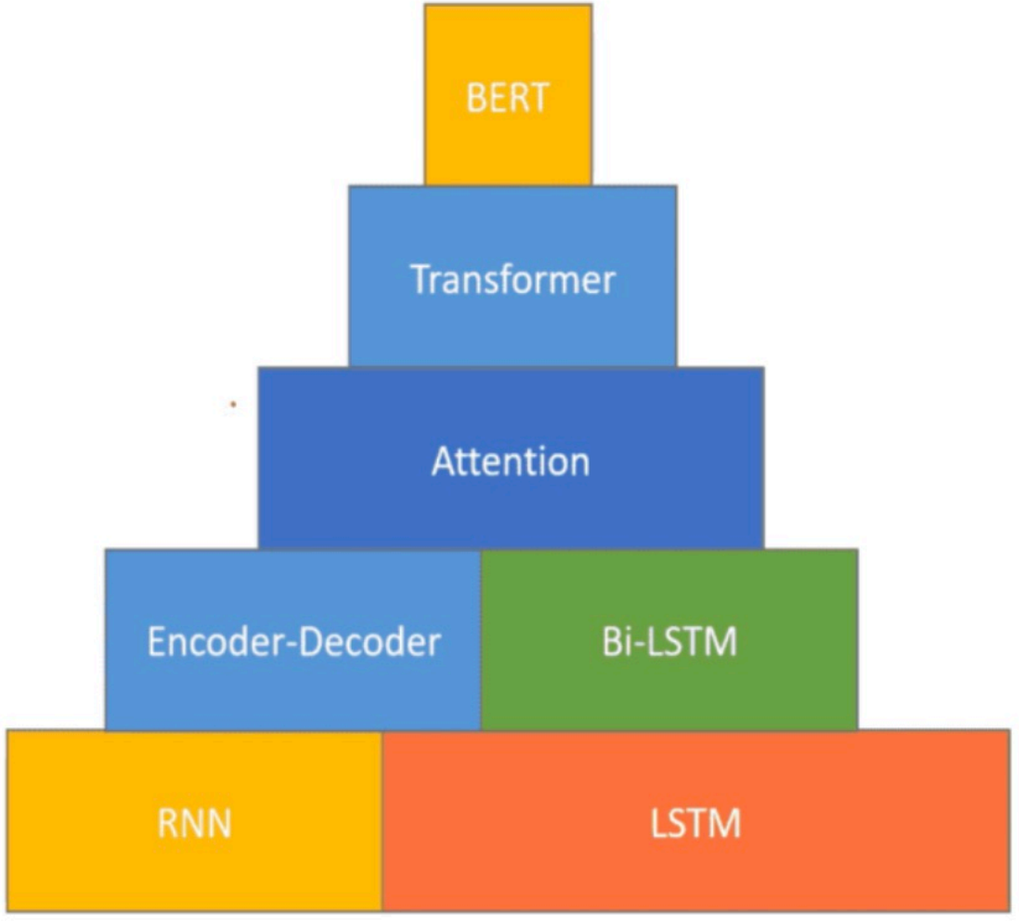
Natural Language Processing

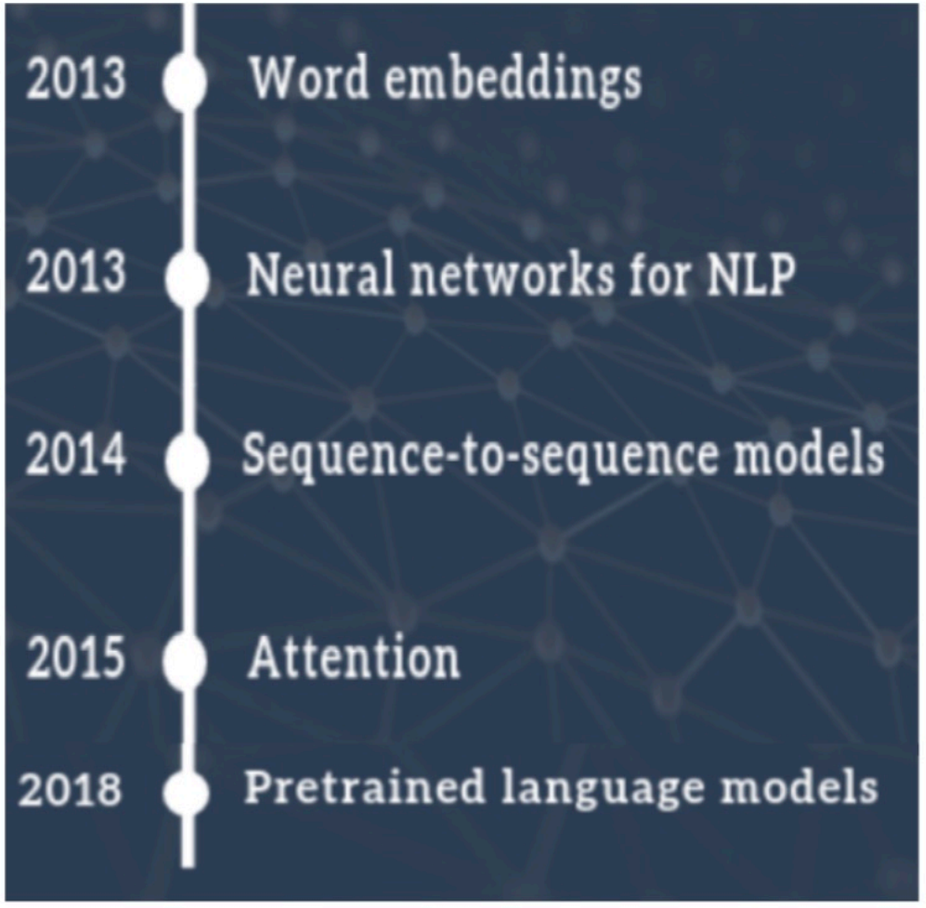
Session-6

- Transformers
- Attention
- BERT
- Working Logic of BERT

Bidirectional Encoder Representations From Transformers (BERT)

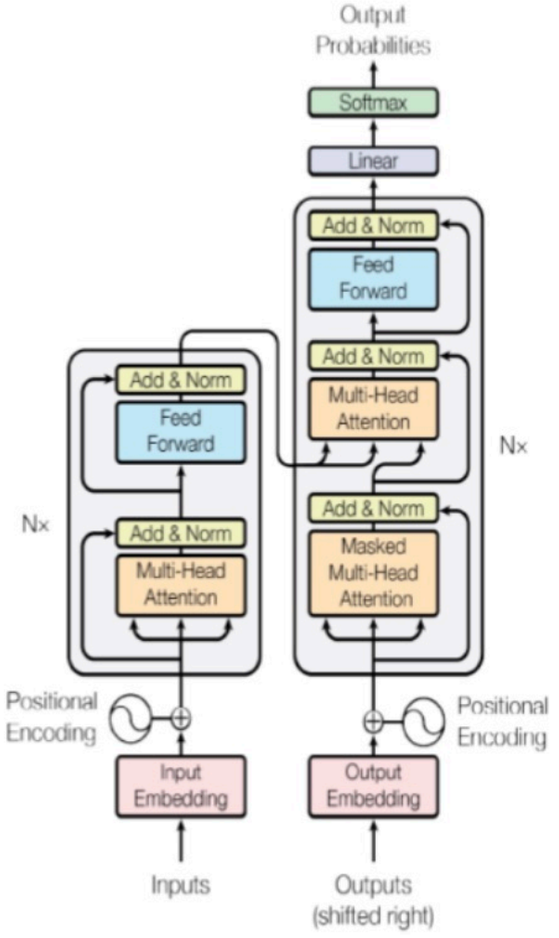
Introduction of BERT architecture





Bidirectional Encoder Representations From Transformers (BERT)

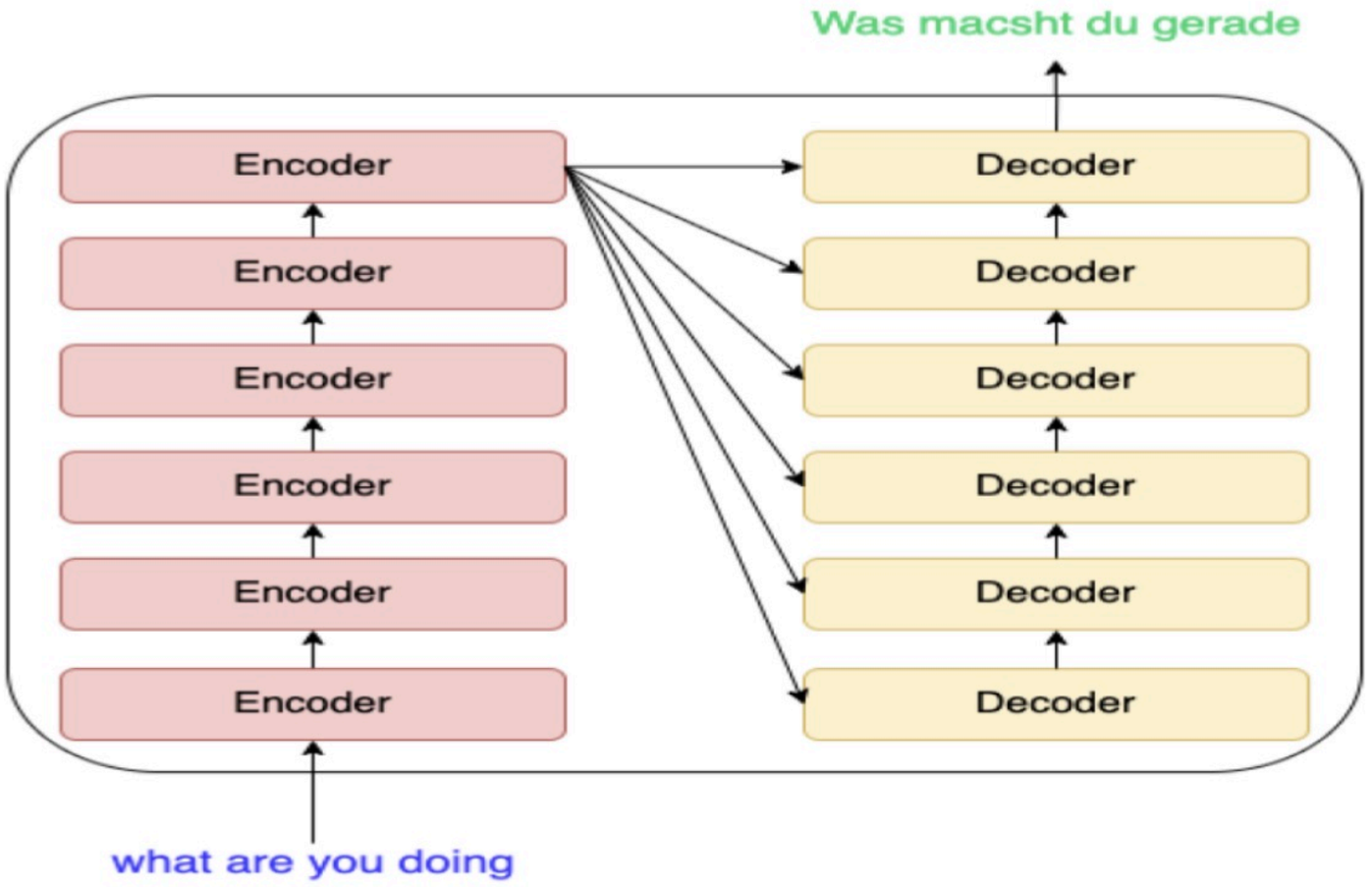
Transformer



- The transformer is built on **encoder-decoder** and **attention** mechanisms and consists of **6 encoder-decoders** working in **parallel** with each other.

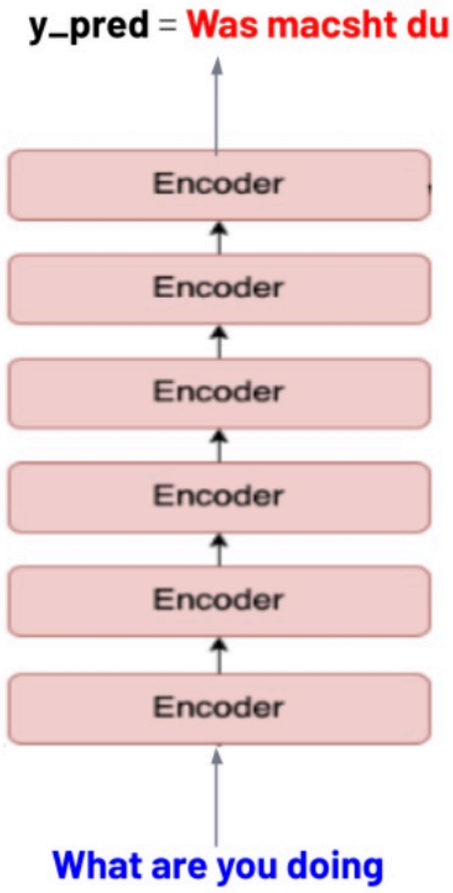
Bidirectional Encoder Representations From Transformers (BERT)

Transformer



Bidirectional Encoder Representations From Transformers (BERT)

BERT Transformer



y_test = Was macsht du gerade

However, Only **encoders** are used in BERT models, **decoders** are **not used**.

Bidirectional Encoder Representations From Transformers (BERT)

Attention

-The **attention** mechanism (called **self-attention**) is the mechanism that decides which tokens to focus on by detecting all the contextual relationships among all tokens on which the model is trained, by means of the **query**, **key**, and **values** vectors created by the model.

Bidirectional Encoder Representations From Transformers (BERT)

Attention

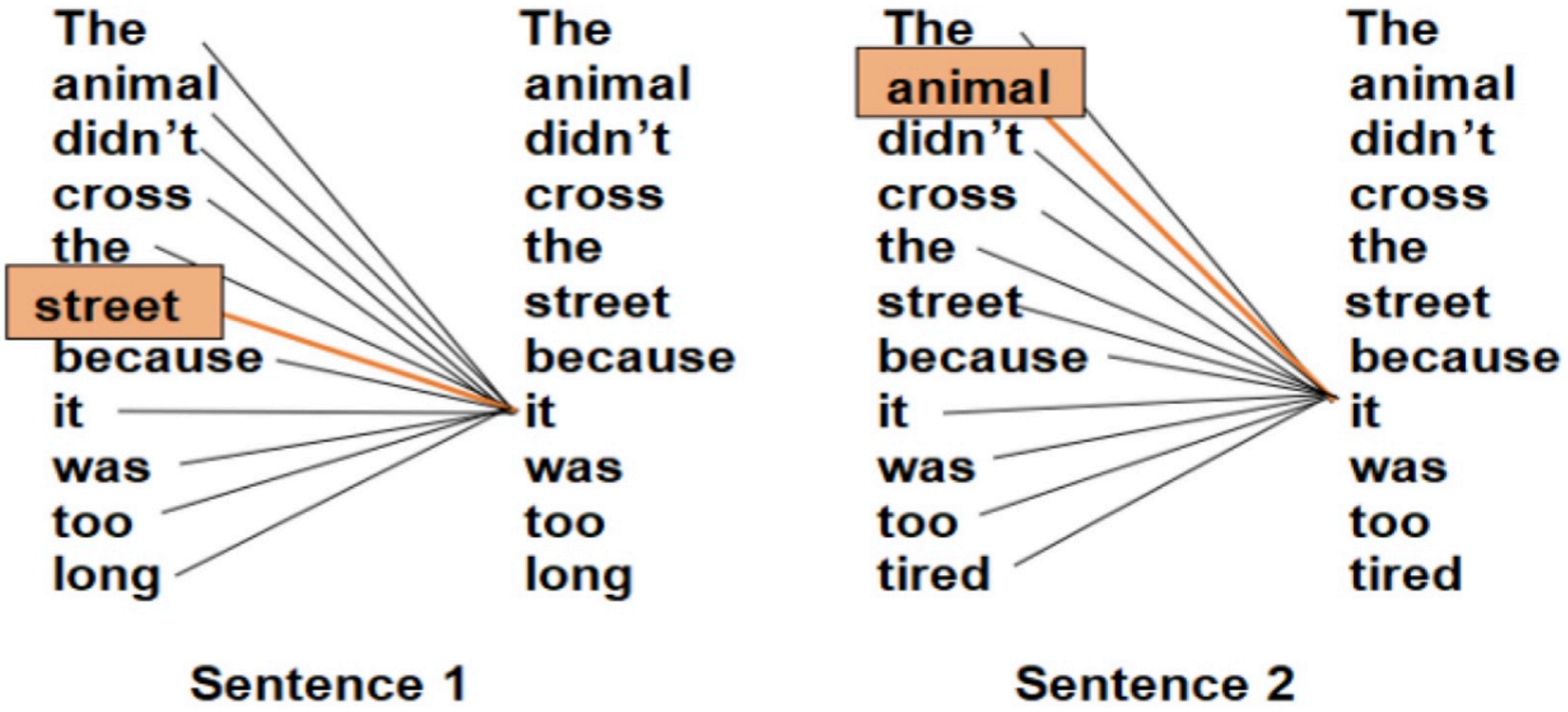
- **Query** refers to any word.

- Key refers to the words that are most contextually related to this word (query)

- Value refers to the most related word selected by the model among the related words

Bidirectional Encoder Representations From Transformers (BERT)

Attention



Bidirectional Encoder Representations From Transformers (BERT)

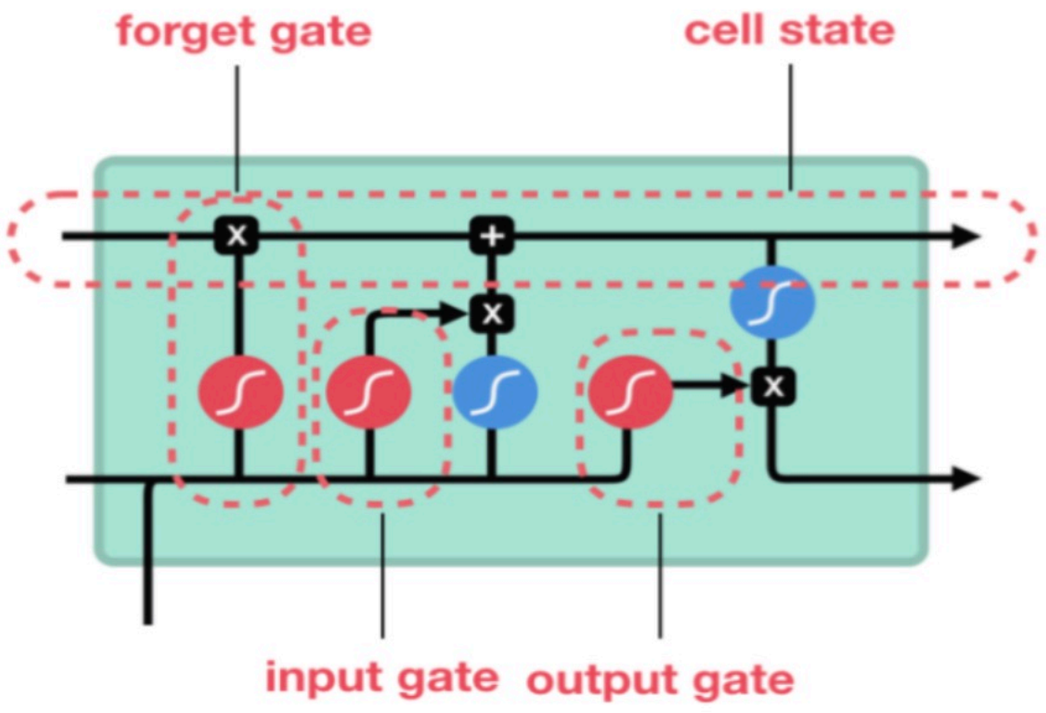
Multi-Head Attention

-The **Multi-Head attention** mechanism tries to better capture the semantic bonds between words by doing self-attention process **many times in different ways**.

- The train cost will be reduced as the model concentrates on weighted important words instead of all words.

Bidirectional Encoder Representations From Transformers (BERT)

Working Logic of Transformer



- **LSTM** and **GRU** are also insufficient in **very long sentences**

- In order to **prevent this forgetting**, the transformer models' have been used

Bidirectional Encoder Representations From Transformers (BERT)

Working Logic of Transformer

Bidirectional Encoder Representations From Transformers (BERT)

BERT

- BERT was introduced by **Google** in **2018**.

- The BERT architecture builds on top of the Transformer. We currently have two variants available:

- **BERT Base**: 12 layers (transformer blocks), 12 attention heads, and 110 million parameters.
- **BERT Large**: 24 layers (transformer blocks), 16 attention heads, and 340 million parameters.

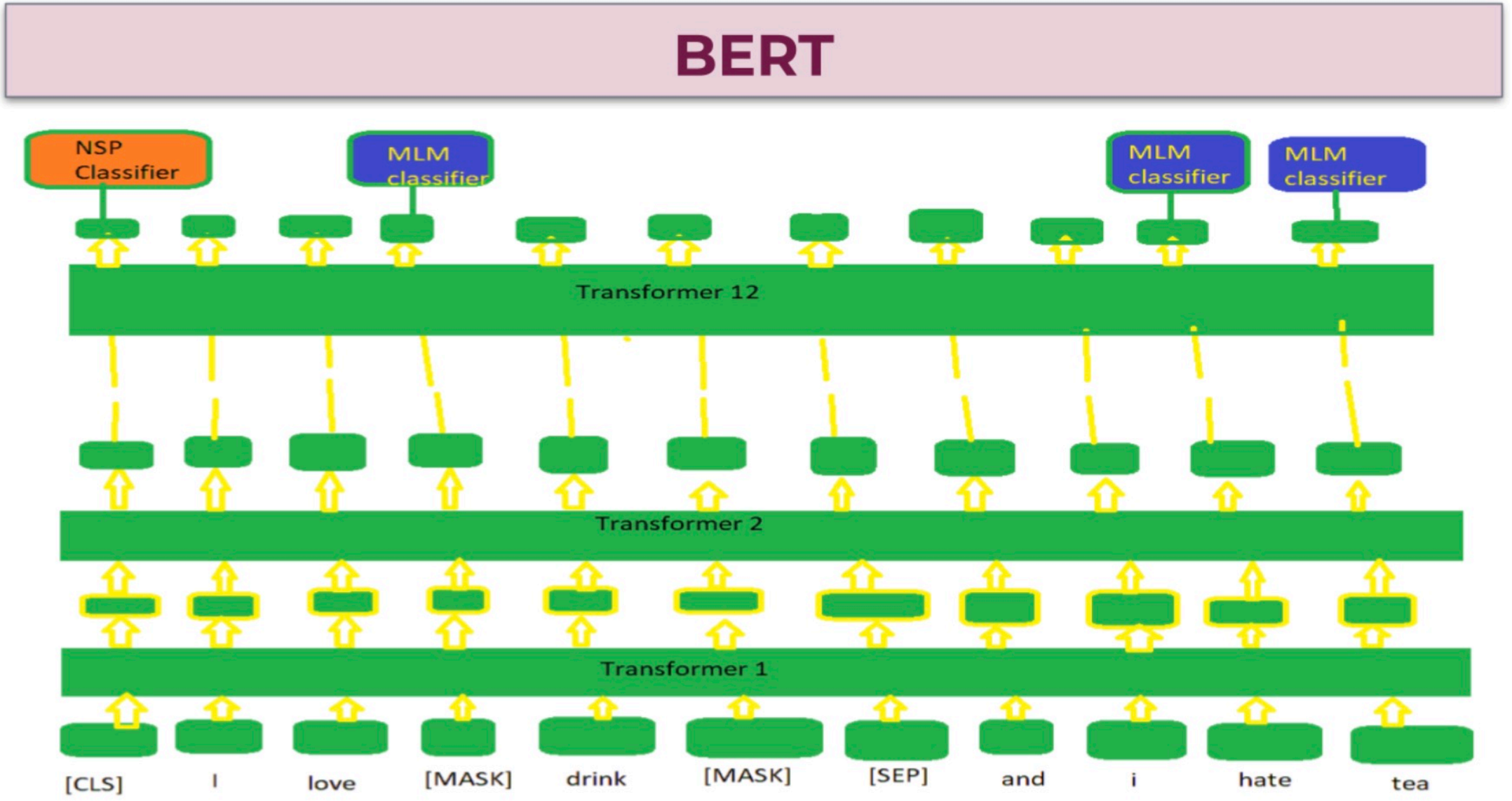
Bidirectional Encoder Representations From Transformers (BERT)

BERT

- BERT was trained in 4 days on very powerful machines on the BookCorpus, which has 800 million vocabularies, and Wikipedia, which has 2.5 billion vocabularies.

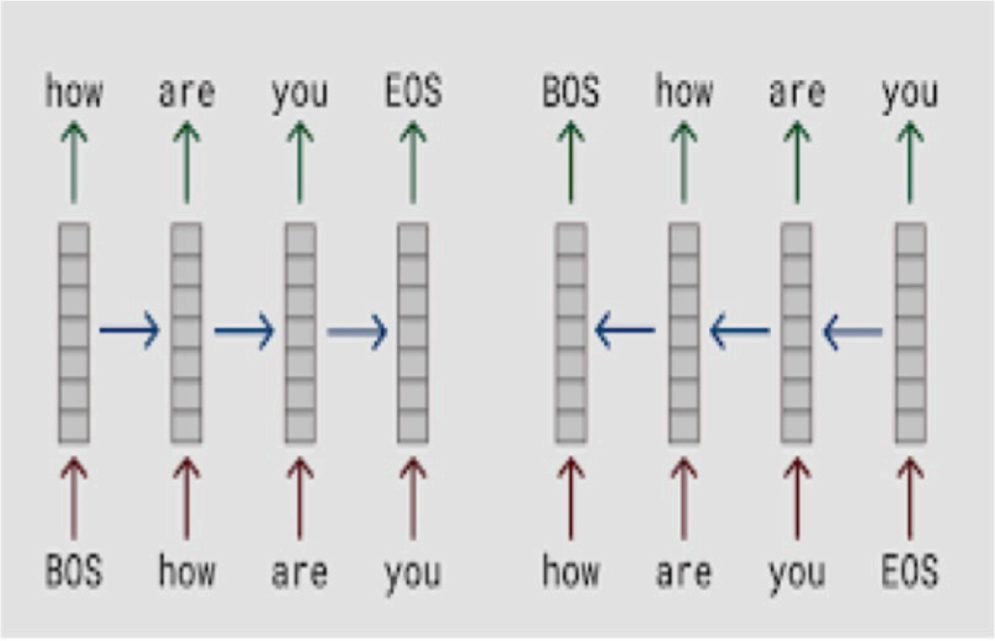
- BERT is trained with two techniques called Masked Language Modeling (MLM) and Next Sentence Prediction (NSP), apart from being bidirectional.

Bidirectional Encoder Representations From Transformers (BERT)



Bidirectional Encoder Representations From Transformers (BERT)

BERT



- BERT use bidirectional transformer (**both left-to-right and right-to-left direction**) rather than unidirectional transformer (left-to-right direction).

Bidirectional Encoder Representations From Transformers (BERT)

BERT

BERT can be used for;

Machine Translation,

Question Answering,

Sentence Similarty,

Prediction Next Token and Sentences,

Text, and **Token classification (NER)**.

Bidirectional Encoder Representations From Transformers (BERT)

Preparing the Data for The BERT Model

- In BERT models, the maximum length of sentences is **512 tokens** and each token is represented as a **768**-dimensional vector (**BERT-Base**), **1024**-dimensional vector (**BERT-Large**).

- All inputs given to BERT models must be of **fixed length**, as with RNNs.

Bidirectional Encoder Representations From Transformers (BERT)

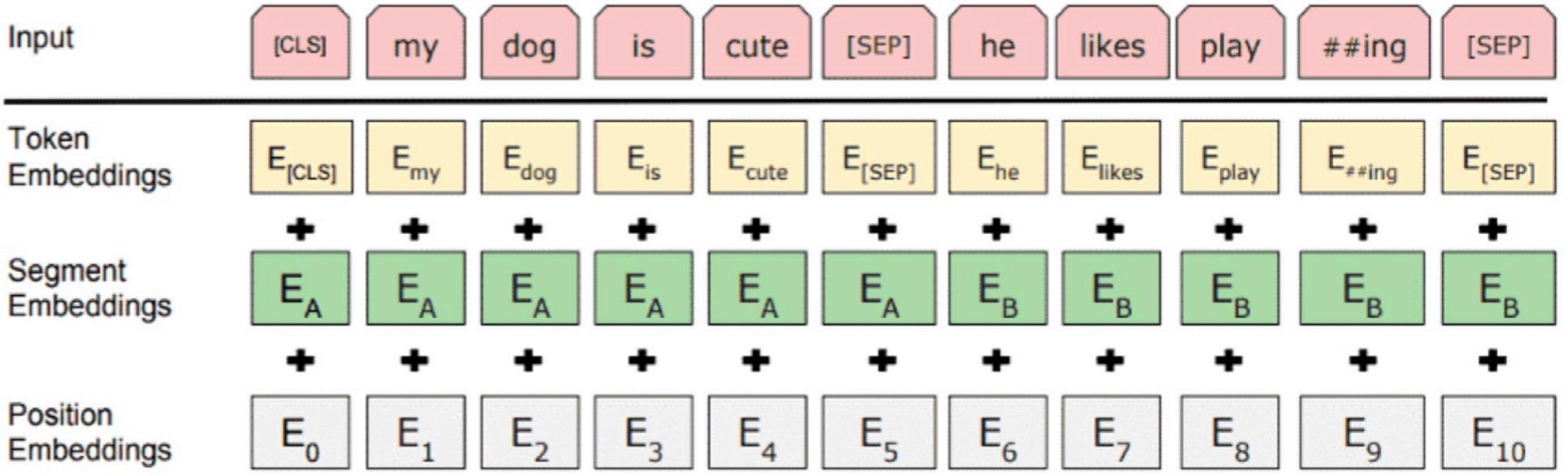
Preparing the Data for The BERT Model

- The tokenization in BERT is done using a method called **WordPiece** tokenization.

- In this method, tokens are tokenized separately according to their origin and attachment. For example, strawberry is expressed as two different tokens (**straw** and **##berry**) as root and suffixes

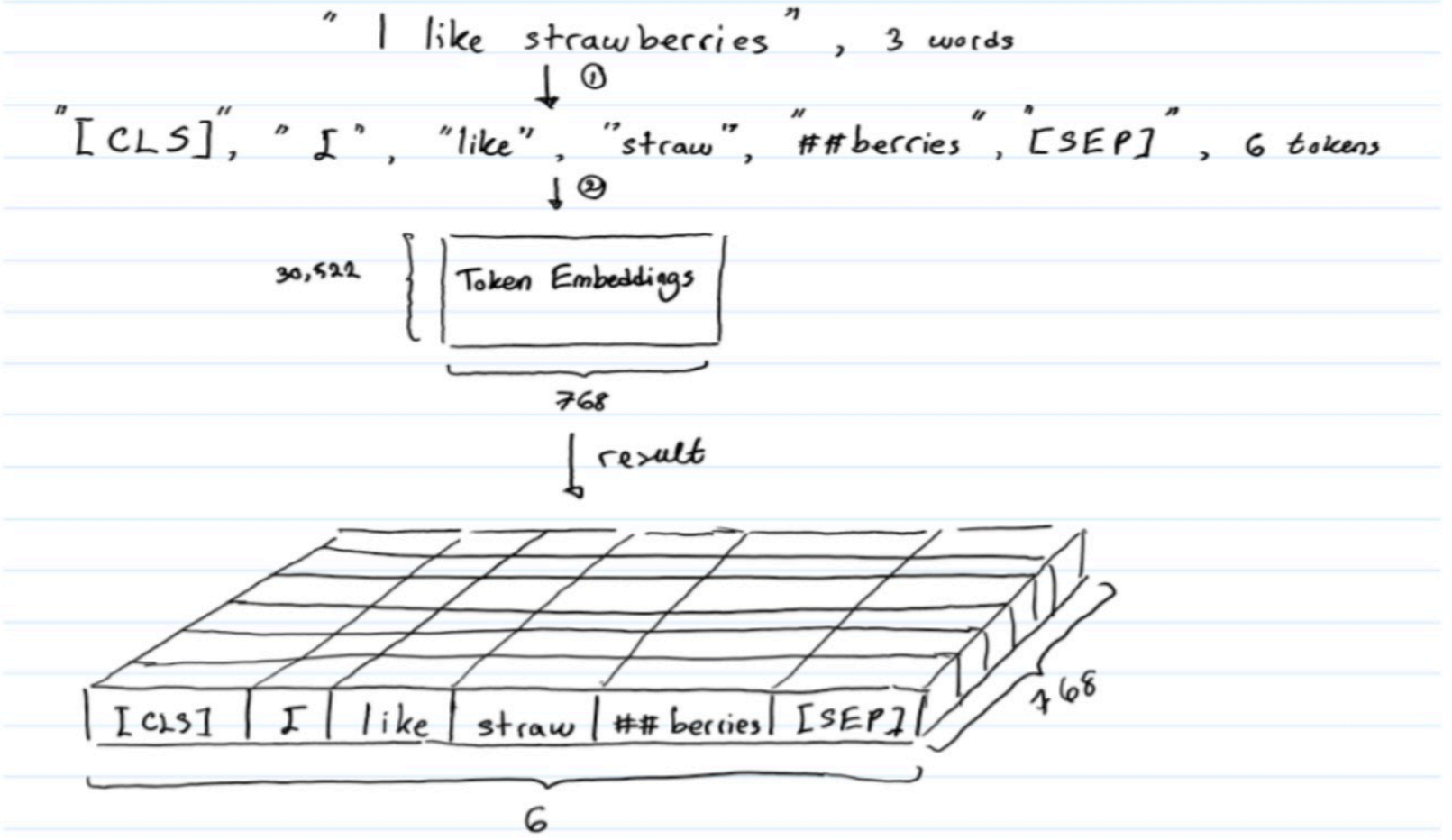
Bidirectional Encoder Representations From Transformers (BERT)

Preparing the Data for The BERT Model



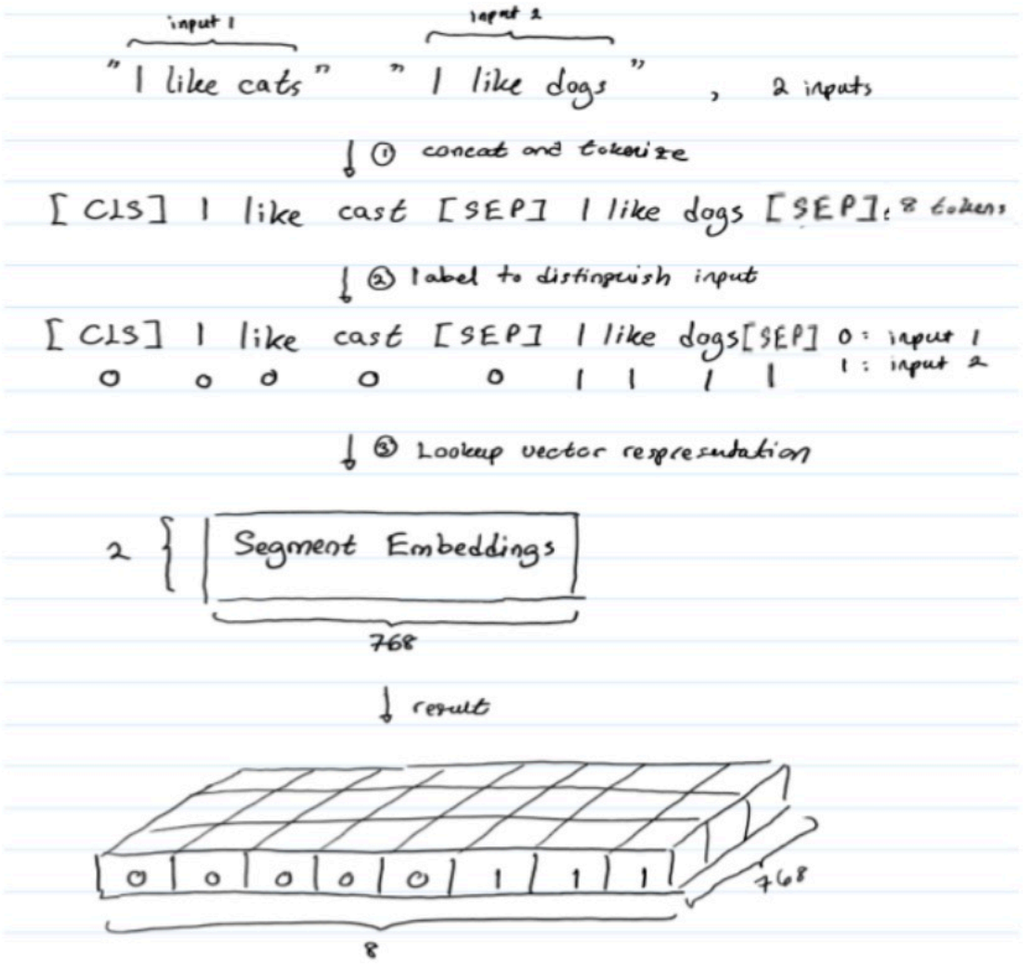
Bidirectional Encoder Representations From Transformers (BERT)

Token Embeddings (input_ids)



Bidirectional Encoder Representations From Transformers (BERT)

Segment Embeddings (token_type_ids)



- For Detecting the semantic similarities of two different sentences or tokens, translating a sentence or predicting the next sentence, etc. segment embedding layer is used.

- **It is not used for text and token classification.**

Bidirectional Encoder Representations From Transformers (BERT)

Position Embeddings (attention_mask)

```
sentence = "Şentepeli Şükrü abi?".lower()
tokens = tokenizers.encode_plus(sentence, add_special_tokens=True)
print(tokens)
```

```
{'input_ids': [2, 14192, 5364, 2031, 9204, 9025, 13780, 35, 3], 'token_type_ids': [0, 0, 0, 0, 0, 0, 0, 0, 0], 'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1]}
```

- **The order information of tokens** in the text is encoded **into the word embeddings that represent the 1s**.

- In this way, **the order of tokens is determined by the model**.