CSCI 5541: Natural Language Processing

Lecture 8: Contextualized Word Embeddings

Dongyeop Kang (DK), University of Minnesota

dongyeop@umn.edu | twitter.com/dongyeopkang | dykang.github.io

computer science & engineering

MINNESOTA NLP · EST. 2021

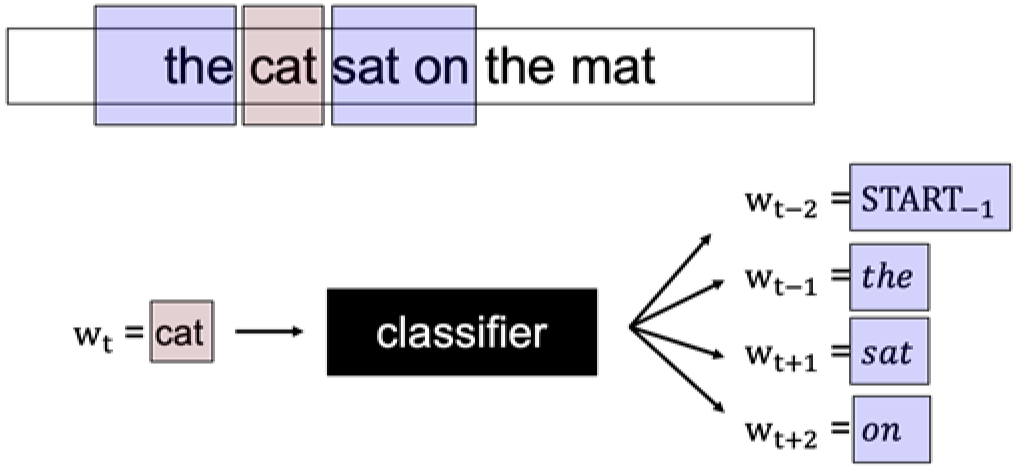
University of Minnesota
Driven to Discover®

Different kinds of encoding "context"

Recap

- ~~Count-based~~
  - PMI, TF-IDF
- ~~Distributed prediction-based (type) embeddings~~
  - Word2vec, GloVe, Fasttext

	Hamlet	Macbeth	Romeo & Juliet	Richard III	Julius Caesar	Tempest
knife	1	1	4	2		2
dog				6	12	2
sword	2	2	7	5		5
love	64		135	63		12
like	75	38	34	36	34	41
...						



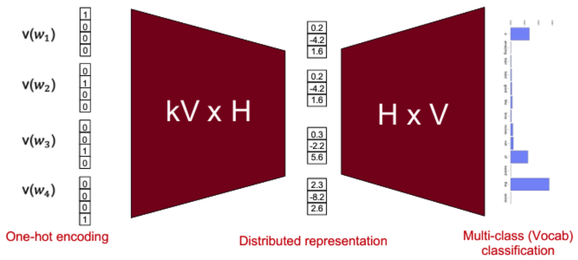
Different kinds of encoding “context”

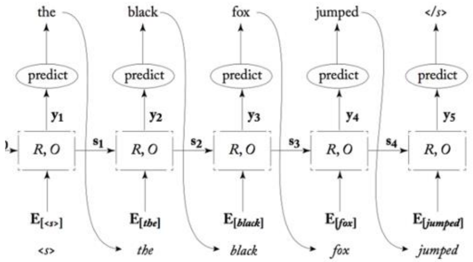
- ❑ ~~Count-based~~
  - PMI, TF-IDF
- ❑ ~~Distributed prediction-based (type) embeddings~~
  - Word2vec, GloVe, Fasttext
- ❑ **Distributed contextual (token) embeddings from language models**
  - ELMo, BERT, GPT
- ❑ Many more variants
  - Multilingual / multi-sense / syntactic embeddings, etc

$$\text{Unigram LM} : p(w_1^N) = \prod_{n=1}^{N} p(w_n)$$

$$\text{Bigram LM} : p(w_1^N) = \prod_{n=1}^{N} p(w_n | w_{n-1})$$

$$\text{Trigram LM} : p(w_1^N) = \prod_{n=1}^{N} p(w_n | w_{n-2}, w_{n-1})$$





Types and tokens

Type: gopher

5.2	1.5	...	0.2	0.6

Token:

- The gopher is a resident of the dry plains.
- One day, while I was out chasing a gopher, I wandered off too far.
- It's not often a team loses with stats like this. Gophers played very well tonight.

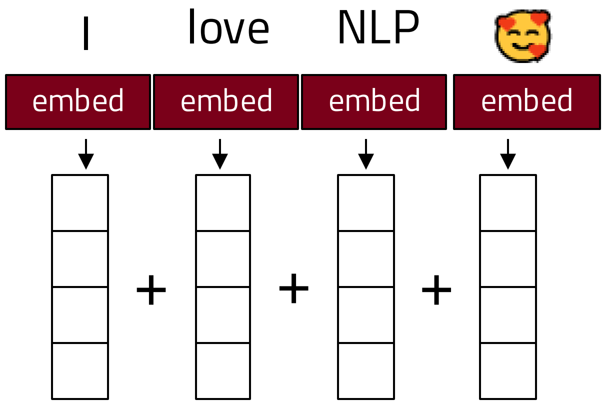
"gopher"

5.2	1.5	...	0.2	0.6

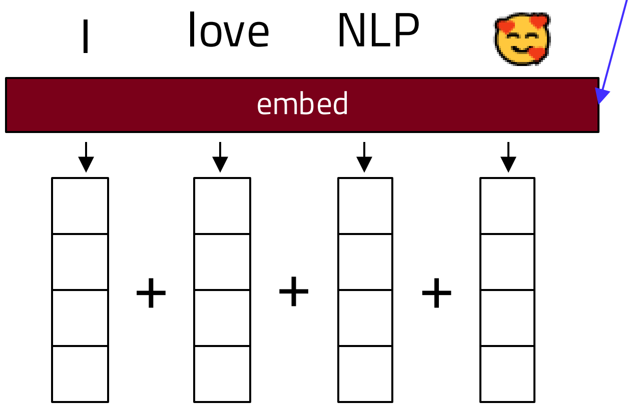
3.2	8.5	...	0.6	8.1

-2.2	2.4	...	5.2	3.4

Contextualization of word representations

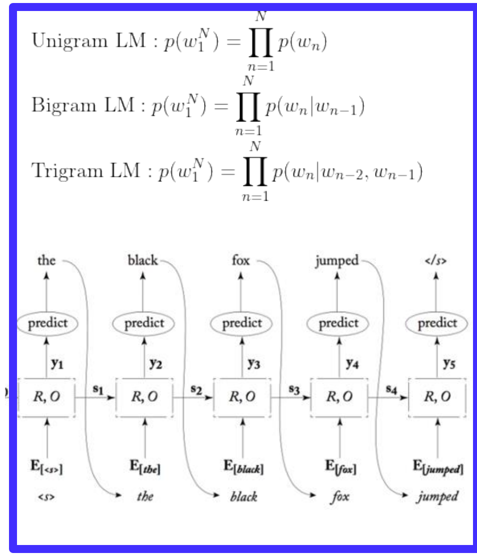


Static or non-contextualized representations



Contextualized representations

Language models



Contextualized word representations

Transform the representation of a token in a sentence (e.g., from a static word embedding) to be sensitive to its local context in a sentence

ELMo

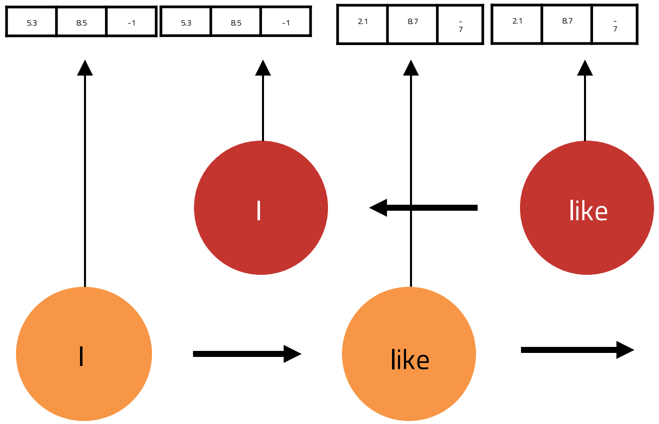
(Peters et al., 2018)

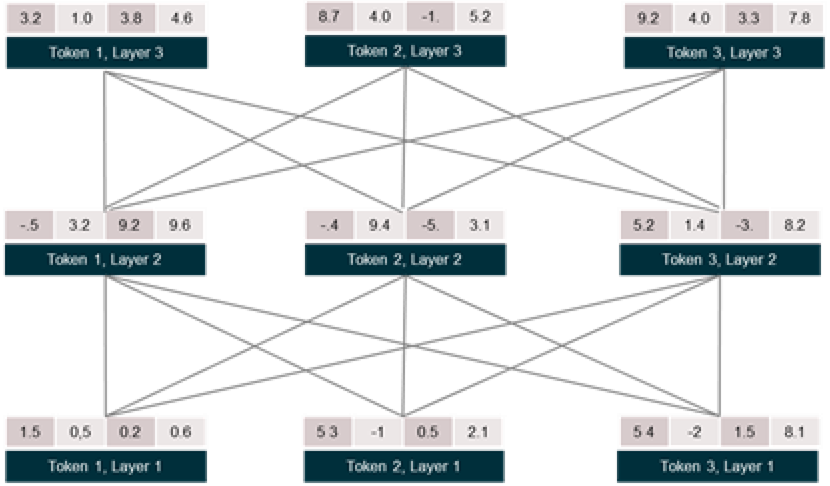
BERT

(Devlin et al., 2019)

Stacked Bidirectional RNN trained to predict next word in language modeling task

Transformer-based model to predict masked word using bidirectional context and next sentence prediction





ELMo

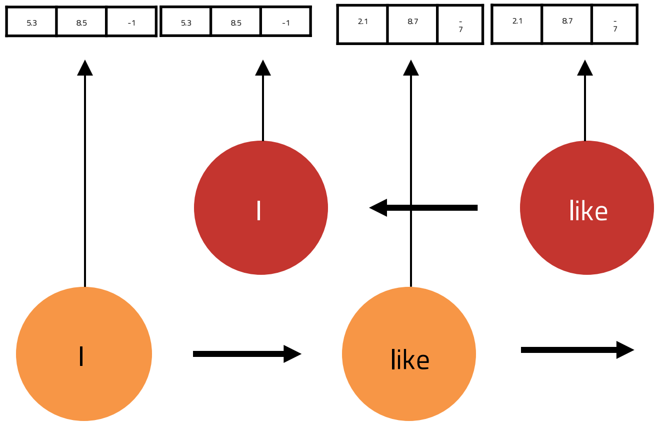
(Peters et al., 2018)

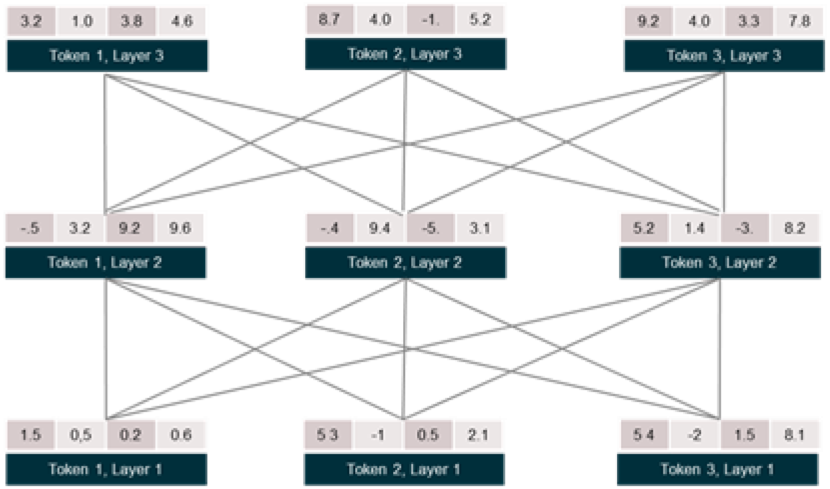
BERT

(Devlin et al., 2019)

Stacked Bidirectional RNN trained to predict next word in language modeling task

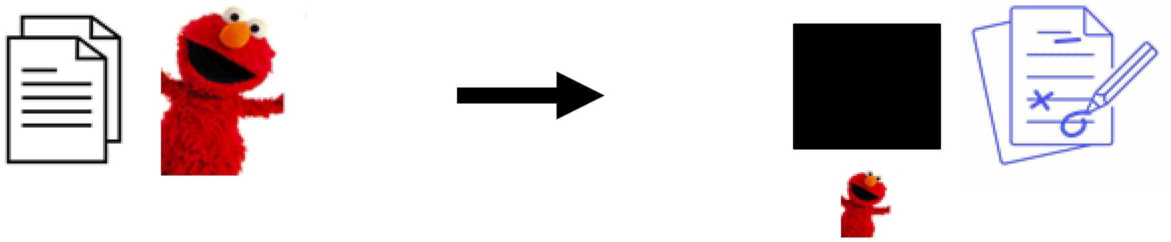
Transformer-based model to predict masked word using bidirectional context and next sentence prediction





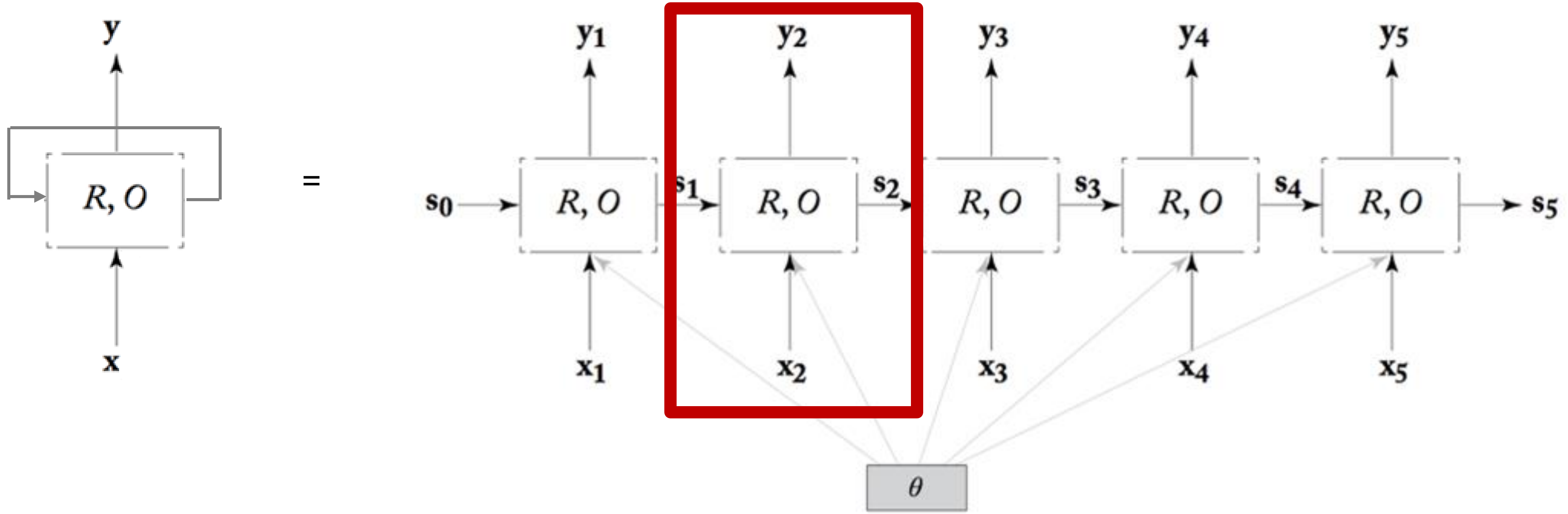
ELMo (Embeddings from Language Models)

- Big idea: (i) transform the representation of a word (e.g., from a static word embedding) to be sensitive to its local context in a sentence and (ii) optimized for a specific NLP task.
- Output = word representations that can be plugged into just about any architecture a word embedding can be used.



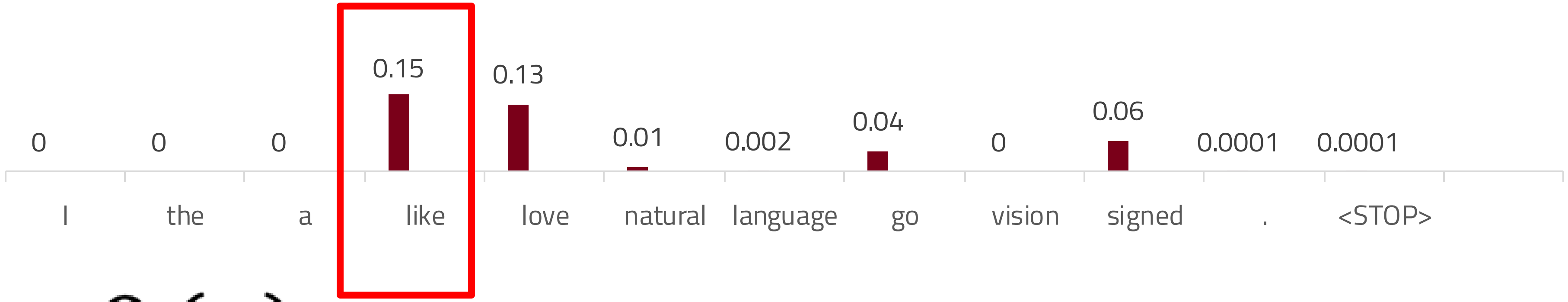
Recurrent Neural Network

RNN allow arbitrarily-sized conditioning contexts;
condition on the entire sequence history.

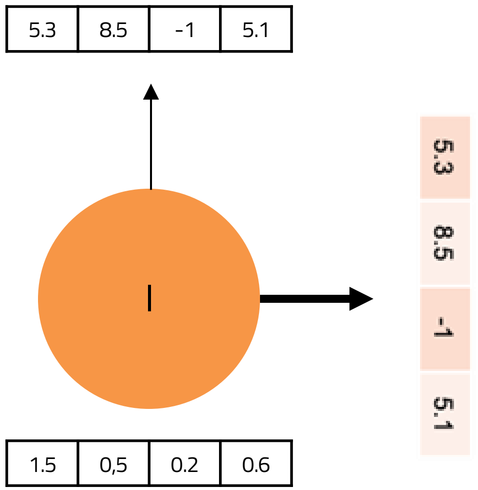


Goldberg 2017

Recurrent neural network language model

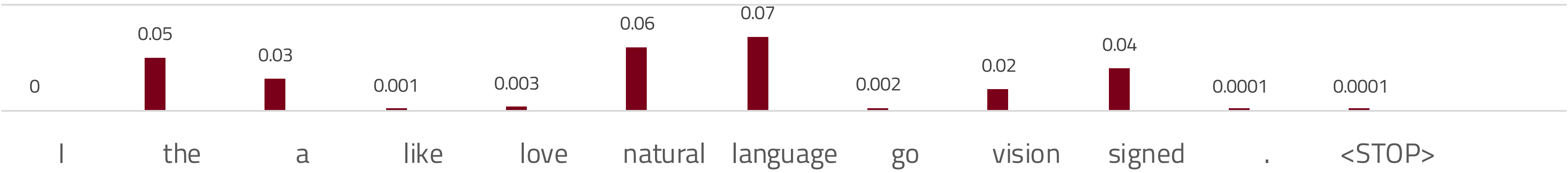


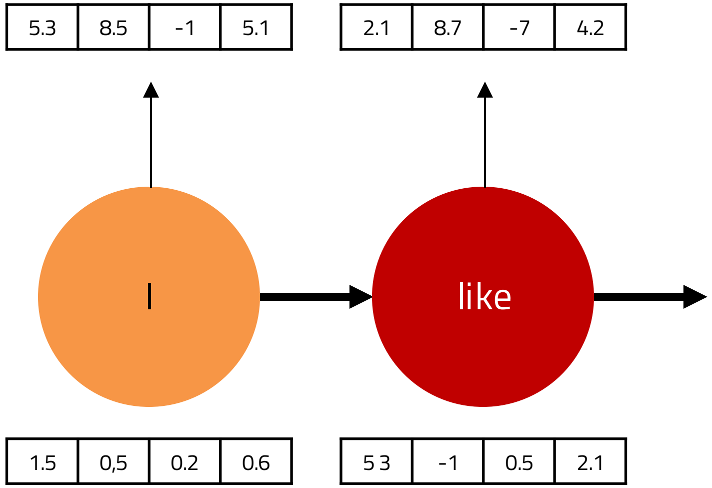
$$y_i = O(s_i)$$



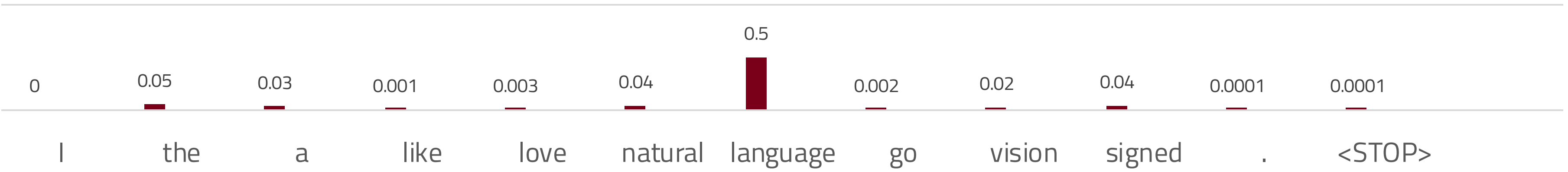
$$s_i = R(x_i, s_{i-1})$$

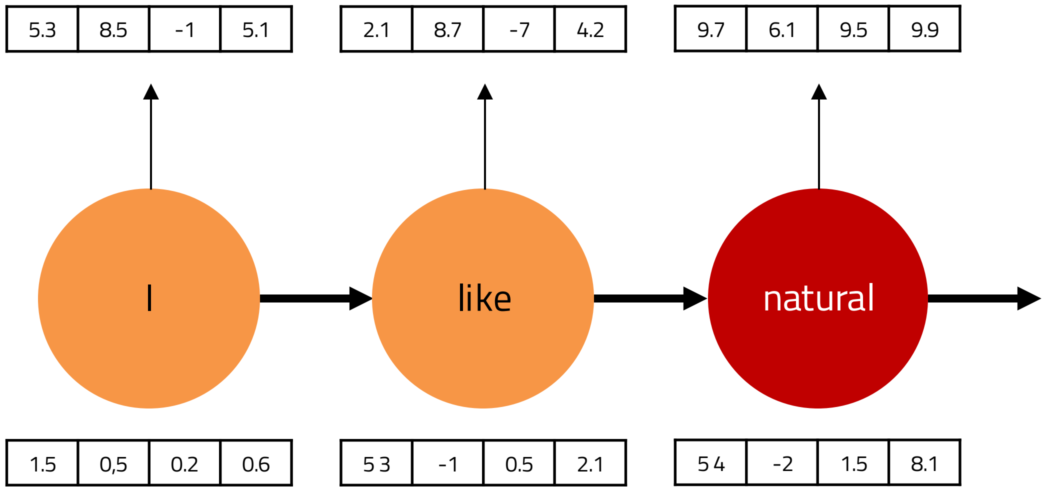
Recurrent neural network language model



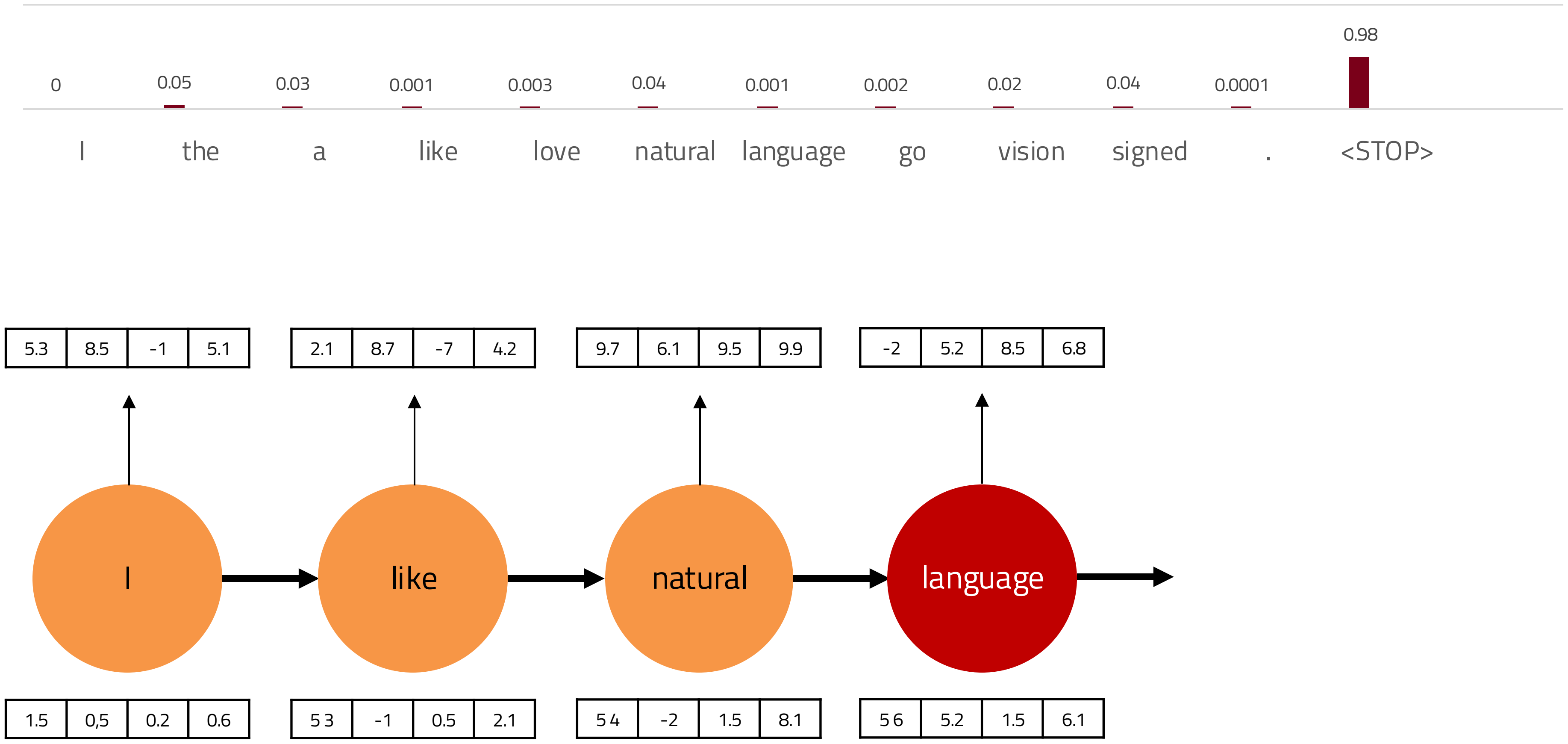


Recurrent neural network language model

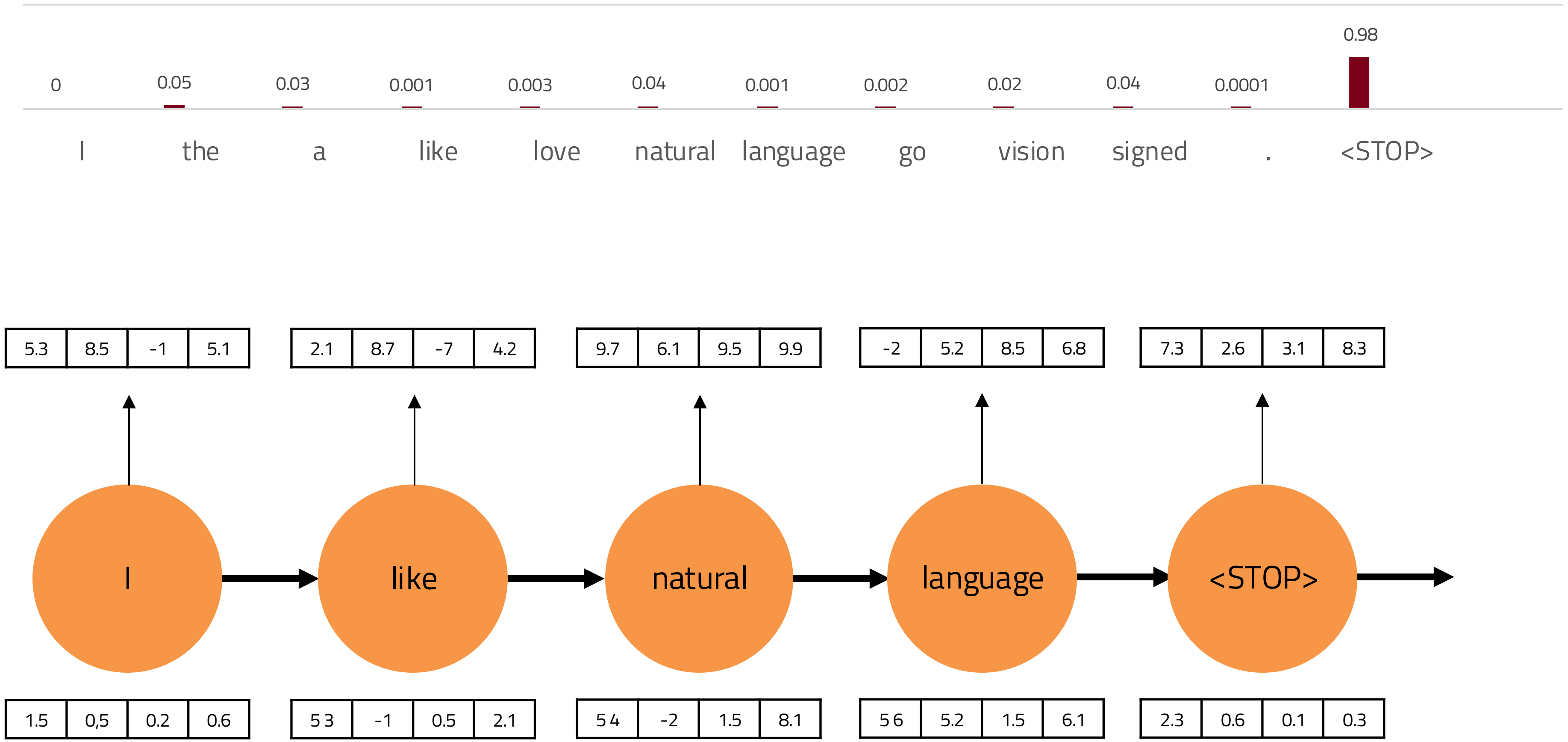




Recurrent neural network language model

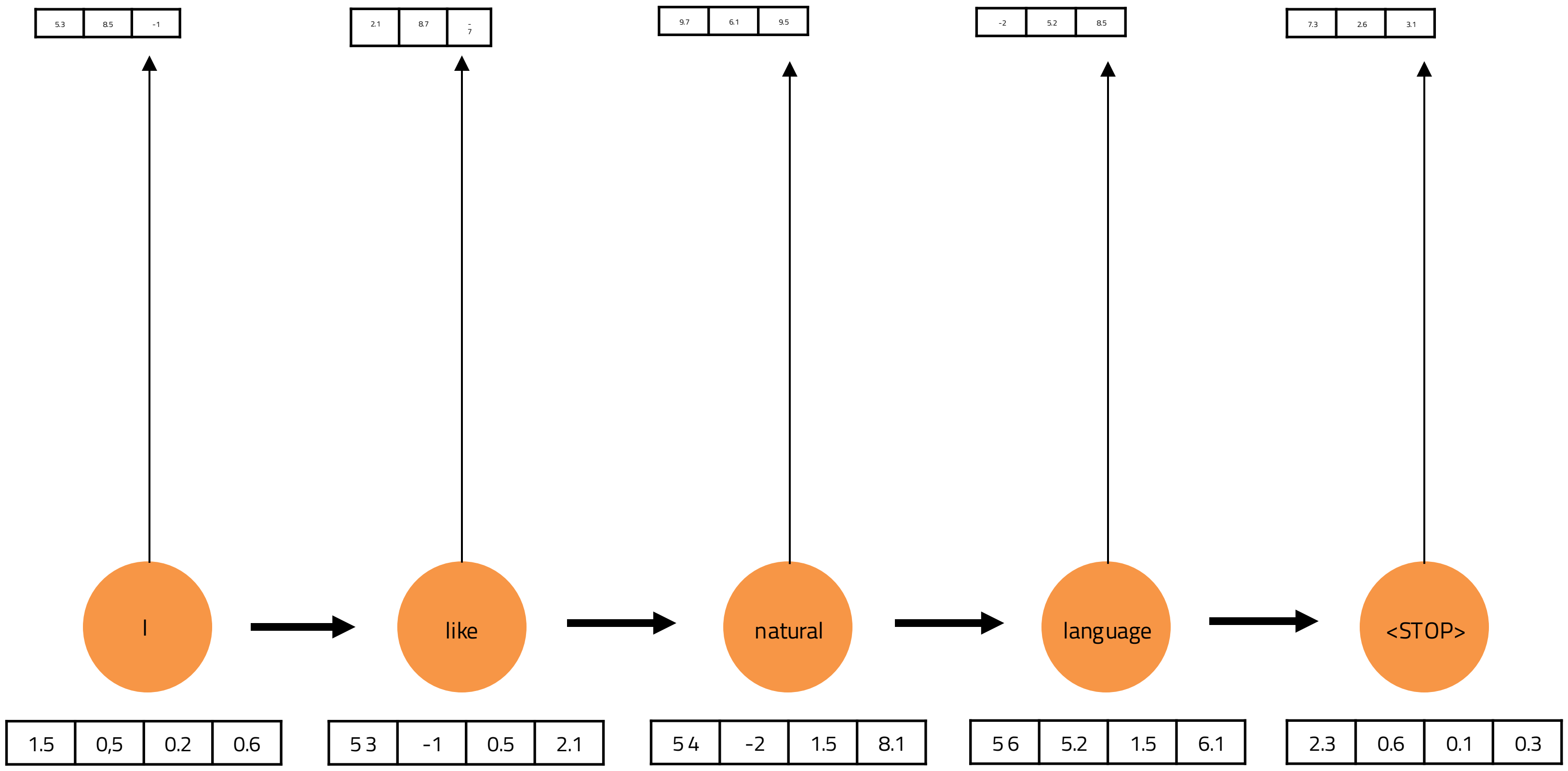


Recurrent neural network language model



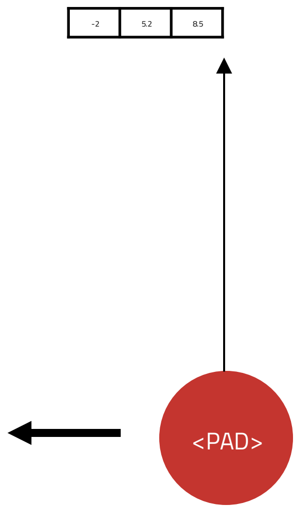
Bidirectional RNN

Forward RNN



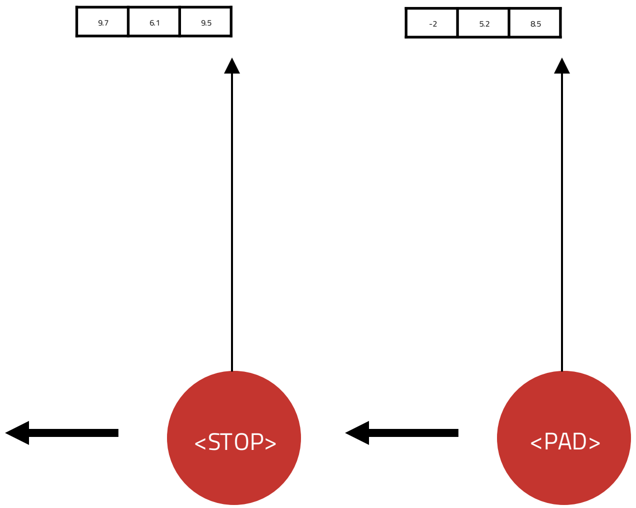
Bidirectional RNN

Backward RNN



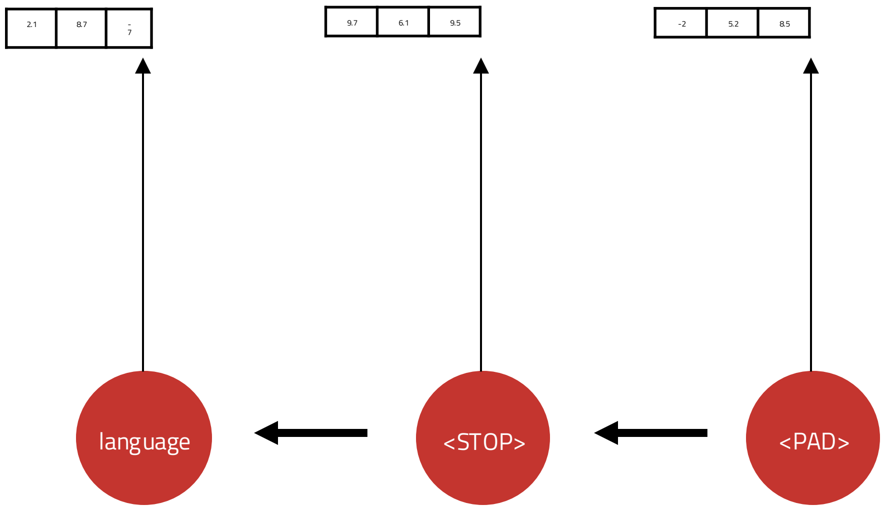
Bidirectional RNN

Backward RNN



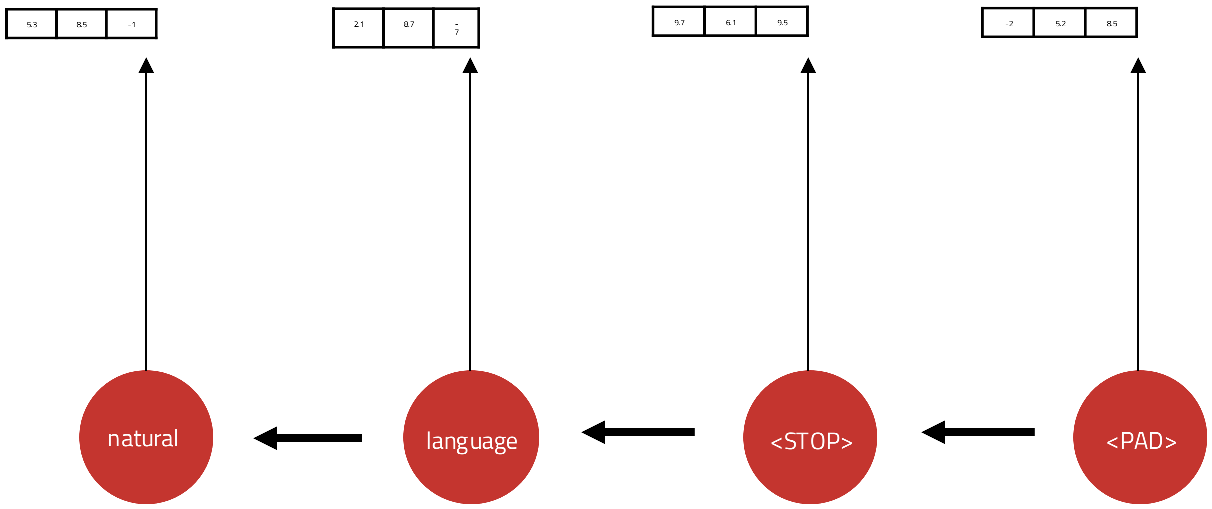
Bidirectional RNN

Backward RNN

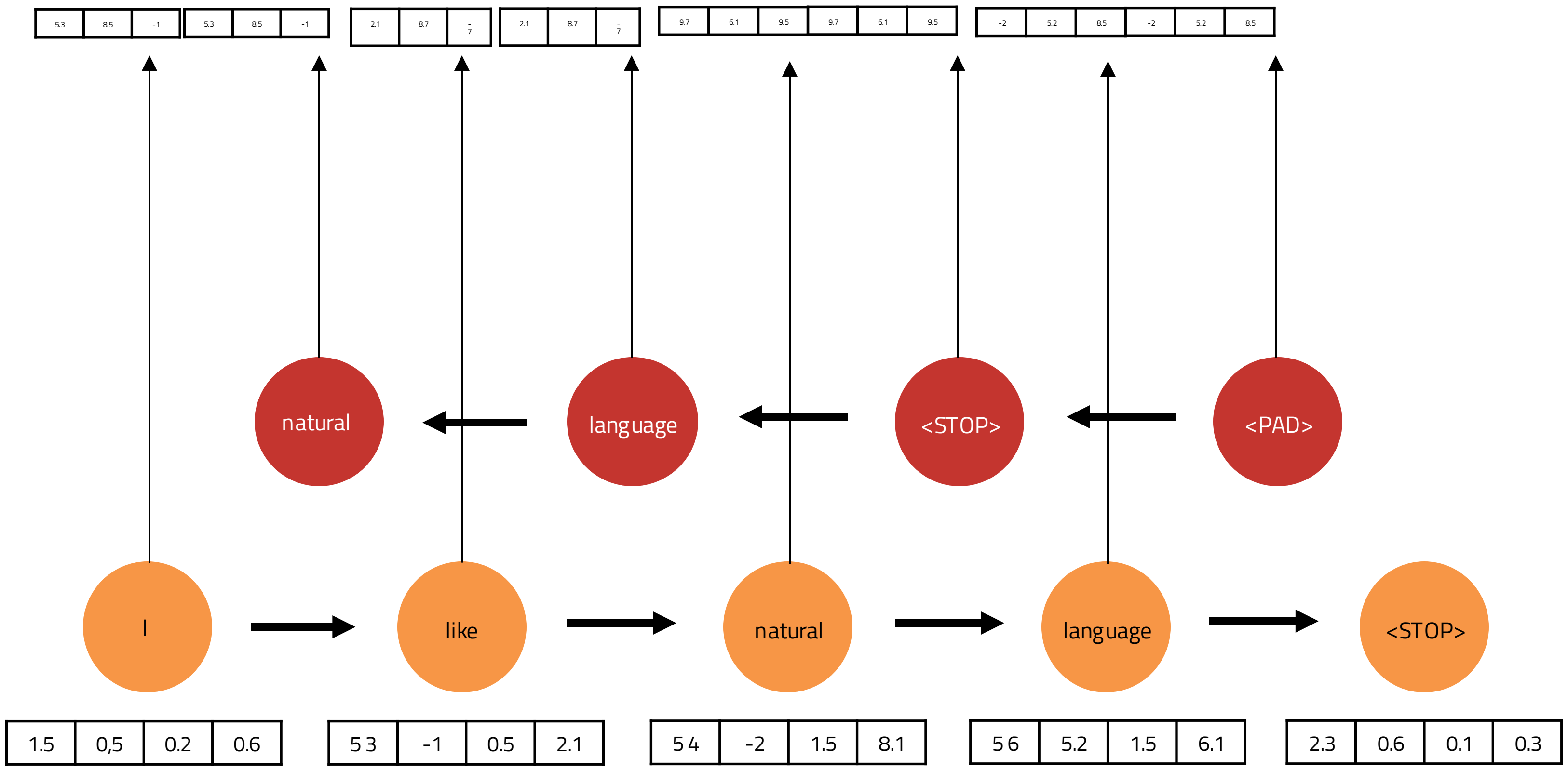


Bidirectional RNN

Backward RNN

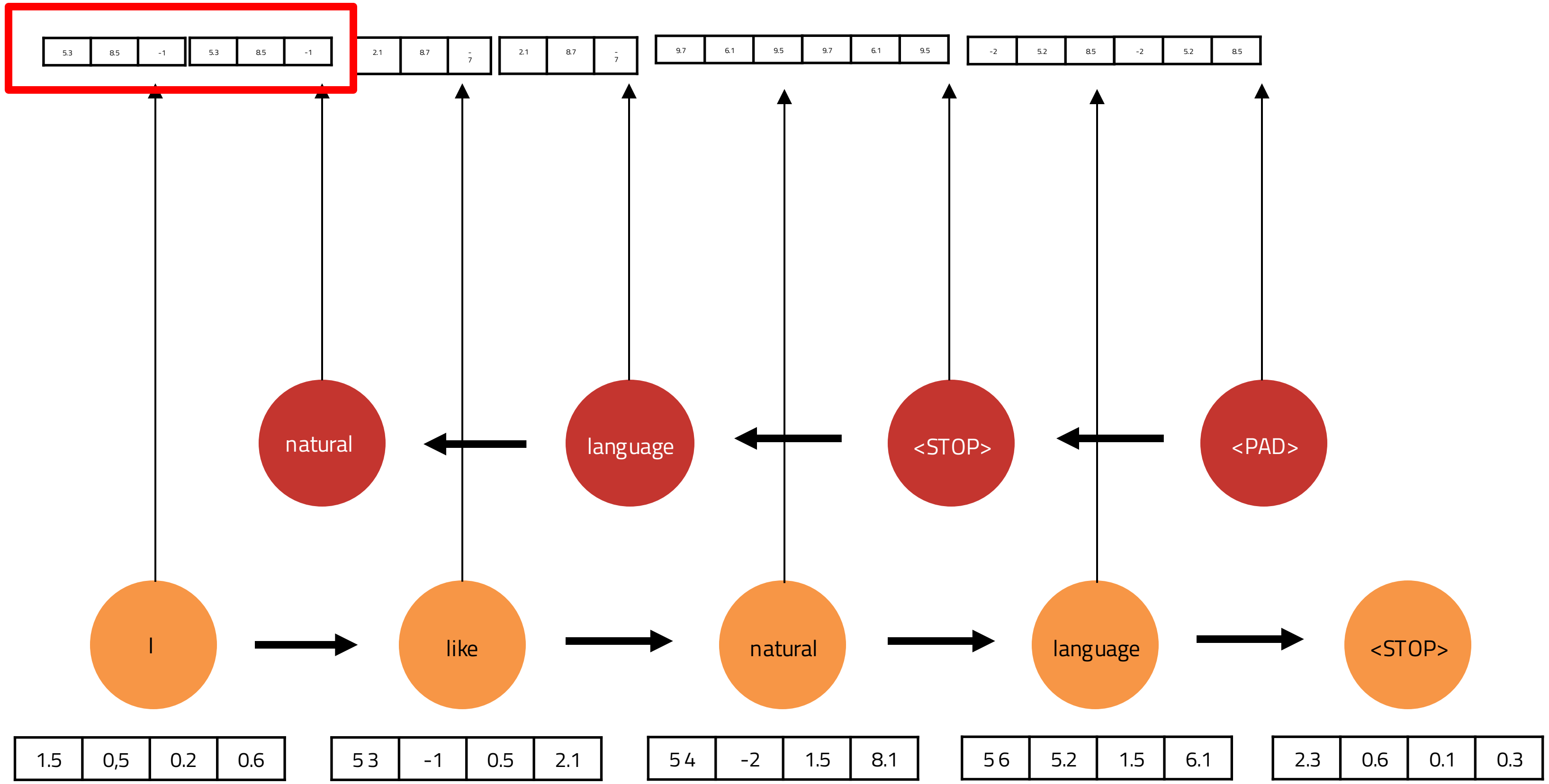


Bidirectional RNN



Bidirectional RNN

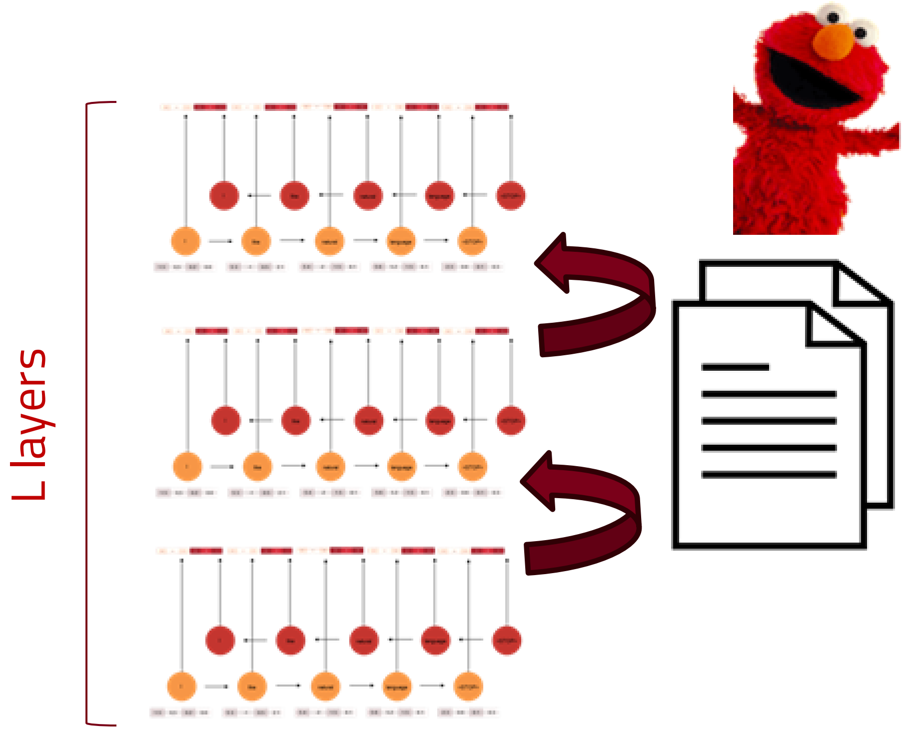
$$y_i = O\,(s_i^f ; s_i^b)$$

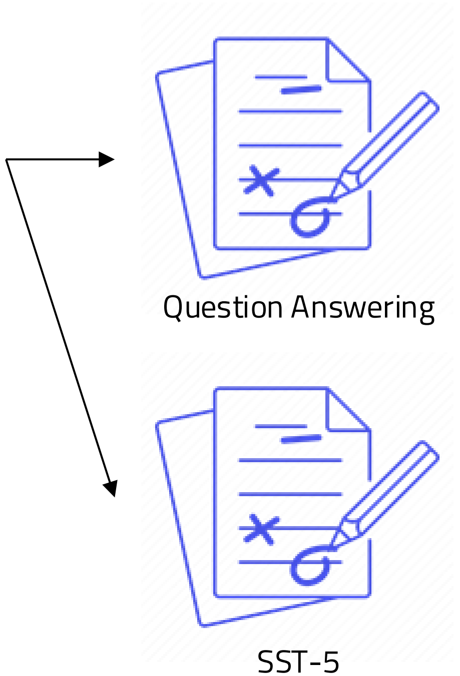


ELMo (Embeddings from Language Models)

Pre-training stage:
Train a Bi-RNN LM with L layers
on unlabeled text corpora

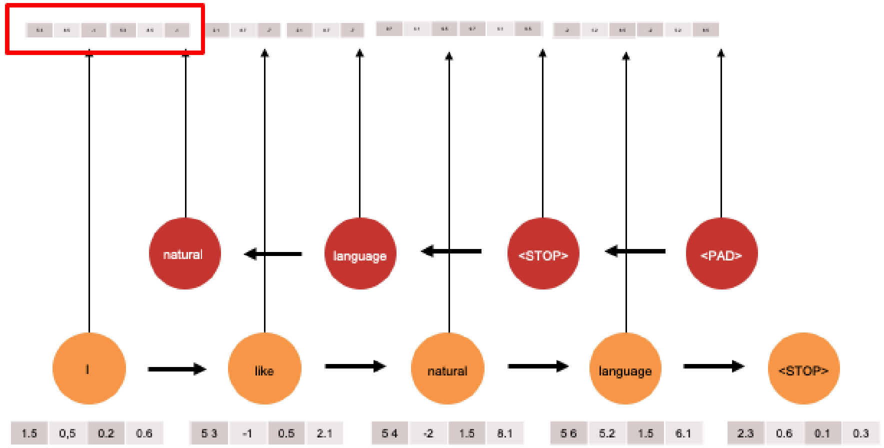
Fine-tuning stage:
Fine-tune it for a specific task by combining
RNN output across all layers





TASK	ELMO + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
SQuAD	85.8	4.7 / 24.9%
SNLI	88.7 ± 0.17	0.7 / 5.8%
SRL	84.6	3.2 / 17.2%
Coref	70.4	3.2 / 9.8%
NER	92.22 ± 0.10	2.06 / 21%
SST-5	54.7 ± 0.5	3.3 / 6.8%

Types and tokens



Type: gopher

5.2	1.5	...	0.2	0.6

Token:

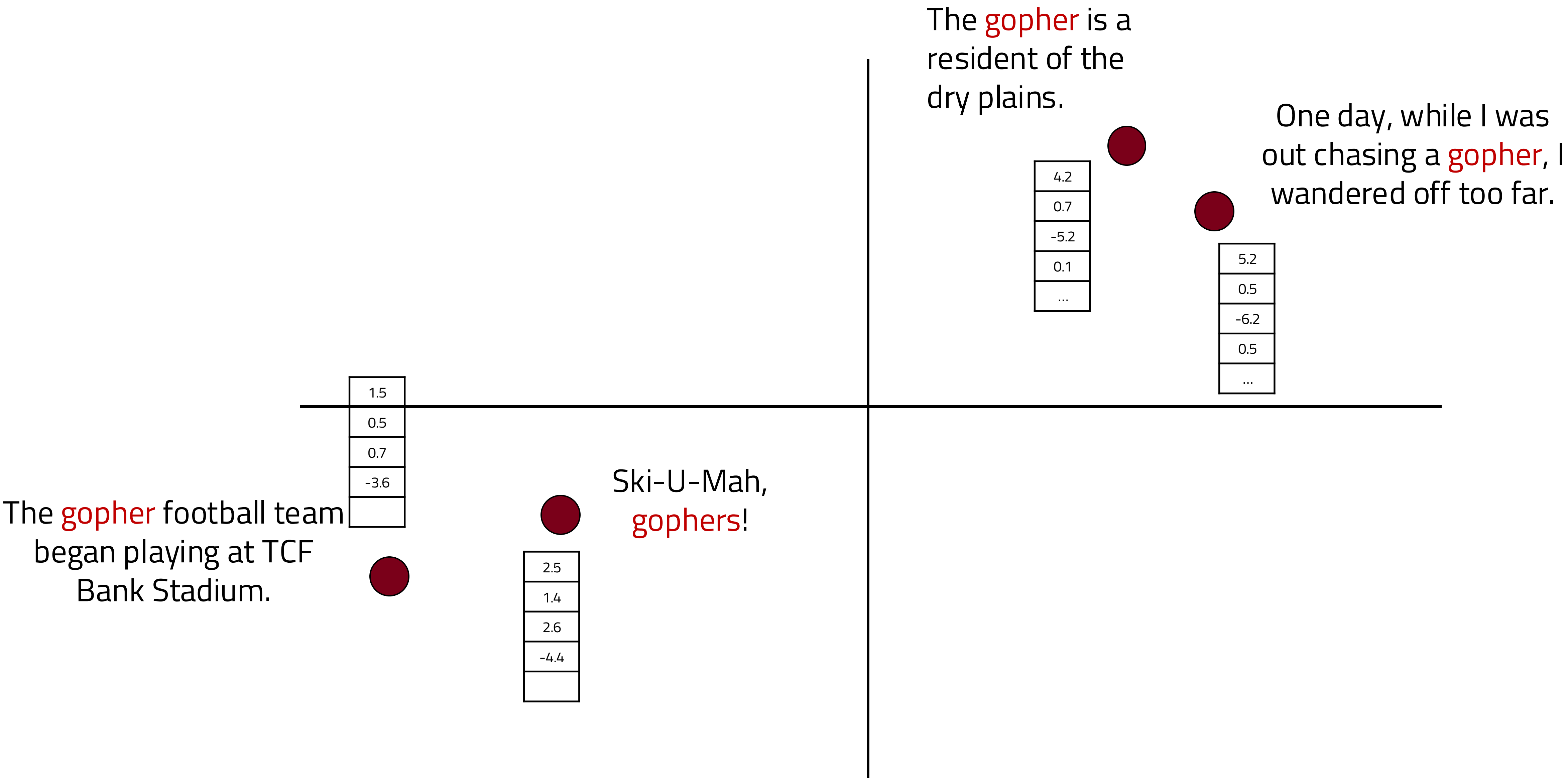
- The gopher is a resident of the dry plains.
- One day, while I was out chasing a gopher, I wandered off too far.
- It's not often a team loses with stats like this. Gophers played very well tonight.

"gopher"

5.2	1.5	...	0.2	0.6

3.2	8.5	...	0.6	8.1

-2.2	2.4	...	5.2	3.4


The gopher is a resident of the dry plains.
4.2
0.7
-5.2
0.1
...
One day, while I was out chasing a gopher, I wandered off too far.
5.2
0.5
-6.2
0.5
...
1.5
0.5
0.7
-3.6
The gopher football team began playing at TCF Bank Stadium.
Ski-U-Mah, gophers!
2.5
1.4
2.6
-4.4

ELMo

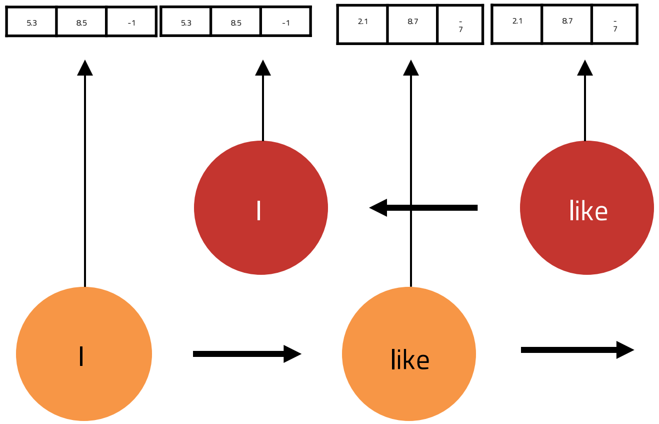
(Peters et al., 2018)

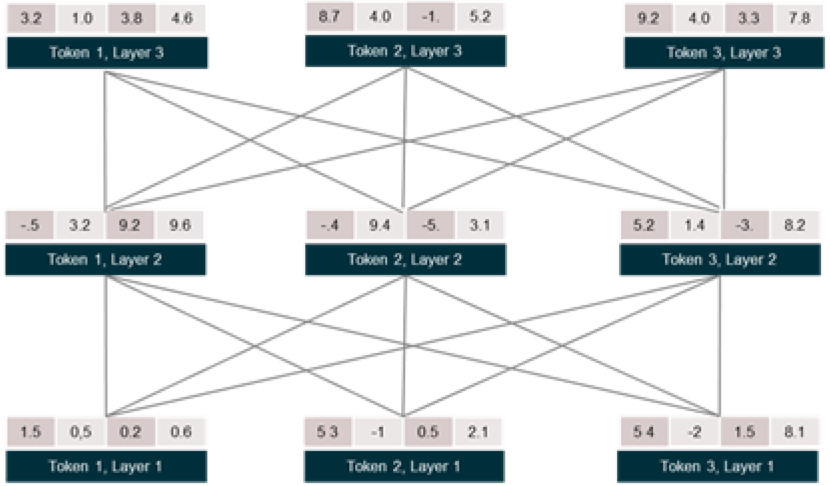
BERT

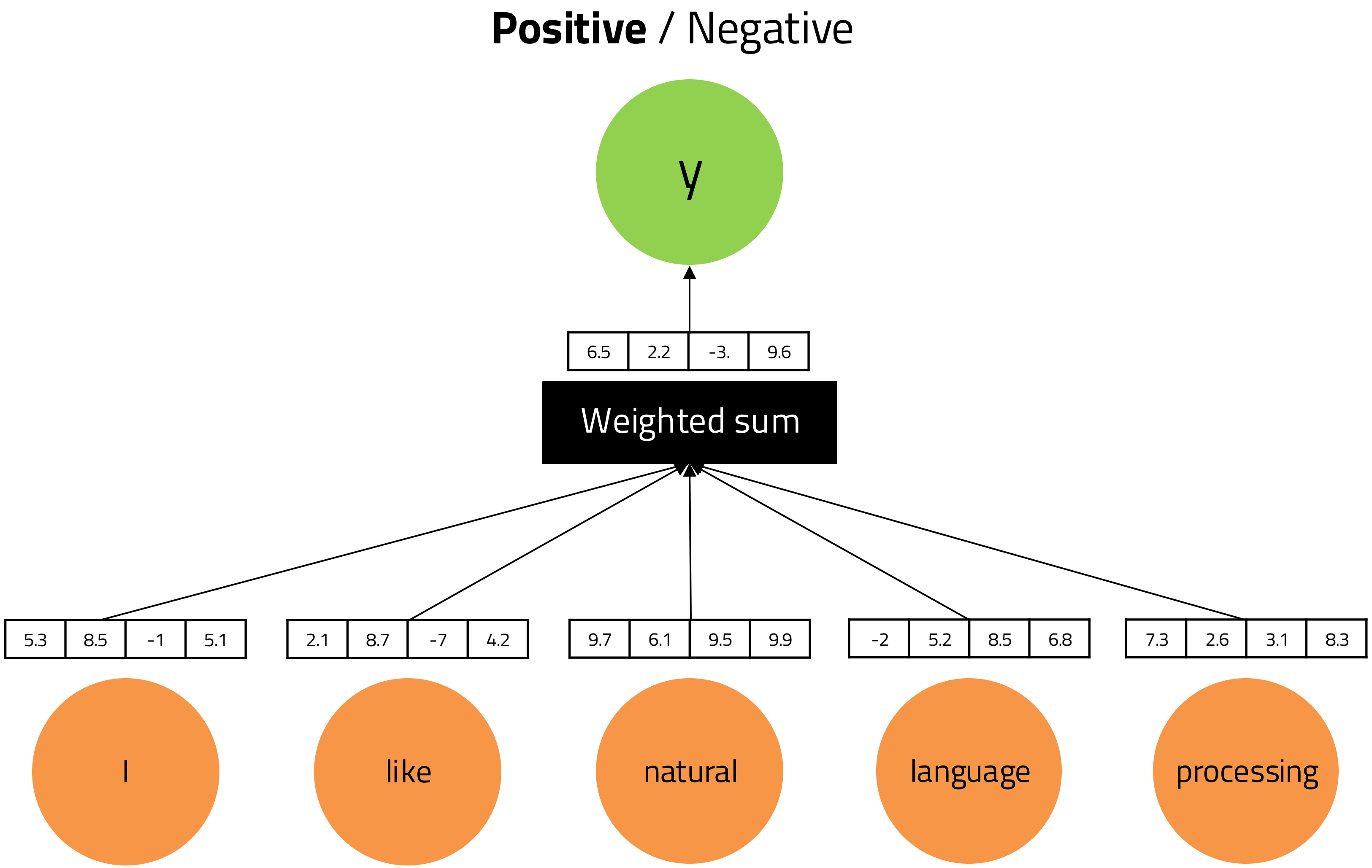
(Devlin et al., 2019)

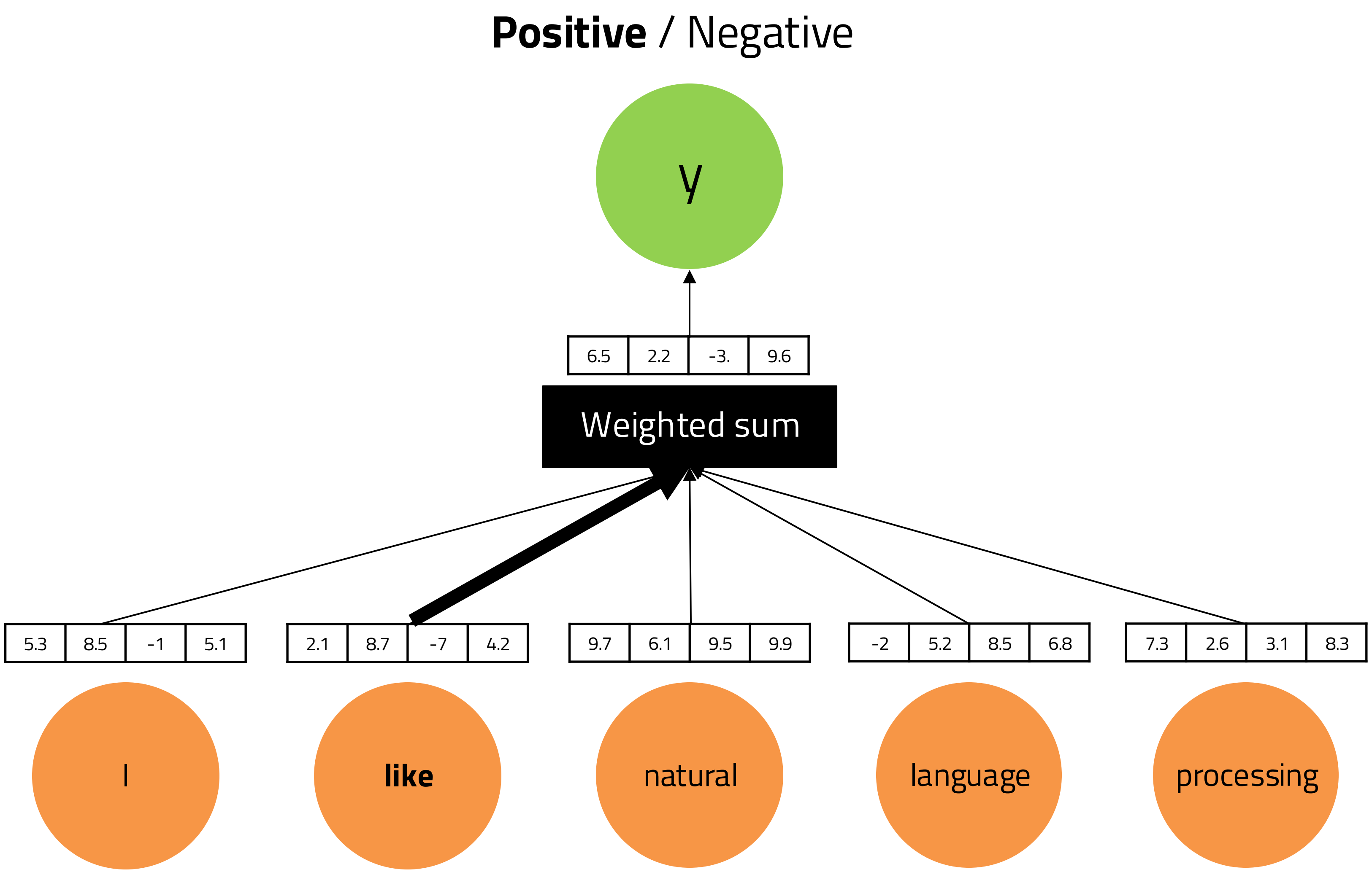
Stacked Bidirectional RNN trained to predict next word in language modeling task

Transformer-based model to predict masked word using bidirectional context and next sentence prediction






Positive / Negative
y
6.5 | 2.2 | -3. | 9.6
Weighted sum
5.3 | 8.5 | -1 | 5.1
2.1 | 8.7 | -7 | 4.2
9.7 | 6.1 | 9.5 | 9.9
-2 | 5.2 | 8.5 | 6.8
7.3 | 2.6 | 3.1 | 8.3
I
like
natural
language
processing



Positive / Negative
y
6.5 2.2 -3. 9.6
Weighted sum
5.3 8.5 -1 5.1
2.1 8.7 -7 4.2
9.7 6.1 9.5 9.9
-2 5.2 8.5 6.8
7.3 2.6 3.1 8.3
I
like
natural
language
processing

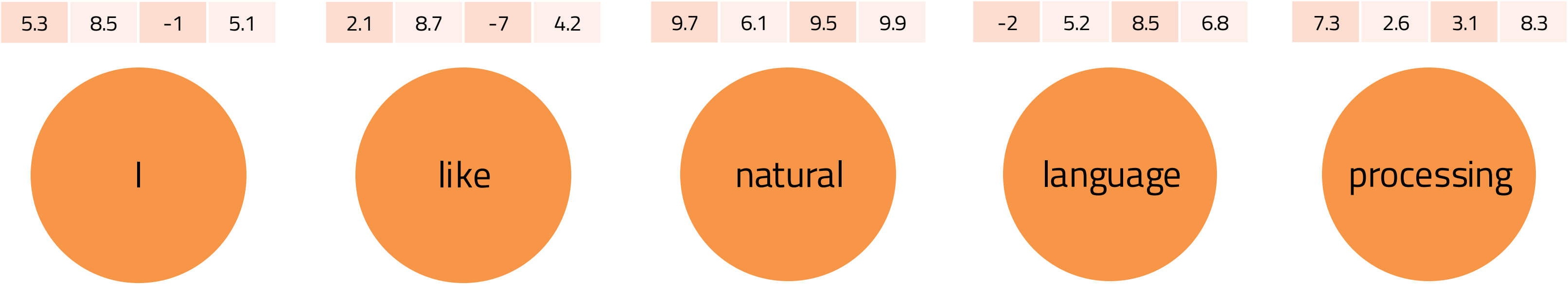
Attention

Incorporate structure (and parameters) into a network that captures which elements in the input we should be attending to (and which we can ignore).

$v \in R^h$

6.5	2.2	-3.	9.6

Define v be a vector to be learned; think of it as an "word importance" vector. The dot product measures how similar each input vector is to that "word importance" vector.



$v \in R^h$

6.5	2.2	-3.	9.6

$r_1 = -3.2$ $\quad r_2 = 2.4$ $\quad r_3 = -0.8$ $\quad r_4 = -1.2$ $\quad r_5 = 1.7$

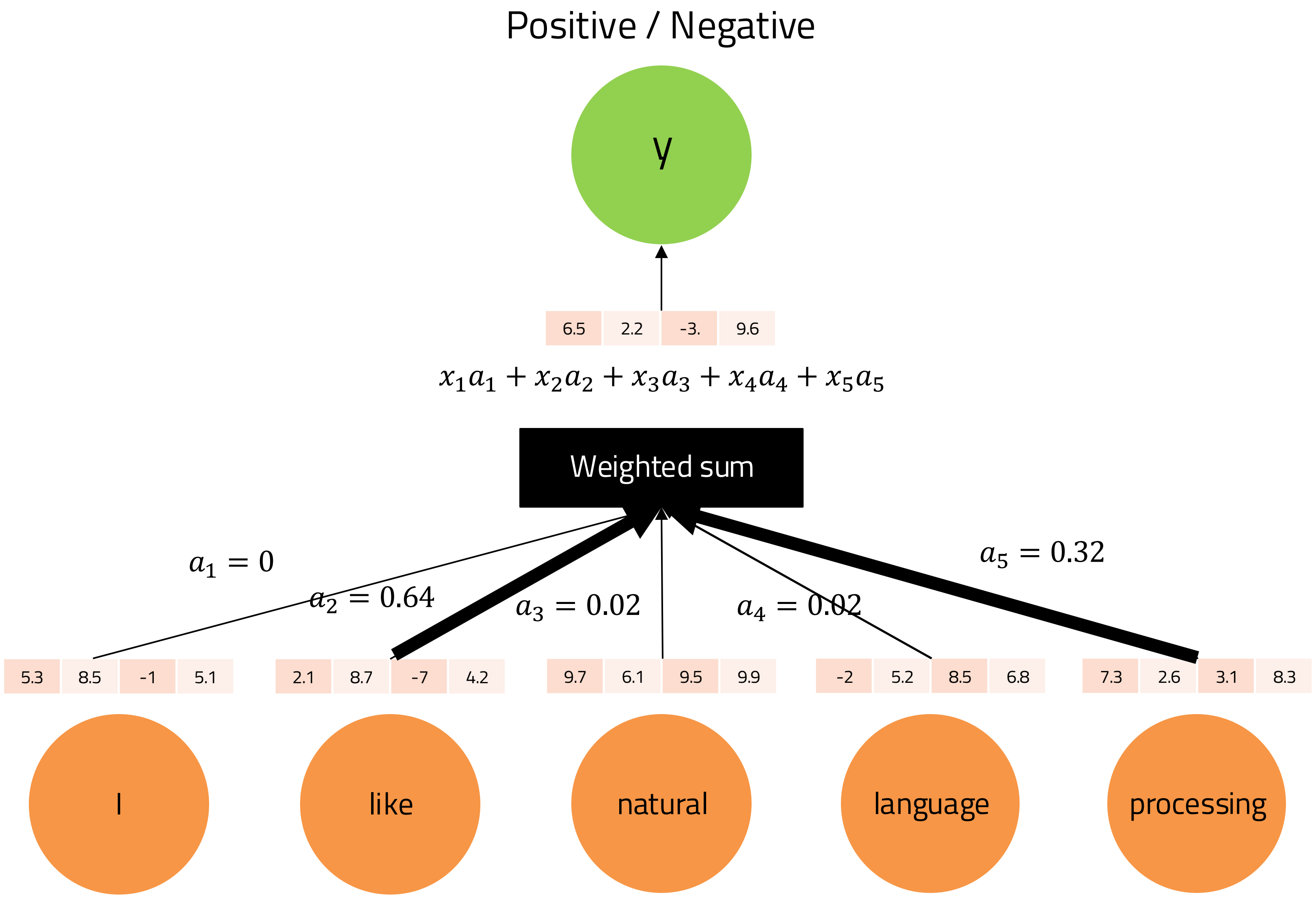
$r_1 = v^T x_1$ $\quad r_2 = v^T x_2$ $\quad r_3 = v^T x_3$ $\quad r_4 = v^T x_4$ $\quad r_5 = v^T x_5$

I	like	natural	language	processing
5.3 8.5 -1 5.1	2.1 8.7 -7 4.2	9.7 6.1 9.5 9.9	-2 5.2 8.5 6.8	7.3 2.6 3.1 8.3

Convert r into a vector of normalized weights that sum to 1.

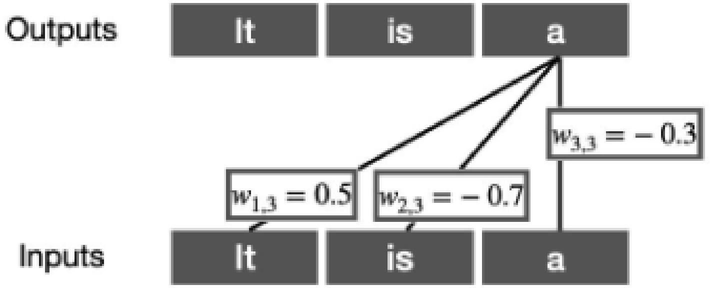
$$a = softmax\ (r)$$

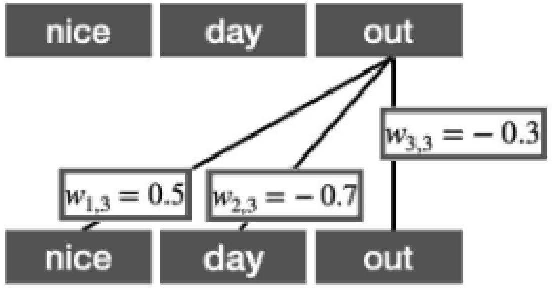
$a_1 = 0$	$a_2 = 0.64$	$a_3 = 0.02$	$a_4 = 0.02$	$a_5 = 0.32$
$r_1 = -3.2$	$r_2 = 2.4$	$r_3 = -0.8$	$r_4 = -1.2$	$r_5 = 1.7$
$r_1 = v^T\ x_1$	$r_2 = v^T\ x_2$	$r_3 = v^T\ x_3$	$r_4 = v^T\ x_4$	$r_5 = v^T\ x_5$
5.3, 8.5, -1, 5.1	2.1, 8.7, -7, 4.2	9.7, 6.1, 9.5, 9.9	-2, 5.2, 8.5, 6.8	7.3, 2.6, 3.1, 8.3
I	like	natural	language	processing


Positive / Negative
y
6.5 2.2 -3. 9.6
$x_1a_1 + x_2a_2 + x_3a_3 + x_4a_4 + x_5a_5$
Weighted sum
$a_1 = 0$
$a_2 = 0.64$
$a_3 = 0.02$
$a_4 = 0.02$
$a_5 = 0.32$
5.3 8.5 -1 5.1
2.1 8.7 -7 4.2
9.7 6.1 9.5 9.9
-2 5.2 8.5 6.8
7.3 2.6 3.1 8.3
I
like
natural
language
processing

Attention vs Weights from fully-connected layer?

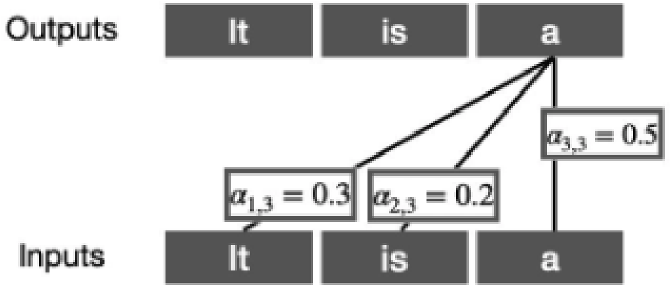
- Fully-connected layer weights W are static w.r.t the input

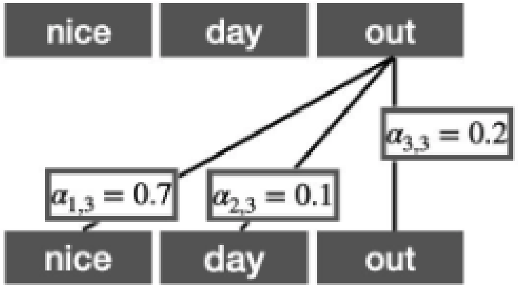




W

- Attention scores a are dynamic w.r.t the input context

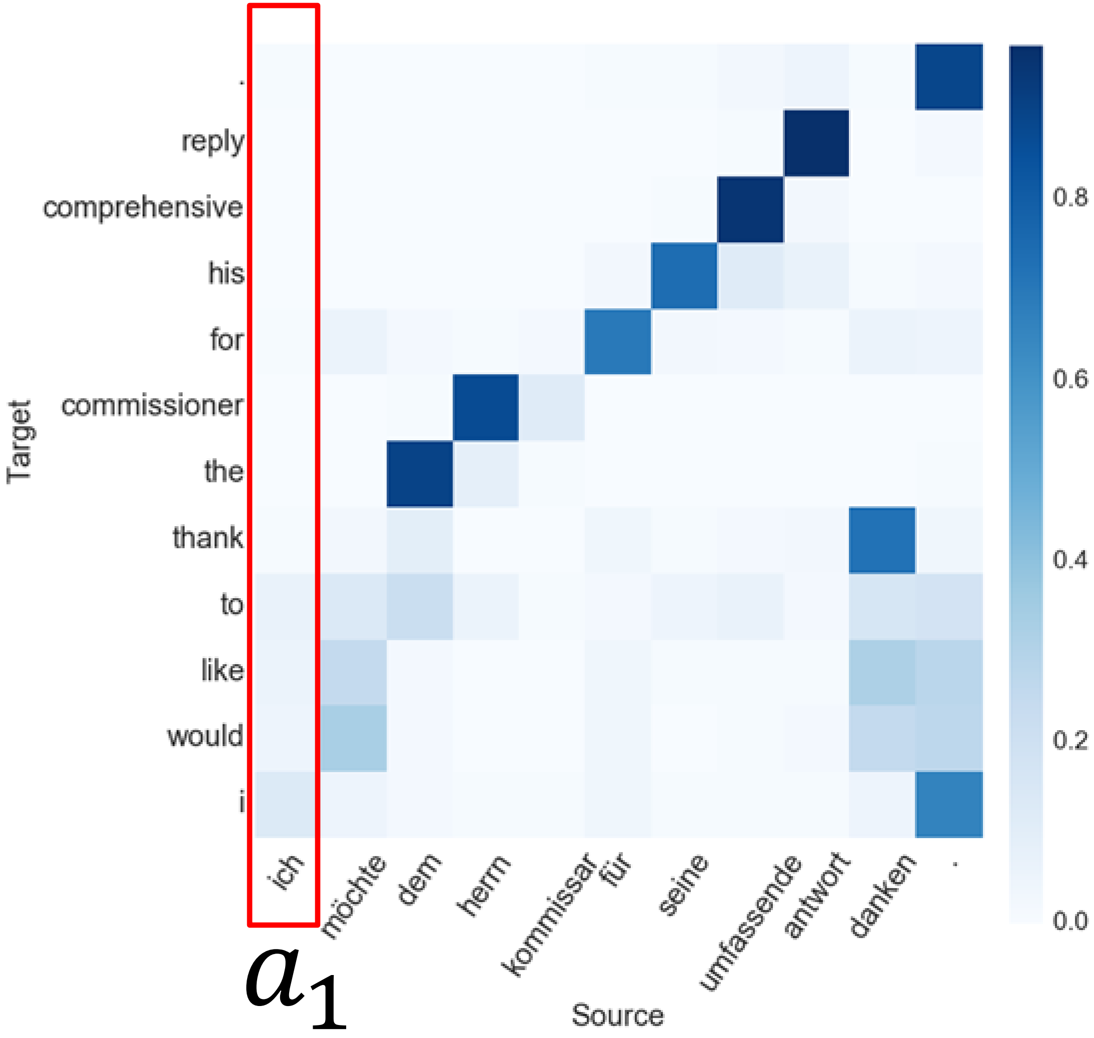




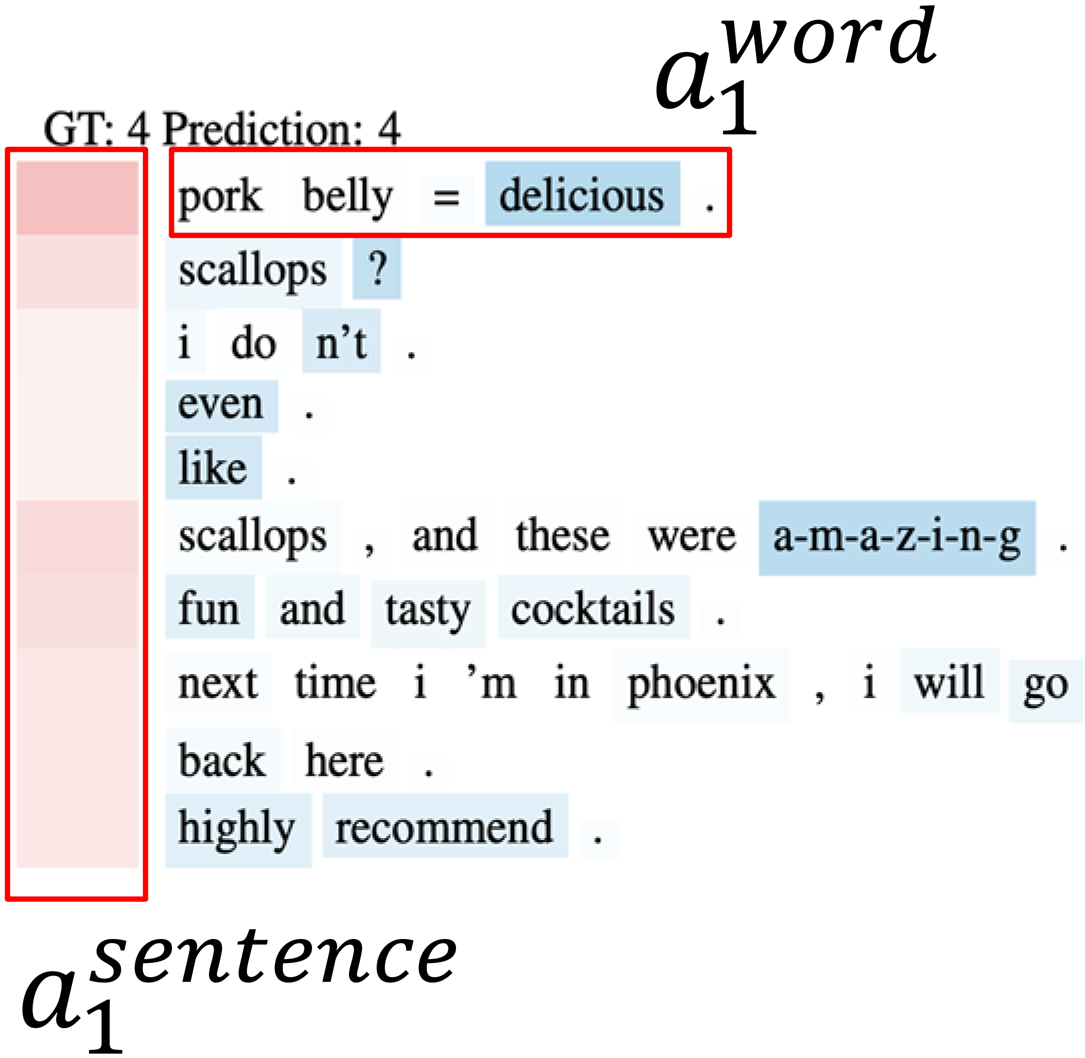
a

Examples from Sebastian Raschka

Attention



Neural Machine Translation by Jointly Learning to Align and Translate



Hierarchical Attention Networks for Document Classification

Attention

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

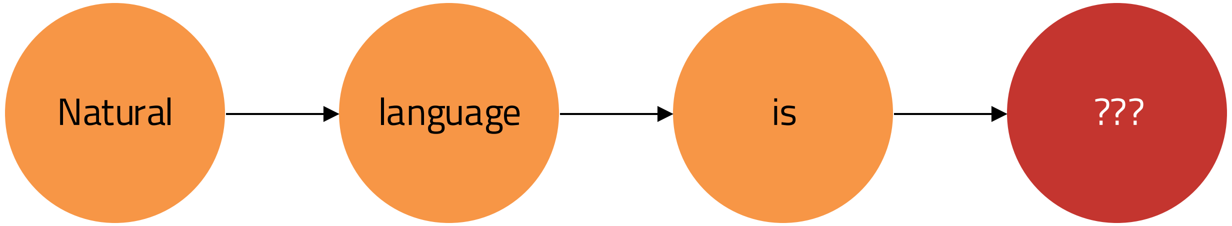
BERT

- Transformer or self-attention based (Vaswani et al., 2017) masked language model using bidirectional context and next sentence prediction

- Generates multiple layers of representations for each token sensitive to its context use.

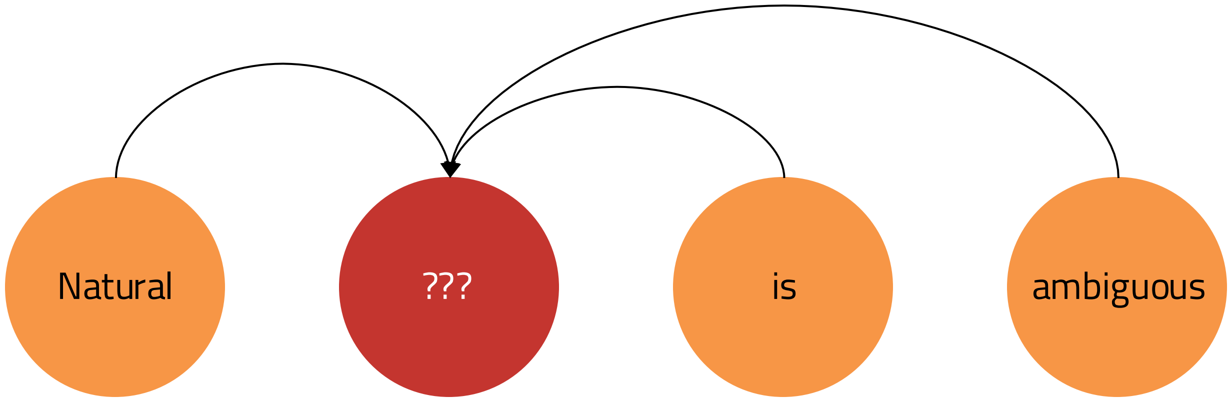
Classical (causal) language model

Consider only the left context to predict the next word
(i.e., the final word in a sequence is masked)



Masked language model

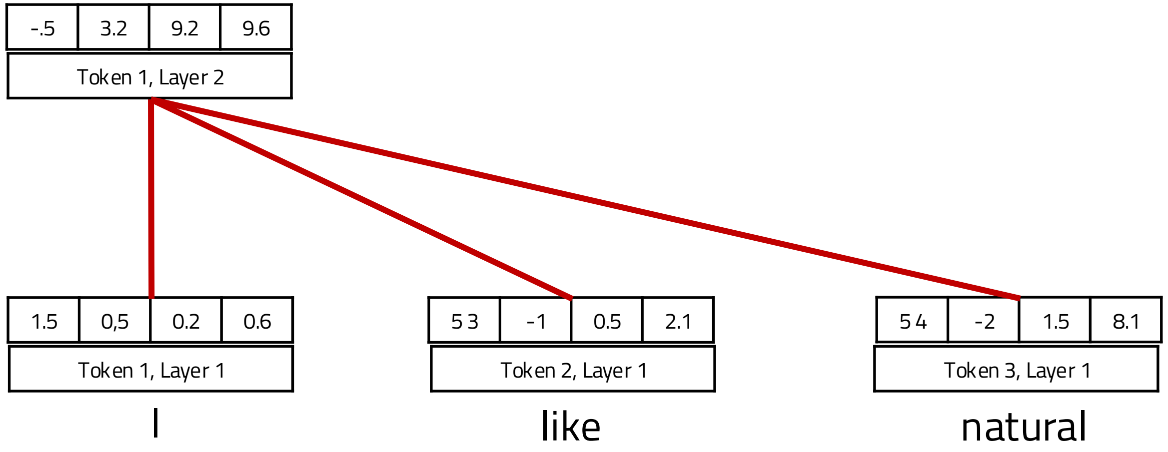
Use any context (left or right) to predict a masked word



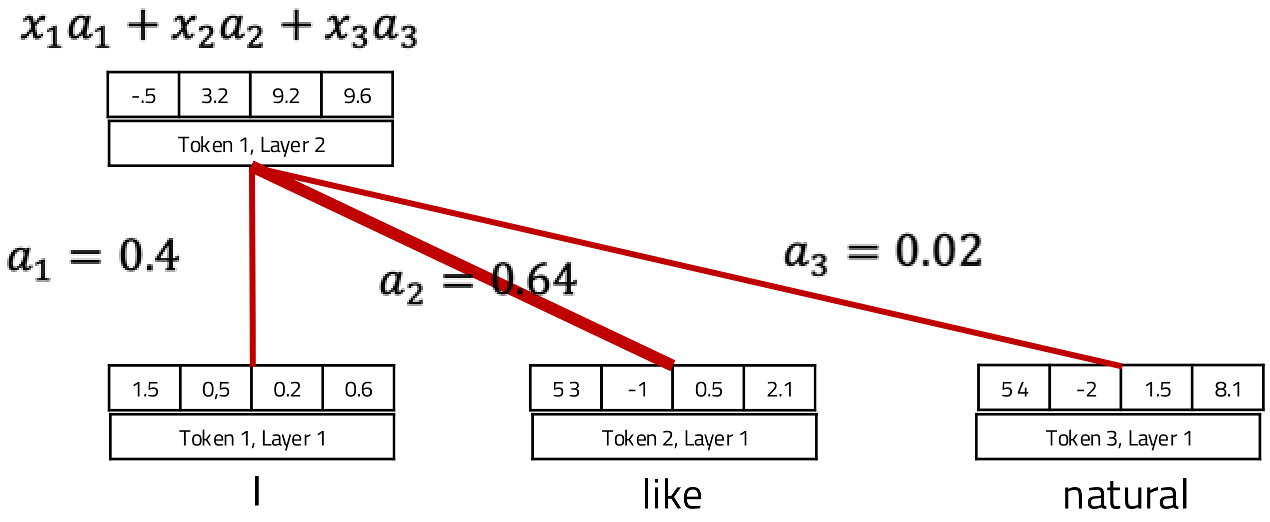
Each token in input starts represented by
token and **position** embeddings

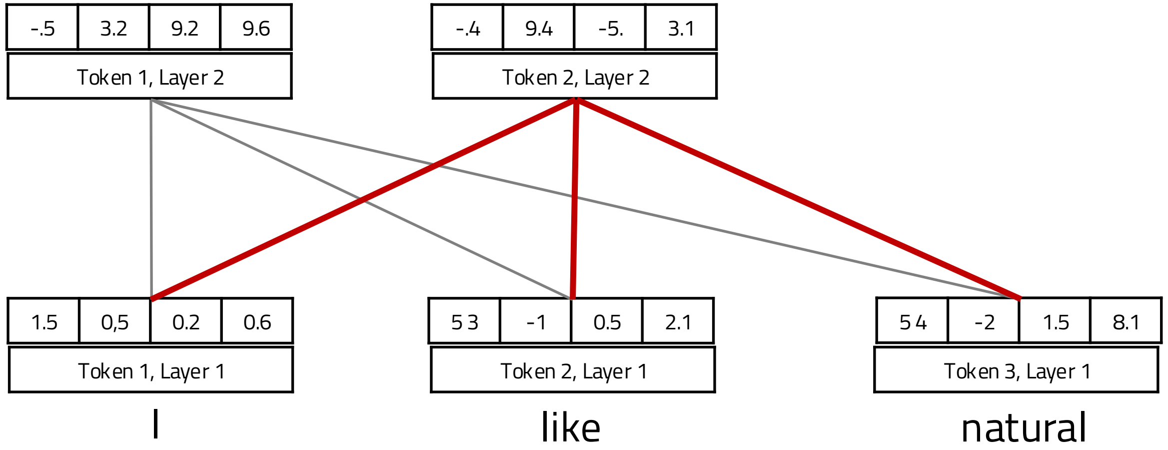
1.5 | 0,5 | 0.2 | 0.6	5 3 | -1 | 0.5 | 2.1	5 4 | -2 | 1.5 | 8.1
Token 1, Layer 1	Token 2, Layer 1	Token 3, Layer 1
I	like	natural

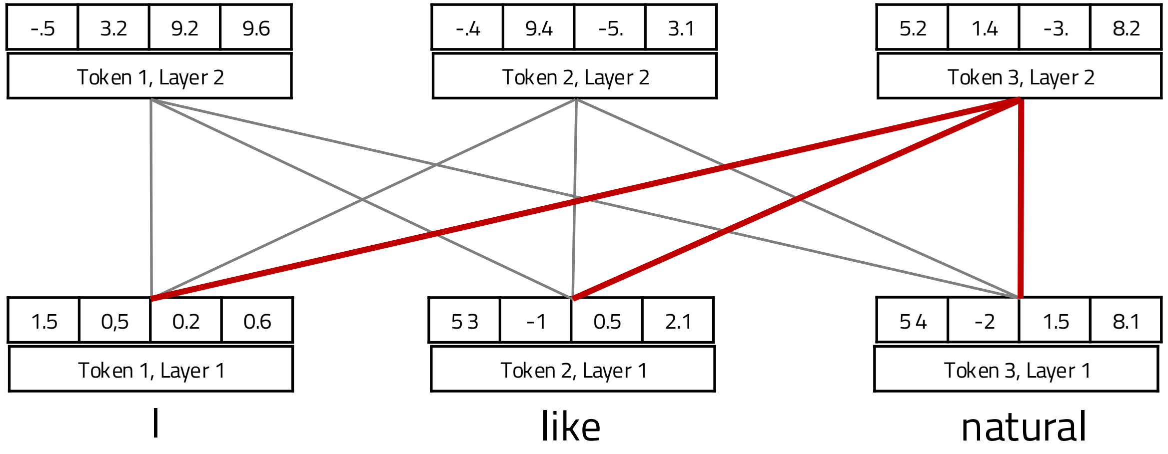
The value for time step j at layer i is the result of attention over all time steps in the previous layer i-1

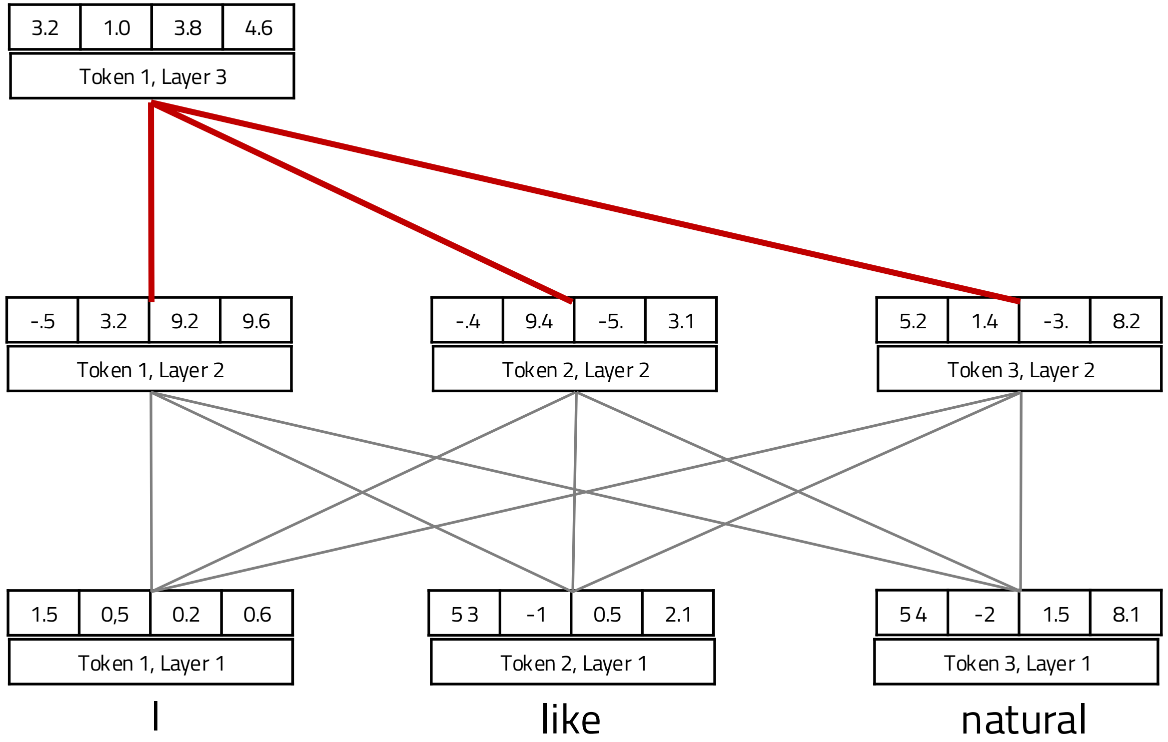


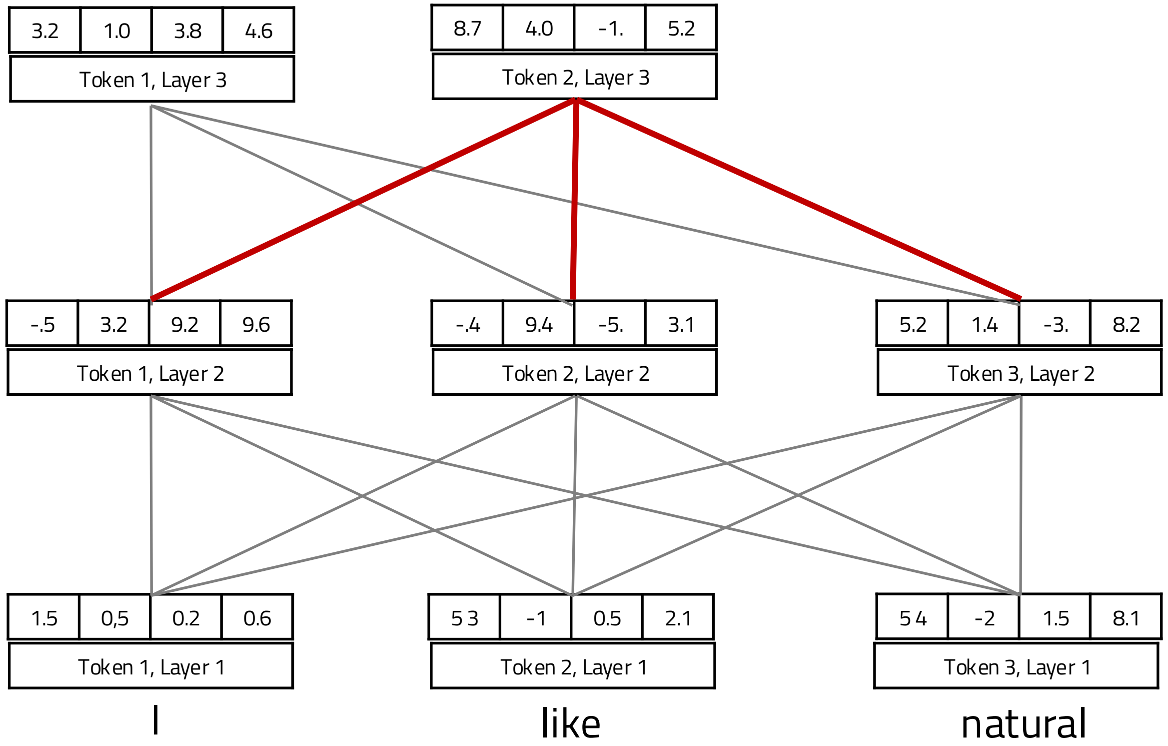
The value for time step j at layer i is the result of attention over all time steps in the previous layer i-1

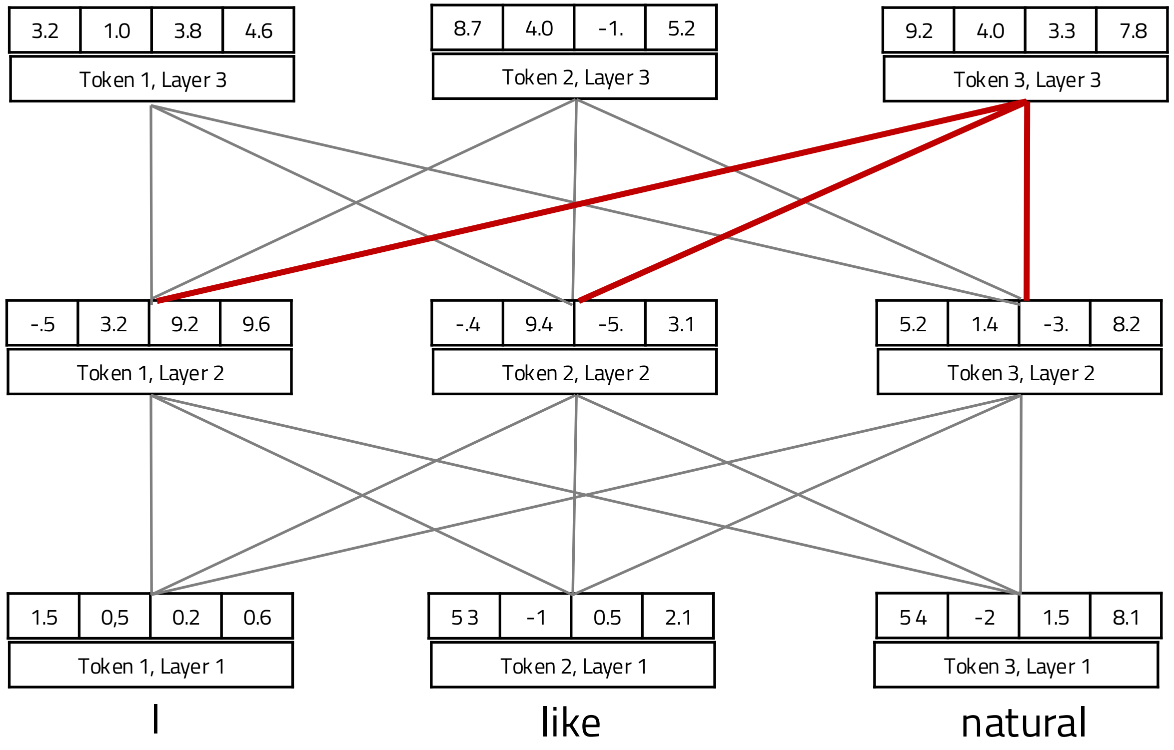



-.5
3.2
9.2
9.6
Token 1, Layer 2
-.4
9.4
-5.
3.1
Token 2, Layer 2
1.5
0,5
0.2
0.6
Token 1, Layer 1
5 3
-1
0.5
2.1
Token 2, Layer 1
5 4
-2
1.5
8.1
Token 3, Layer 1
I
like
natural

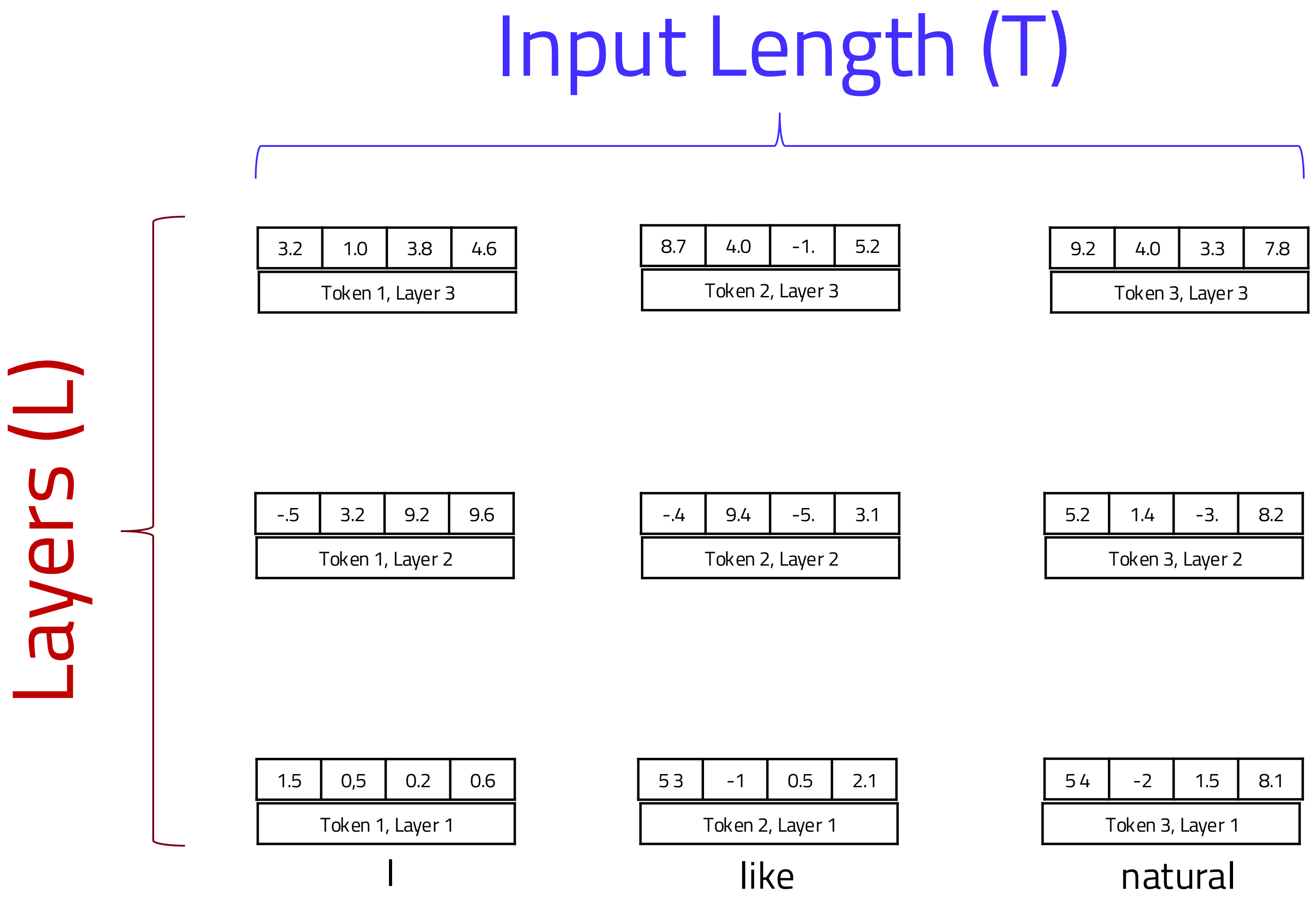

-.5
3.2
9.2
9.6
Token 1, Layer 2
-.4
9.4
-5.
3.1
Token 2, Layer 2
5.2
1.4
-3.
8.2
Token 3, Layer 2
1.5
0,5
0.2
0.6
Token 1, Layer 1
5 3
-1
0.5
2.1
Token 2, Layer 1
5 4
-2
1.5
8.1
Token 3, Layer 1
I
like
natural


3.2
1.0
3.8
4.6
Token 1, Layer 3
-.5
3.2
9.2
9.6
Token 1, Layer 2
-.4
9.4
-5.
3.1
Token 2, Layer 2
5.2
1.4
-3.
8.2
Token 3, Layer 2
1.5
0,5
0.2
0.6
Token 1, Layer 1
5 3
-1
0.5
2.1
Token 2, Layer 1
5 4
-2
1.5
8.1
Token 3, Layer 1
I
like
natural


3.2 1.0 3.8 4.6
Token 1, Layer 3
8.7 4.0 -1. 5.2
Token 2, Layer 3
-.5 3.2 9.2 9.6
Token 1, Layer 2
-.4 9.4 -5. 3.1
Token 2, Layer 2
5.2 1.4 -3. 8.2
Token 3, Layer 2
1.5 0,5 0.2 0.6
Token 1, Layer 1
5 3 -1 0.5 2.1
Token 2, Layer 1
5 4 -2 1.5 8.1
Token 3, Layer 1
I
like
natural


3.2 | 1.0 | 3.8 | 4.6
Token 1, Layer 3
8.7 | 4.0 | -1. | 5.2
Token 2, Layer 3
9.2 | 4.0 | 3.3 | 7.8
Token 3, Layer 3
-.5 | 3.2 | 9.2 | 9.6
Token 1, Layer 2
-.4 | 9.4 | -5. | 3.1
Token 2, Layer 2
5.2 | 1.4 | -3. | 8.2
Token 3, Layer 2
1.5 | 0,5 | 0.2 | 0.6
Token 1, Layer 1
5 3 | -1 | 0.5 | 2.1
Token 2, Layer 1
5 4 | -2 | 1.5 | 8.1
Token 3, Layer 1
I
like
natural

At the end, we have one representation
for each layer for each token



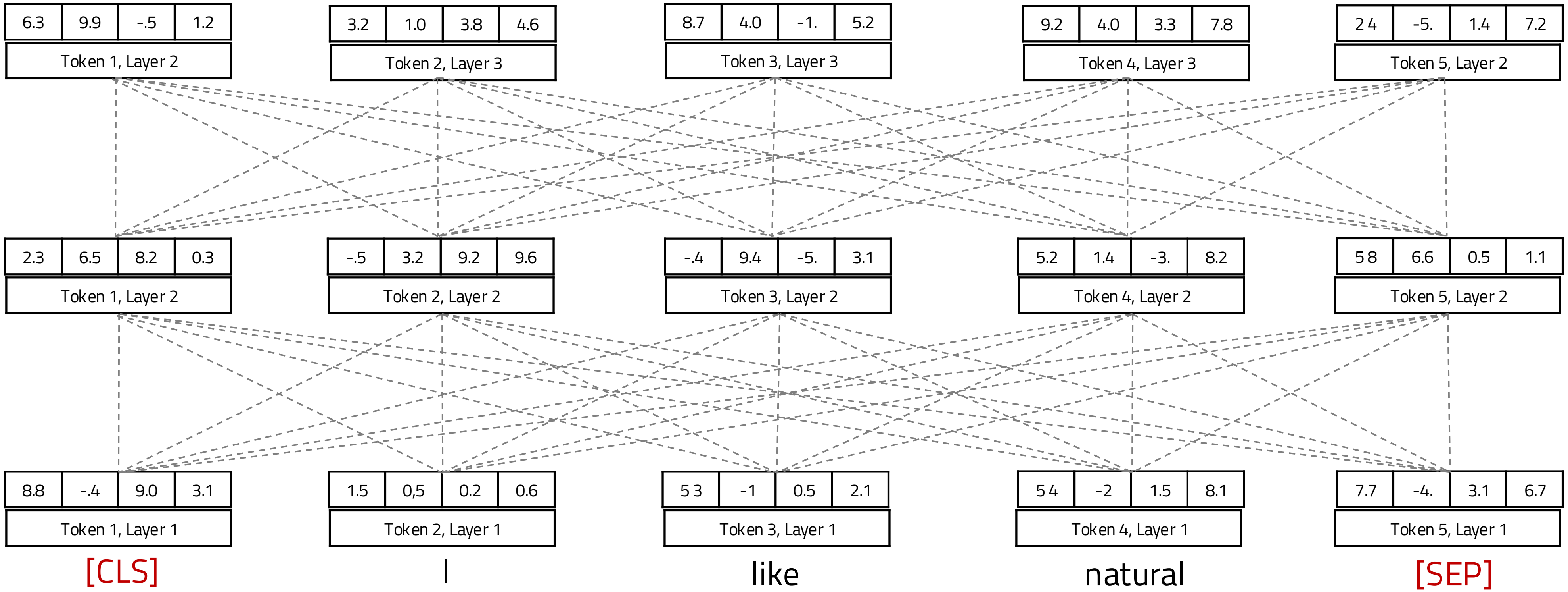
Tokenization in BERT

- BERT uses WordPiece tokenization, which segments some morphological structure of tokens

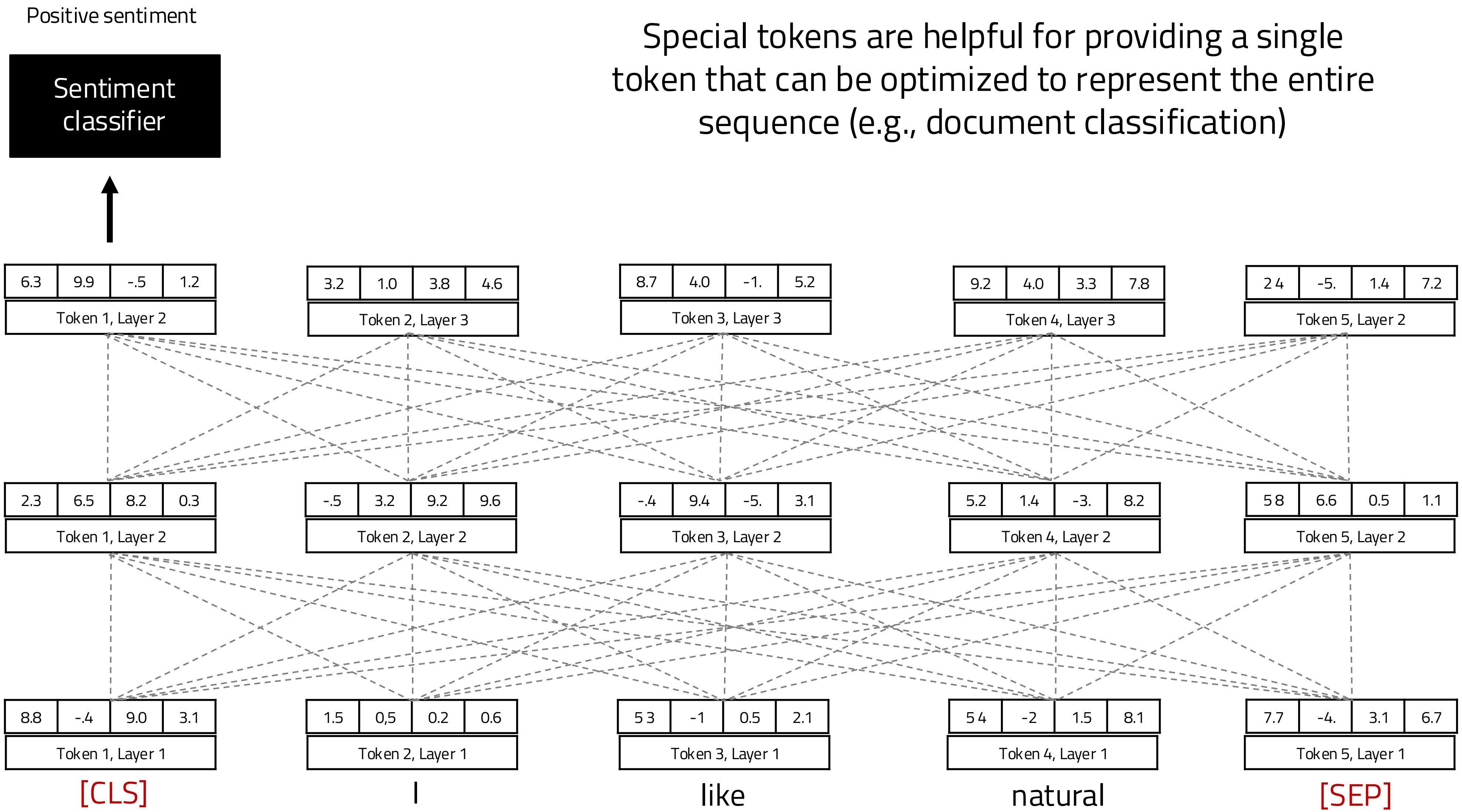
The	The
unwilling	un #will #ing
barked	bark #ed

- Vocabulary size: 30,000

- BERT encodes each sentence by appending a special token to the beginning ([CLS]) and end ([SEP]) of each sequence

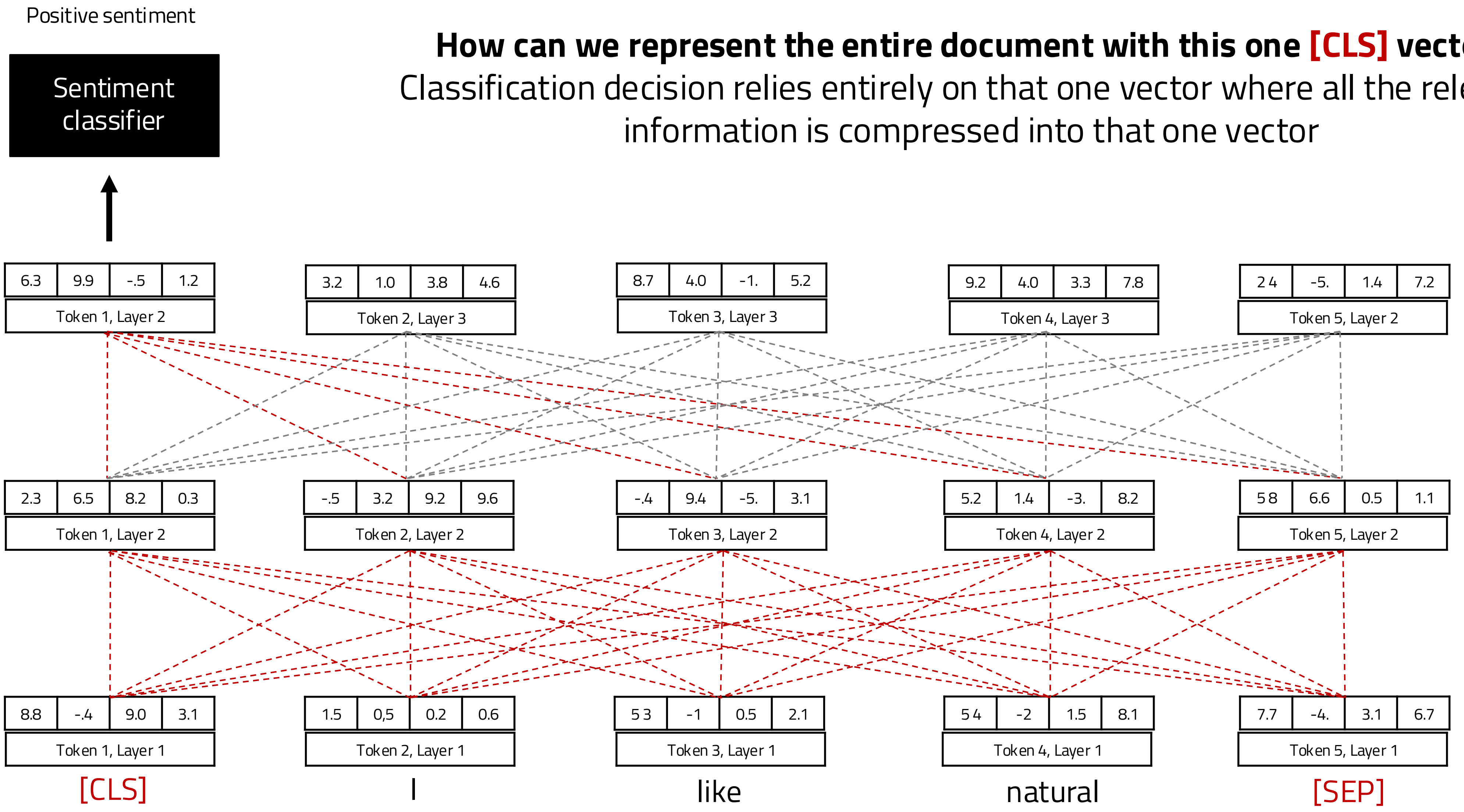

6.3 9.9 -.5 1.2
Token 1, Layer 2
3.2 1.0 3.8 4.6
Token 2, Layer 3
8.7 4.0 -1. 5.2
Token 3, Layer 3
9.2 4.0 3.3 7.8
Token 4, Layer 3
2 4 -5. 1.4 7.2
Token 5, Layer 2
2.3 6.5 8.2 0.3
Token 1, Layer 2
-.5 3.2 9.2 9.6
Token 2, Layer 2
-.4 9.4 -5. 3.1
Token 3, Layer 2
5.2 1.4 -3. 8.2
Token 4, Layer 2
5 8 6.6 0.5 1.1
Token 5, Layer 2
8.8 -.4 9.0 3.1
Token 1, Layer 1
1.5 0,5 0.2 0.6
Token 2, Layer 1
5 3 -1 0.5 2.1
Token 3, Layer 1
5 4 -2 1.5 8.1
Token 4, Layer 1
7.7 -4. 3.1 6.7
Token 5, Layer 1
[CLS]
I
like
natural
[SEP]

Special tokens are helpful for providing a single token that can be optimized to represent the entire sequence (e.g., document classification)



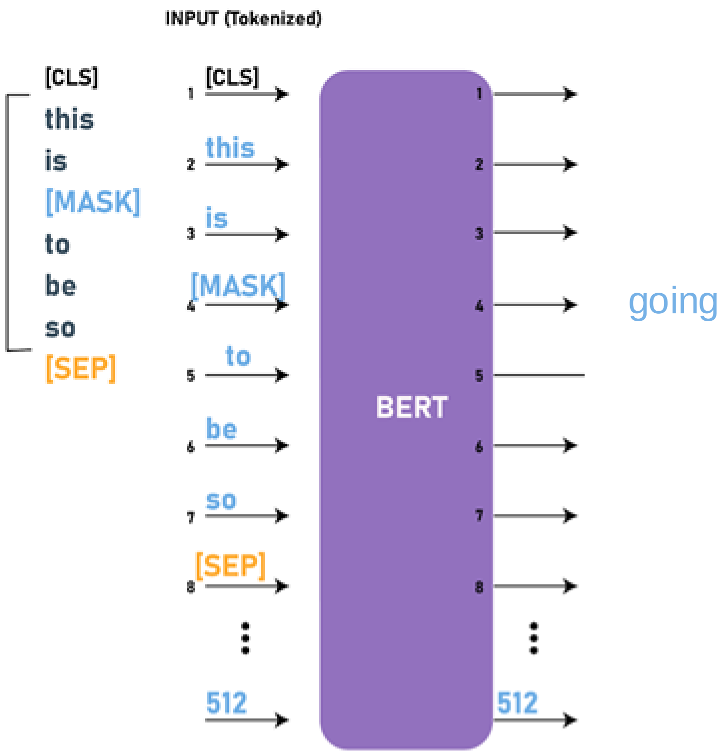
How can we represent the entire document with this one [CLS] vector?

Classification decision relies entirely on that one vector where all the relevant information is compressed into that one vector



Training BERT

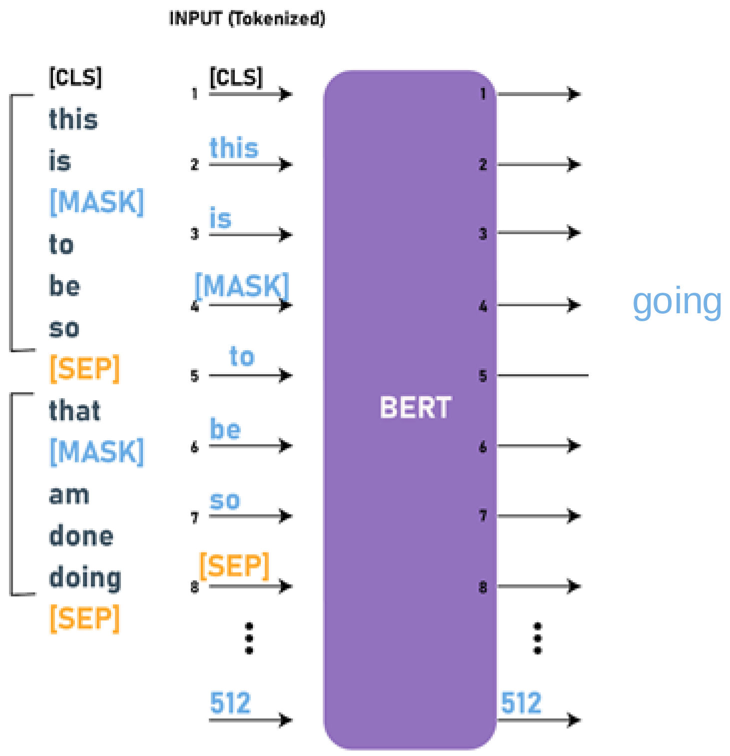
- #1: Masked language modeling
  - [Mask] one word from input and try to predict that word as output
  - Maximum length = 512



Input Length (T)= 512

Training BERT

- #1: Masked language modeling
  - [Mask] one word from input and try to predict that word as output
  - Maximum length = 512
  - Concatenate two sentences with [SEP] token
  - More powerful than Bidirectional-RNN LM

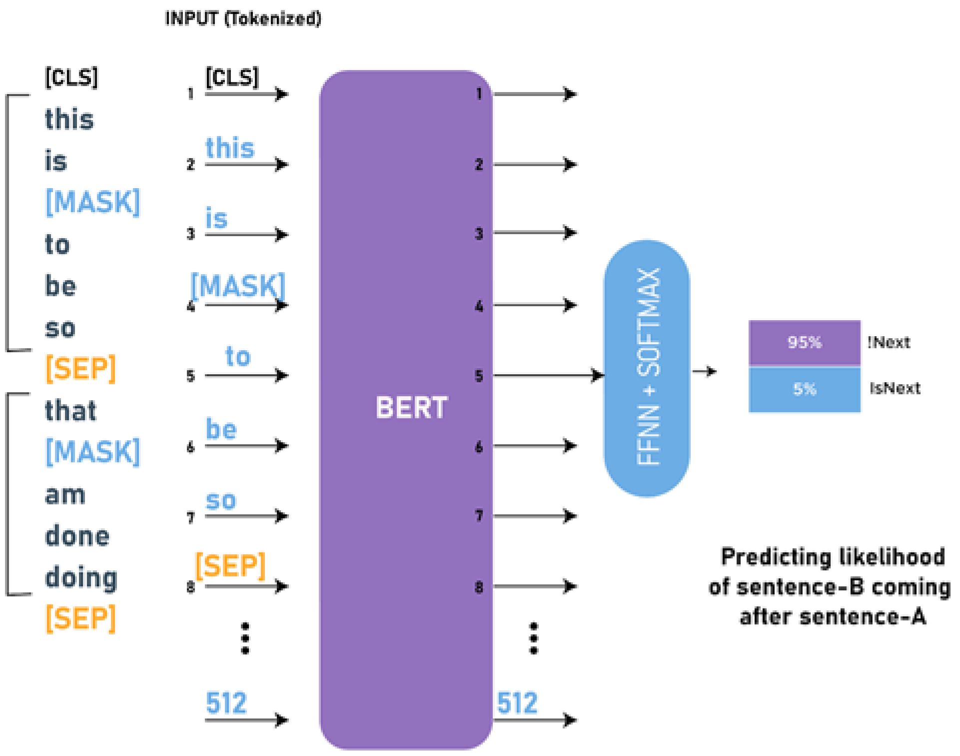


Training BERT

- #2: Next sentence prediction
  - For a pair of sentences, predict from [CLS] representation whether they appeared **sequentially** in the training data

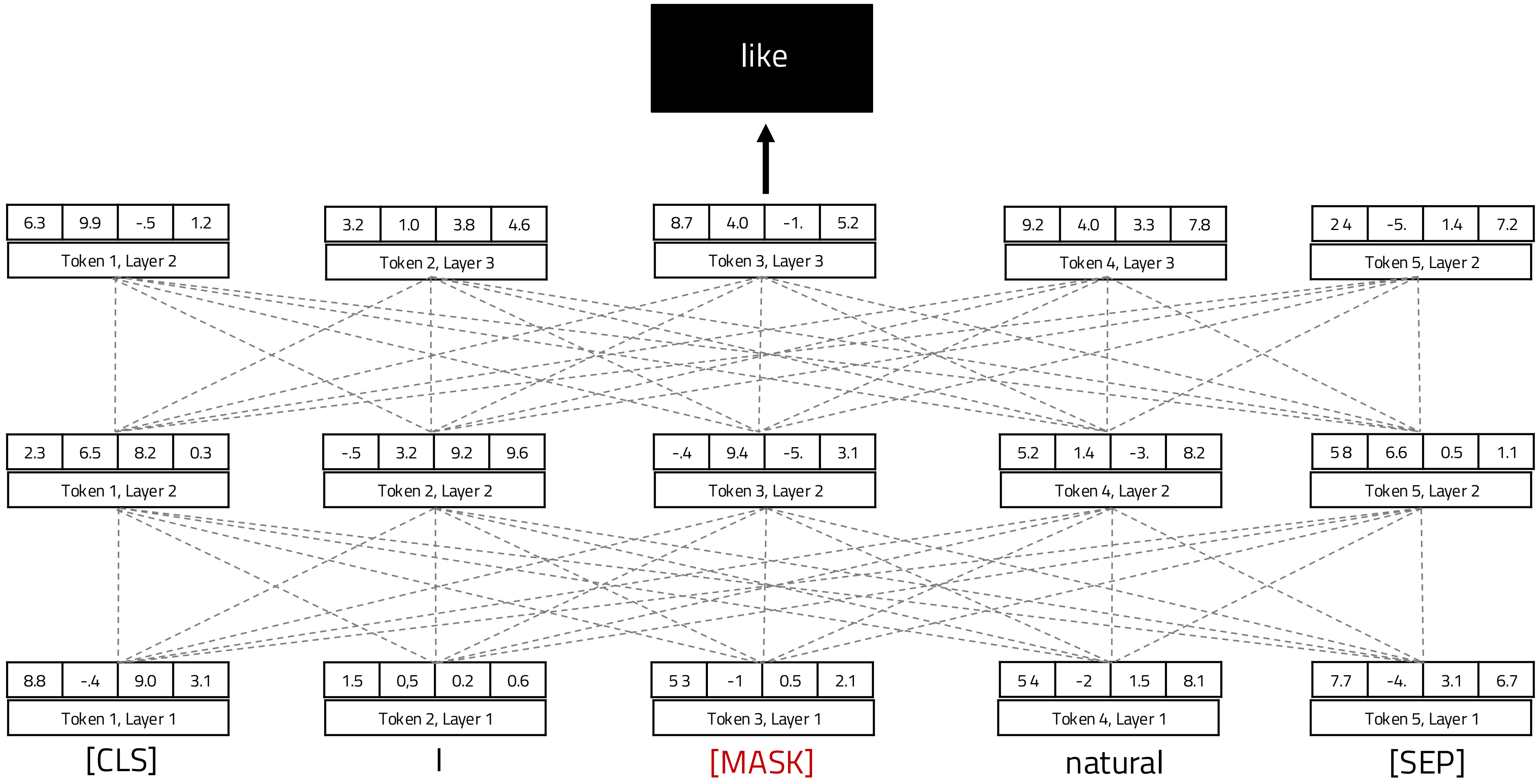
    Next=True [CLS] I like natural language processing [SEP] because NLP is fun

    Next=False [CLS] I like natural language processing [SEP] Minnesota is cold.
  - This objective turns out to be not that effective, found in RoBERTa paper (Liu et al., 2019)



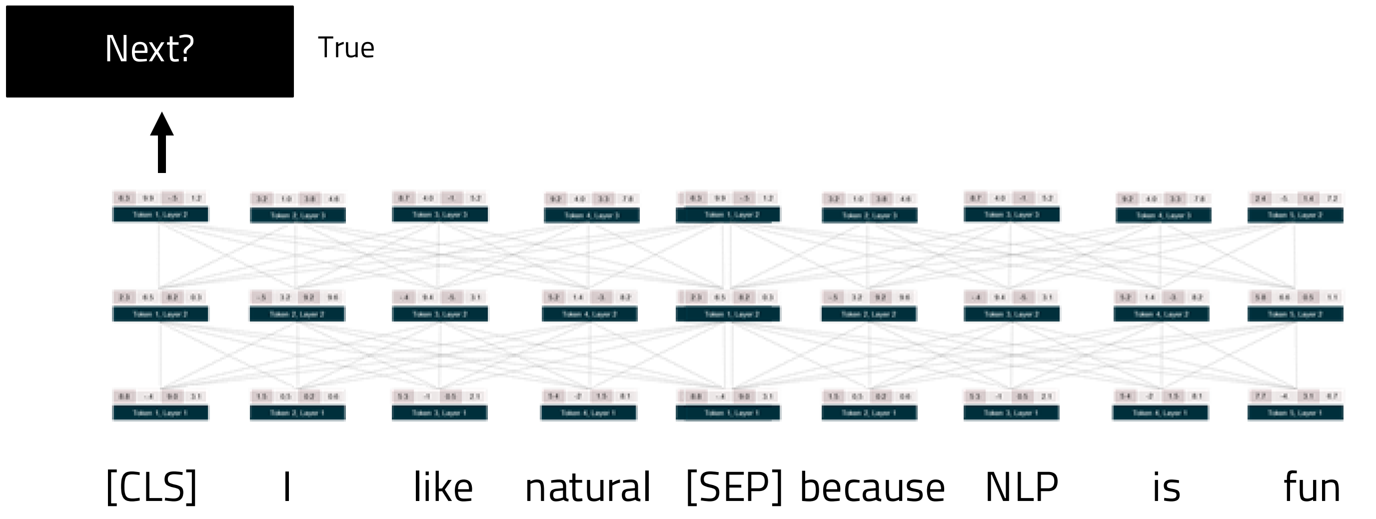
Objective #1: Masked language modeling

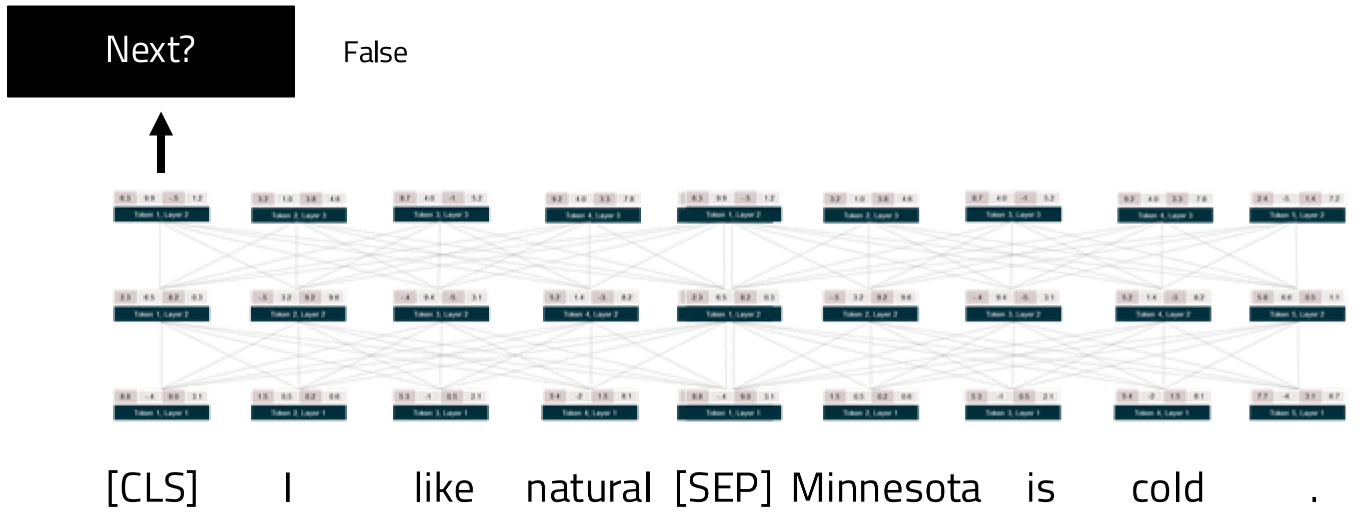
L_{MLM}



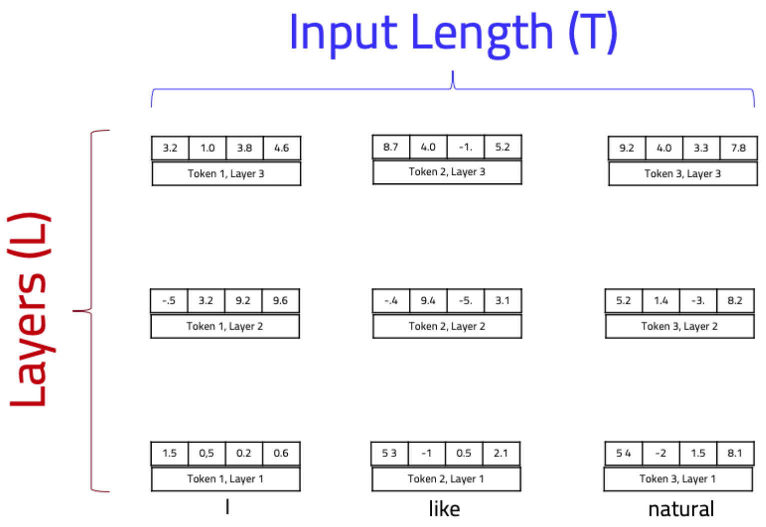
Objective #2: Next sentence prediction

$$L_{MLM} + L_{NSP}$$





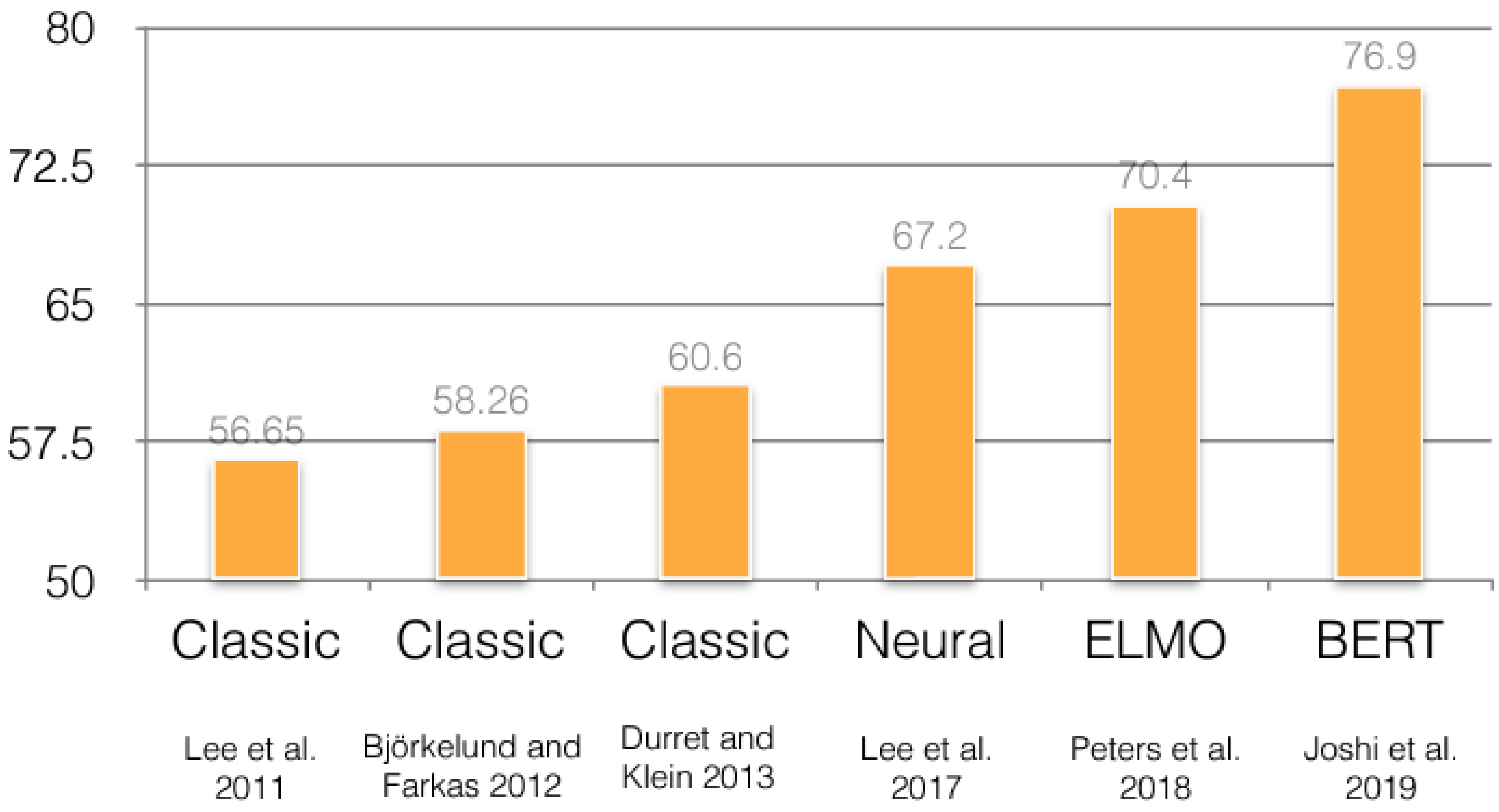
Details of BERT training



- Deep layers
  - 12 layers for BERT-base
  - 24 layers for BERT-large
- Large representation size (768 per layer)
- Pretrained on English Wikipedia (2.5B words) and BookCorpus (800M words)

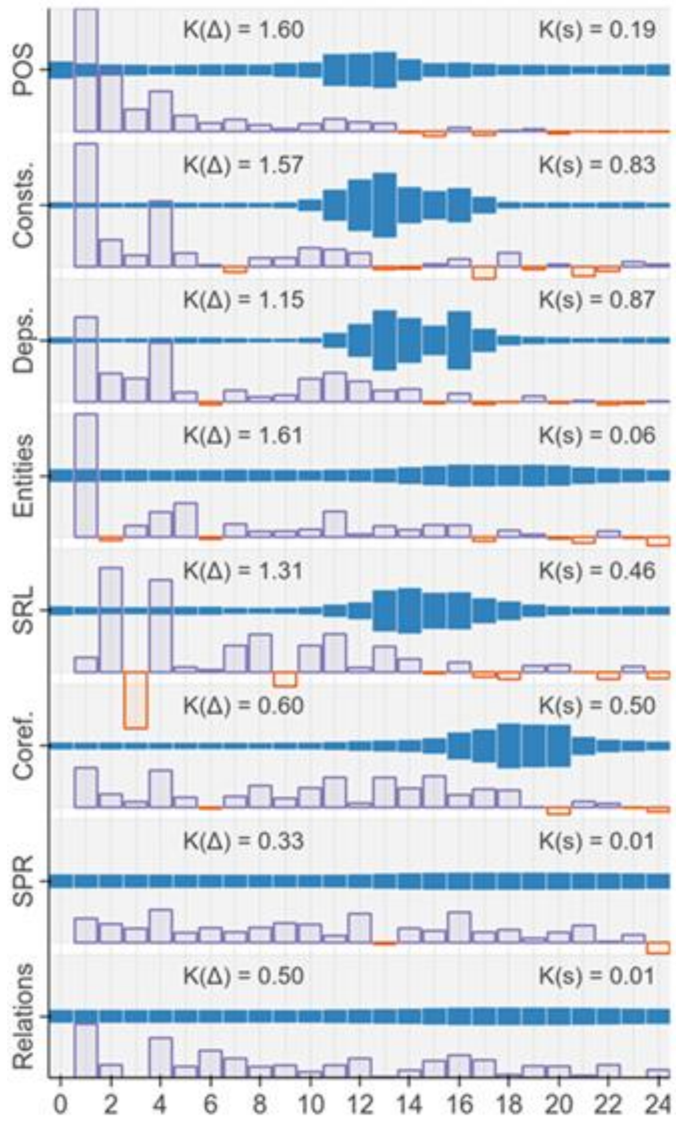
$$L_{MLM} + L_{NSP}$$

Coreference resolution with BERT

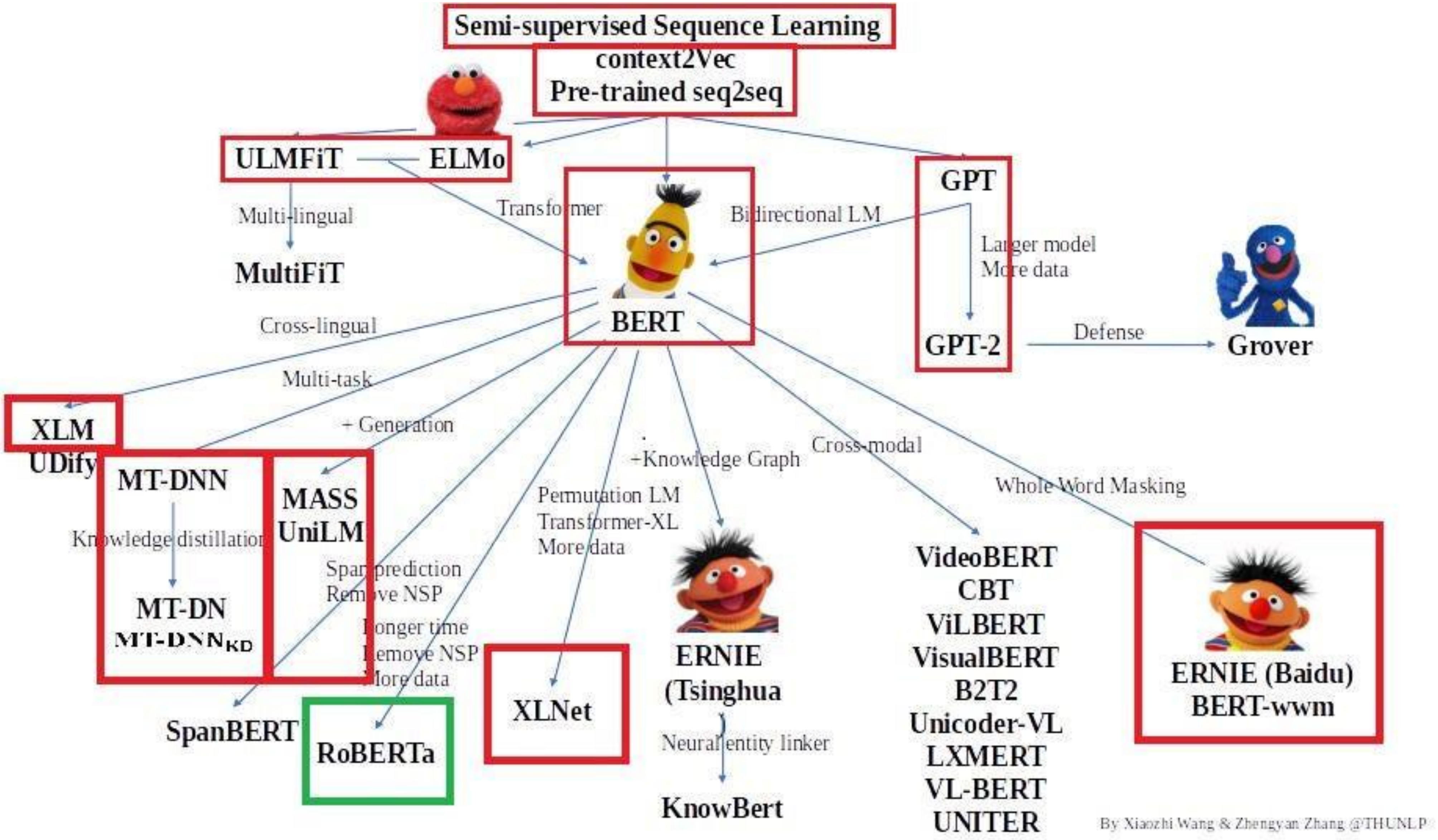


BERTology

- ❑ Hewitt et al. 2019
- ❑ Tenney et al. 2019
- ❑ McCoy et al. 2019
- ❑ Liu et al. 2019
- ❑ Clark et al. 2019
- ❑ Goldberg 2019
- ❑ Michel et al. 2019

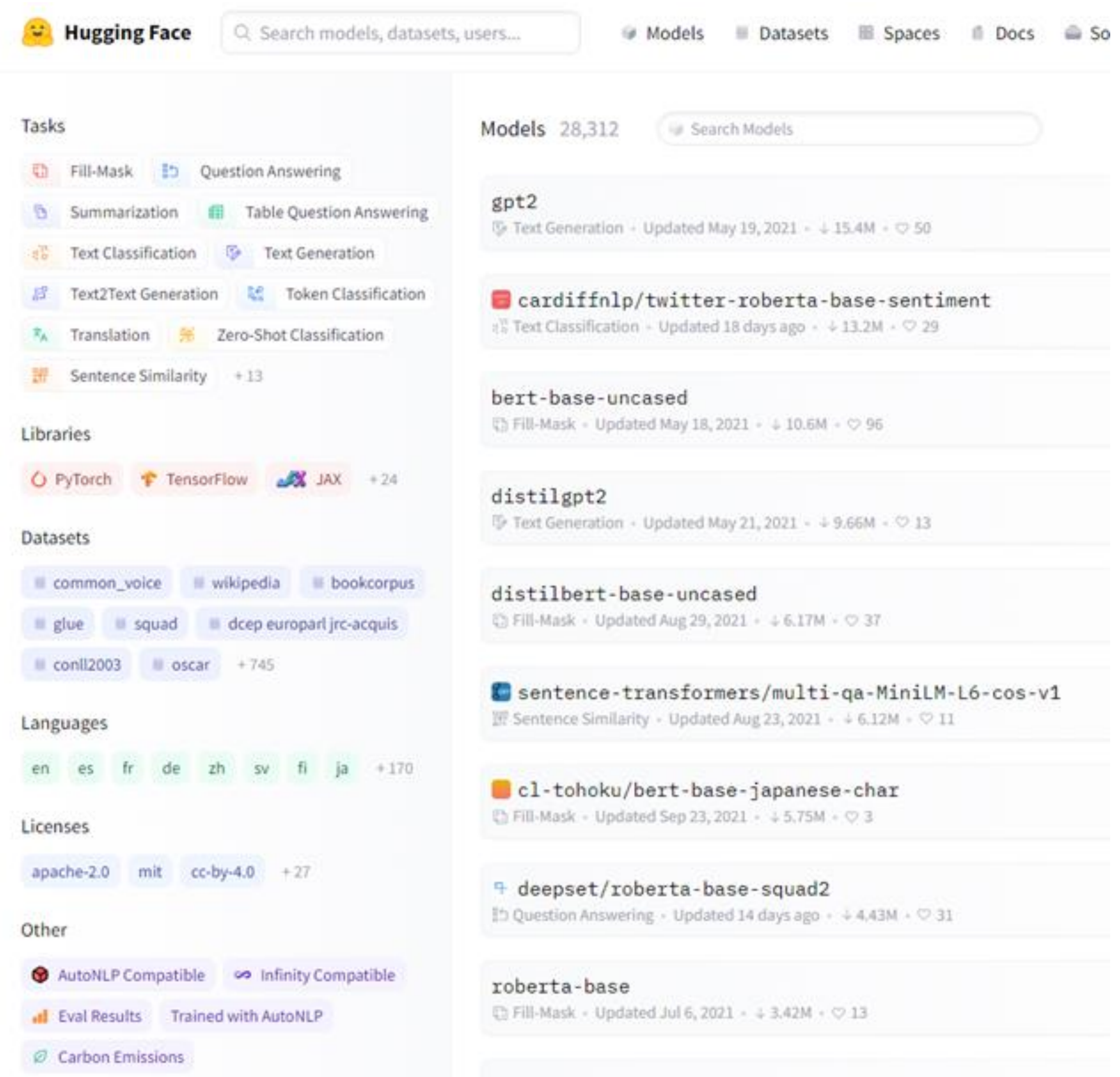


Tenney et al. (2019), "BERT Rediscovers the Classical NLP Pipeline"


Semi-supervised Sequence Learning
context2Vec
Pre-trained seq2seq
ULMFiT
ELMo
Multi-lingual
MultiFiT
Transformer
BERT
Bidirectional LM
GPT
Larger model
More data
GPT-2
Defense
Grover
Cross-lingual
Multi-task
XLM
UDify
+ Generation
MT-DNN
Knowledge distillation
MT-DNN$_{KD}$
MASS
UniLM
Span prediction
Remove NSP
Longer time
Remove NSP
More data
SpanBERT
RoBERTa
Permutation LM
Transformer-XL
More data
XLNet
+Knowledge Graph
ERNIE
(Tsinghua)
Neural entity linker
KnowBert
Cross-modal
VideoBERT
CBT
ViLBERT
VisualBERT
B2T2
Unicoder-VL
LXMERT
VL-BERT
UNITER
Whole Word Masking
ERNIE (Baidu)
BERT-wwm
By Xiaozhi Wang & Zhengyan Zhang @THUNLP

Other pretrained LMs

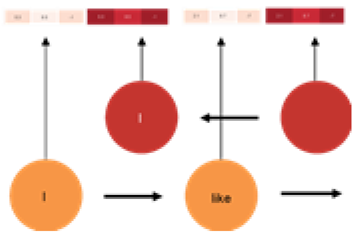
- BERT
- XLNet
- ALBERT
- RoBERTa
- DistilBERT
- GPT-2/3
- Multilingual-BERT

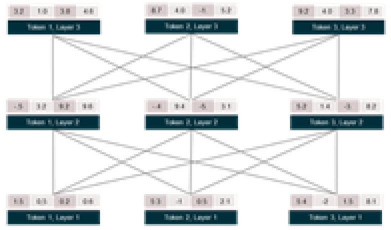


https://huggingface.co/models

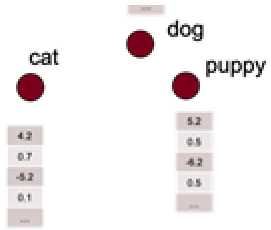
Summary

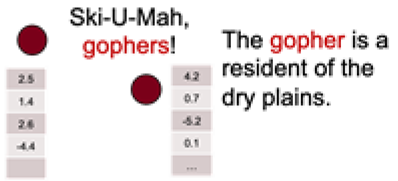
woman	king
5.2	1.5
0.5	0.4
-6.2	0.6





- Word embeddings can be substituted for one-hot encodings in many models (MLP, CNN, RNN, logistic regression).
- Bidirectional modeling in ELMo/BERT helps learn more context sensitive information.
- Attention gives us a mechanism to learn which parts of a sequence to pay attention to more in forming a representation of it.
- Static word embeddings (word2vec, Glove) provide representations of word types; contextualized word representations (ELMo, BERT) provide representations of tokens in context.





Questions

- Any caveats in pre-training and fine-tuning framework?
- Other types of self-supervision objective from unlabeled text, rather than next/masked token prediction?
- Better representation model than self-attention (previously, bi-directional RNN)?
- Scaling up the pre-training guarantees performance gain (scaling law)? Then, NLP will be solved simply by scaling?