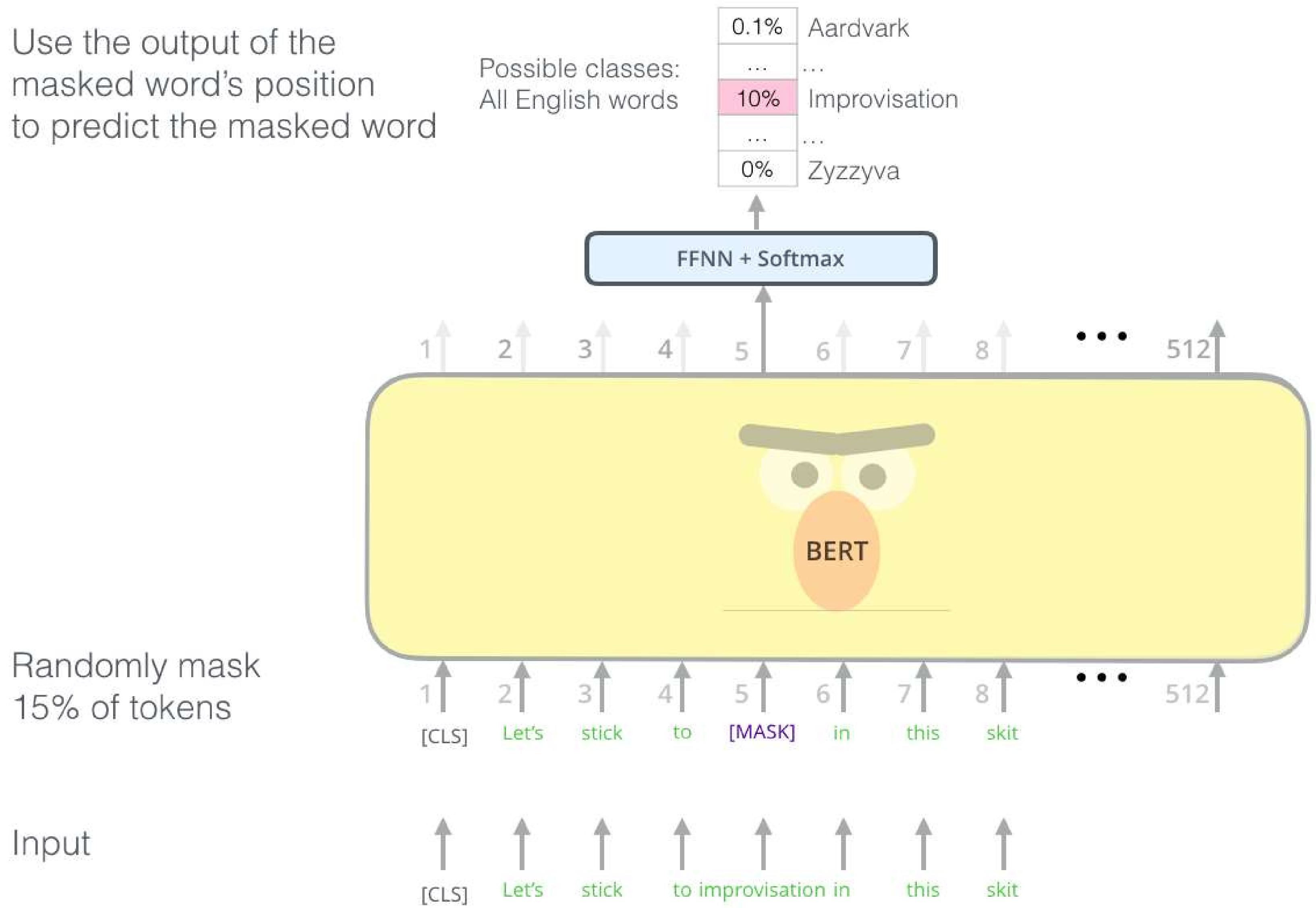
- *Are not random tokens enough? Why did they leave some sentences intact?*

Will random tokens confuse the model?

- *The model will only predict 15% of the tokens but language models predict 100% of tokens, does this mean that the model needs more iterations to achieve the same loss?*

- Training will be slower as model only predicts 15% of words

language modeling: Use [MASK] token for 15% for text



BERT: Two-sentence Tasks

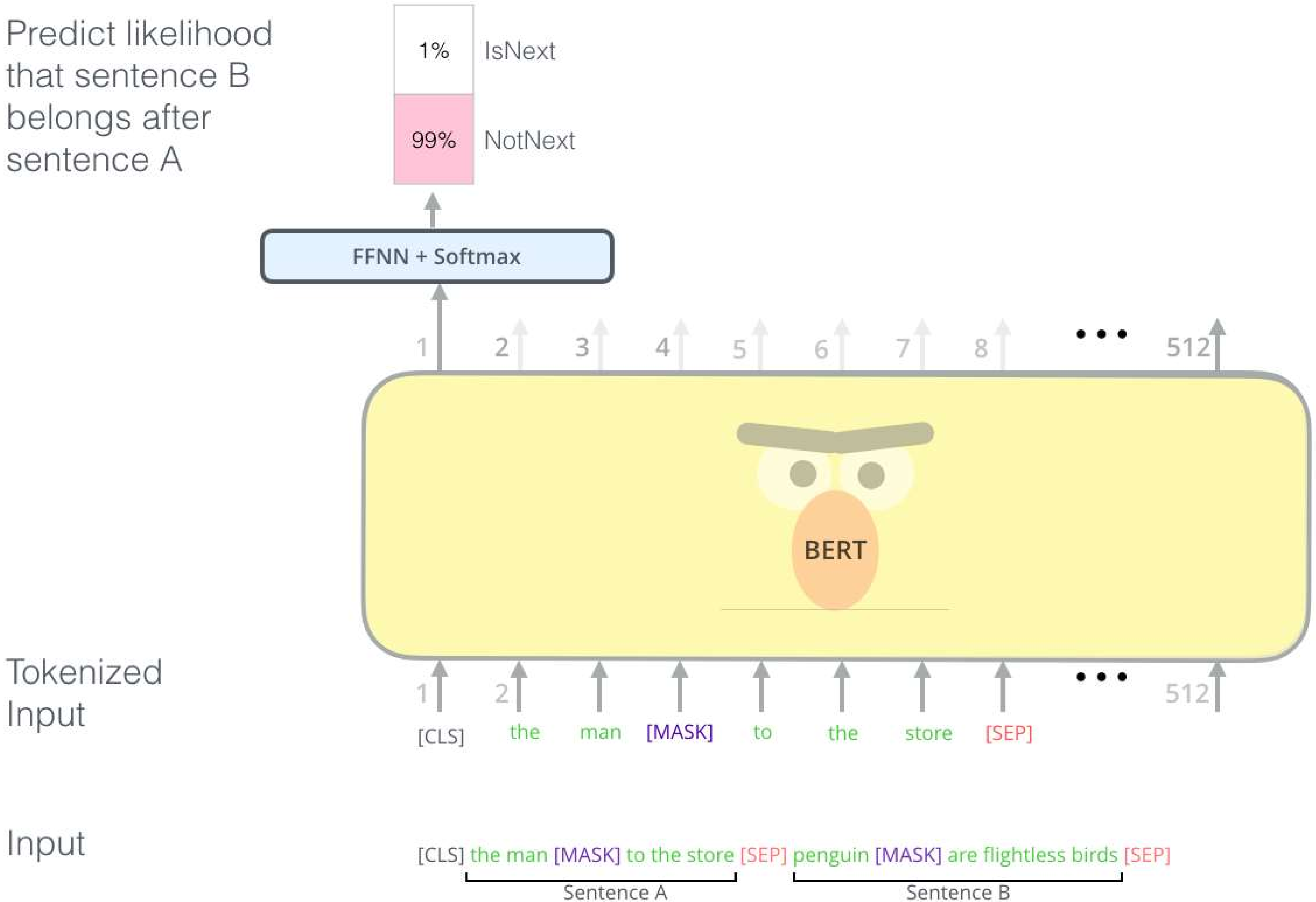
- The sentence pair contains 2 sentences, 50% of the sentence pairs are related sentences which appears in the document one by one, 50% of the sentence pairs are not related, which the sentence are combined randomly.
- Useful for many NLP tasks like QA, entailment, inference etc

BERT: Two-sentence Tasks

- *Input = [CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]*
- In this case the sentences are adjacent, so the label in [CLS] would be ‘<IsNext>’ as in:
- *Input = <IsNext> the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]*

BERT: Two-sentence Tasks

- The loss was calculated as the sum of the mean masked LM likelihood and the mean next sentence prediction likelihood.


Predict likelihood that sentence B belongs after sentence A
1% IsNext
99% NotNext
FFNN + Softmax
1 2 3 4 5 6 7 8 ... 512
BERT
Tokenized Input
1 2 ... 512
[CLS] the man [MASK] to the store [SEP]
Input
[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flightless birds [SEP]
Sentence A
Sentence B

Out of vocabulary words

- BERT Tokenization: WordPiece model

BERT Tokenization: WordPiece model

- This model greedily creates a fixed-size vocabulary of individual characters, subwords, and words that best fits our language data. Since the vocabulary limit size of BERT tokenizer model is 30,000, the WordPiece model generated a vocabulary that contains all English characters plus the ~30,000 most common words and subwords found in the English

BERT Tokenization: WordPiece model

- Whole words
- Subwords occuring at the front of a word or in isolation.
- Subwords not at the front of a word, which are preceded by '##' to denote this case
- Individual characters
- The word "embeddings" is represented:

  ['em', '##bed', '##ding', '##s']

BERT Tokenization: WordPiece model

- Rather than assigning out of vocabulary words to a catch-all token like ‘OOV’ or ‘UNK,’ words that are not in the vocabulary are decomposed into subword and character tokens that we can then generate embeddings for.

Segmentation Embedding

- E.g. token sentence:
- ‘[CLS] I have a dream [SEP] The cat is white [SEP]’,
- segmentation embedding: [0,0,0,0,0,1,1,1,1,1]

Mask word embedding

- 0 represents normal word, 1 represents mask word.

Context Based Similarity

- Similarity of ‘bank’ as in ‘bank robber’ to ‘bank’ as in ‘river bank’
  - Cosine similarity = 0.67

- Similarity of ‘bank’ as in ‘bank robber’ to ‘bank’ as in ‘bank vault’
  - Cosine similarity = 0.9

How to use pretrained model

- Fine Tune

- Feature based