```python
def _iter_():
    for i, (w, c) in enumerate(gen_points()):
        if num_workers == worker_id:
            for
                +5 neg
            yield (pos, 5 neg)
```

Transformer

Encoder

Decoder

Stack by transformer layer

6 or 12 layers

Updated embedding
Position encoding (only for the first layer)

\[ W : q \in \mathbb{R}^d, \quad W^k : \text{keys}, \quad W_v : \text{values} \]

Map to 8 subspaces

\[ d = 64 = \frac{512}{8} \]

Attention:

\[ \tilde{q}_j = d + \langle q_k, k_j \rangle \]

\[ \tilde{v}_c = \sum a_{cj} \tilde{v}_j \]

8 such spaces / heads

Projection matrices:

\[ \begin{array}{cccc}
W_1 & w_2 & \cdots & w_q \\
1 & k & k & k
\end{array} \]
BERT: Masked language model (MLM)

Wordvec: Skip-gram model

Q: Is the loss only from the masked words?

The whole sentence

False

The [MASK] in the book

16 heads of attention
340 million parameters

The Transformer

The [MASK] in [MASK] book

The [MASK] in [COR] book

The [MASK] in [EMB] book

The [MASK] in [EMB] book

The [MASK] in [EMB] book

The [MASK] in [EMB] book

12 layers
8 Transformer

Model: 1024

The [MASK] in [MASK] book

Money the true words

25%: MLM

572 words

15% of words will be chosen to be masked

80% replaced with [MASK]

10% replace with another

10% leave it as it is
Task 2: Next sentence prediction

data:

\[ s_1, s_2 \rightarrow \text{true} \]
\[ s_1, s_2, s_100 \rightarrow \text{false} \]

sentence

\[ s_3 \]
\[ s_4 \]

Mask or \( k \) layers

24 layers

\[ s_1, s_2 \]

\[ \text{Bert} \]

Label \( 1 \rightarrow \text{true} \)
\( 0 \rightarrow \text{false} \)

Min loss

Mask our units

Next layer

\[ u_i \]