Applied Natural Language Processing

Info 256

Lecture 14: Contextual word embeddings (Oct 11, 2023)

David Bamman, UC Berkeley

Lexical semantics

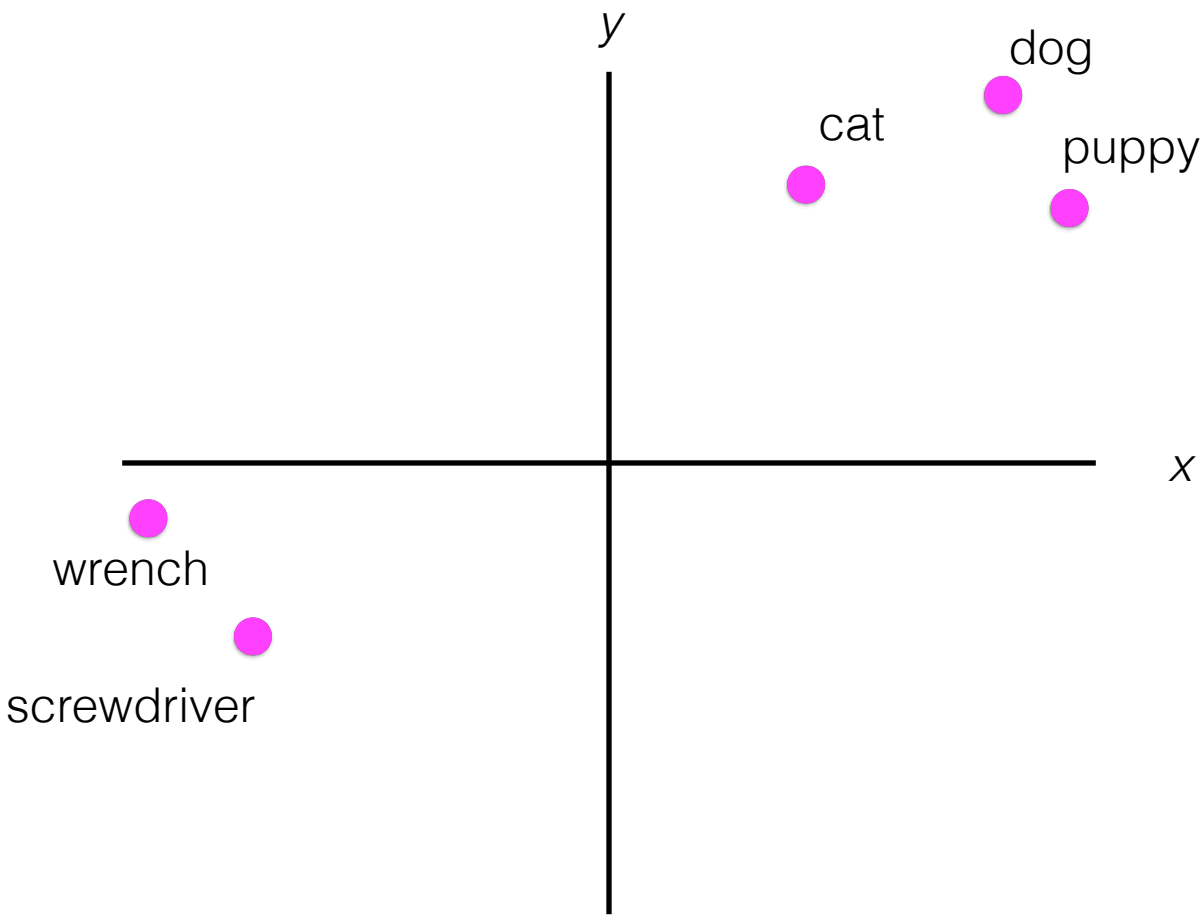
“You shall know a word by the company it keeps”

[Firth 1957]

Zellig Harris, “Distributional Structure” (1954)

Ludwig Wittgenstein, Philosophical Investigations (1953)

everyone likes	_____	
a bottle of	_____	is on the table
	_____	makes you drunk
a cocktail with	_____	and seltzer


y
dog
cat
puppy
x
wrench
screwdriver

Contextualized embeddings

- Models for learning static embeddings like word2vec and Glove learn a single representation for a word type.

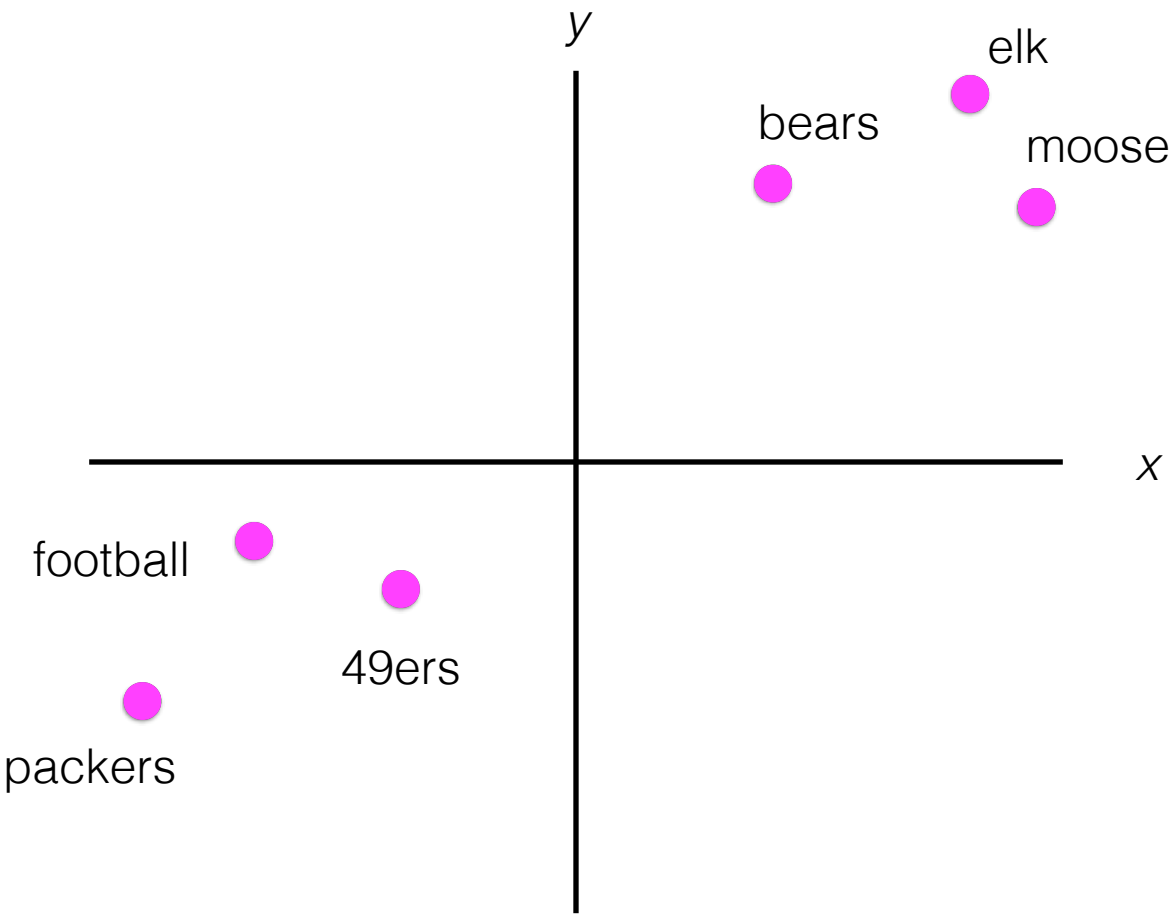
Types and tokens

- Type: bears

- Tokens:
    - The bears ate the honey
    - We spotted the bears from the highway
    - Yosemite has brown bears
    - The chicago bears didn't make the playoffs

"bears"

3.1	1.4	-2.7	0.3
3.1	1.4	-2.7	0.3
3.1	1.4	-2.7	0.3
3.1	1.4	-2.7	0.3

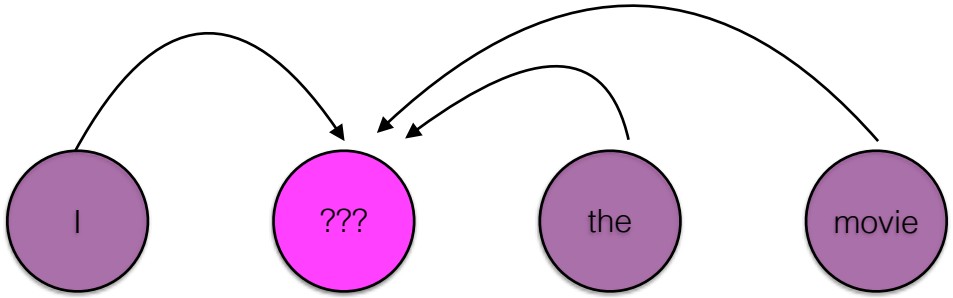

y
elk
bears
moose
x
football
49ers
packers

Contextualized word representations

- Big idea: 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 and trainable to be optimized for a specific NLP task.

Masked language model

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



$$P\left(w_t \mid w_{\neg t}\right)$$

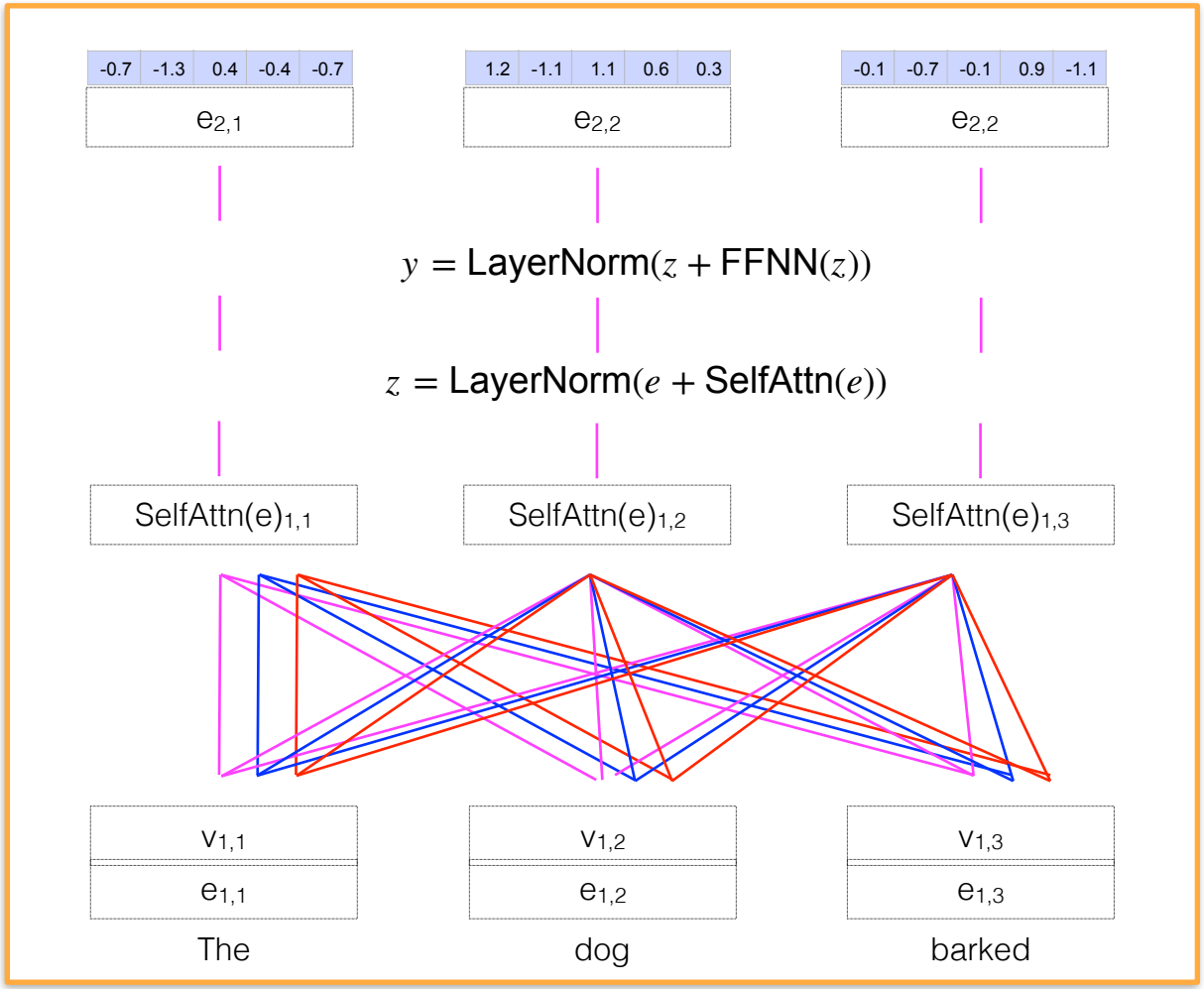
BERT

- Transformer-based model (Vaswani et al. 2017) to predict masked word using bidirectional context + next sentence prediction.
- Generates multiple layers of representations for each token sensitive to its context of use.

Output

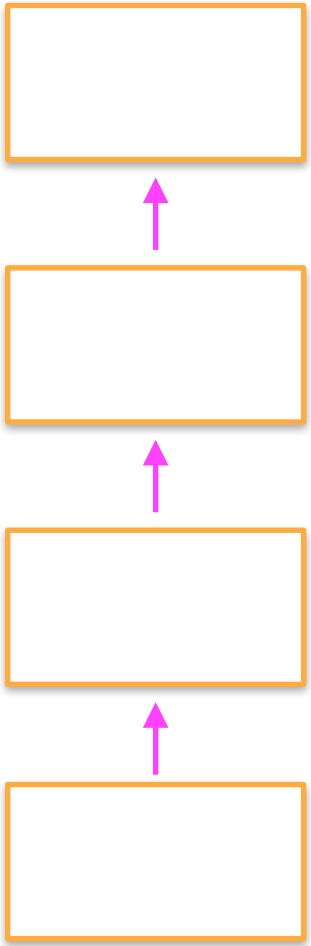
This whole process defines one attention block. The input is a sequence of (e.g. 100-dimensional) vectors; the output of each block is a sequence of (100-dimensional) vectors.

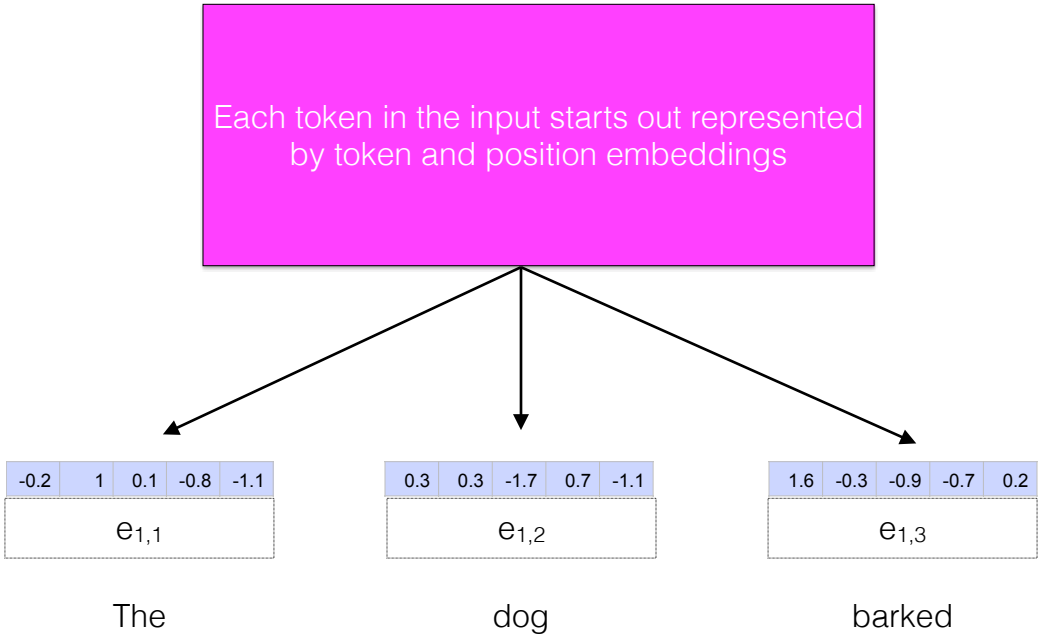
Input

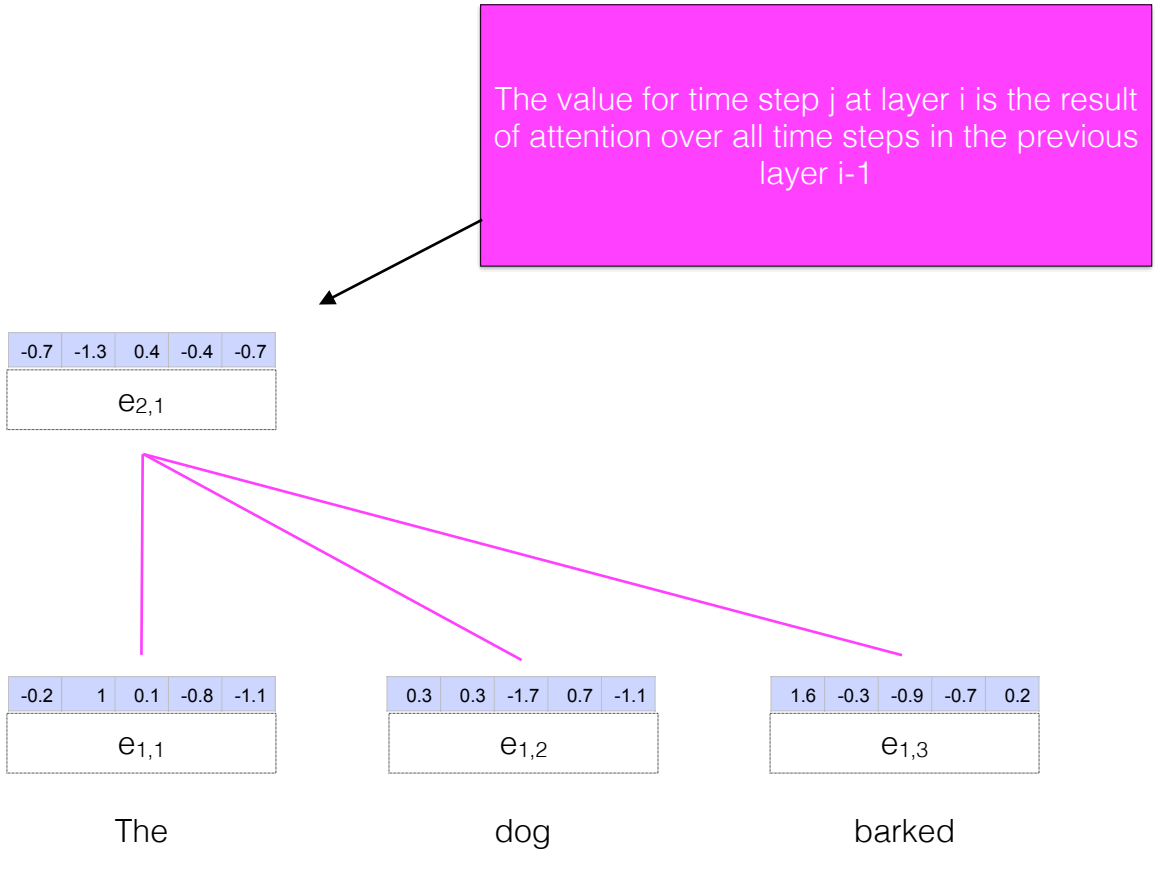


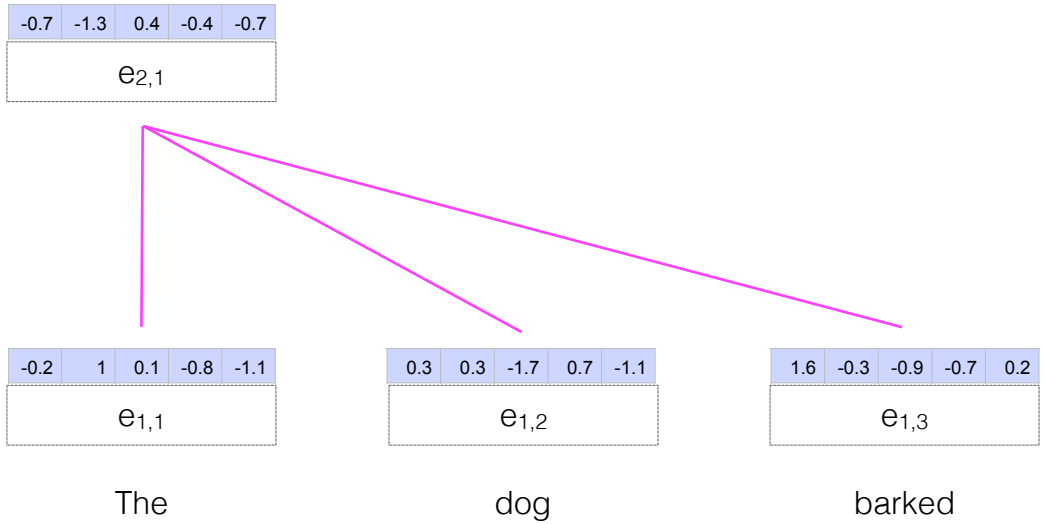
This whole process defines one attention block. The input is a sequence of (e.g. 100-dimensional) vectors; the output of each block is a sequence of (100-dimensional) vectors.

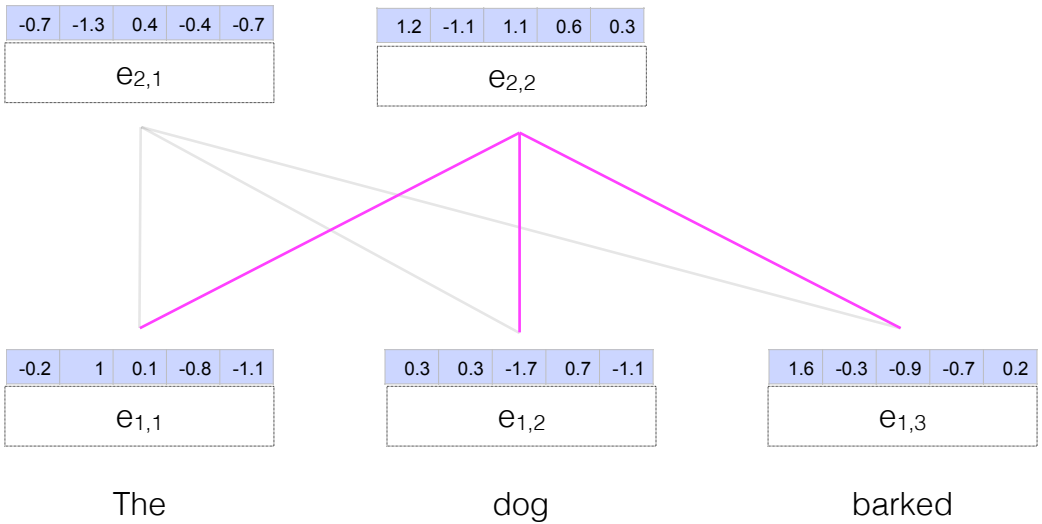
Transformers can stack many such blocks; where the output from block b is the input to block b+1.

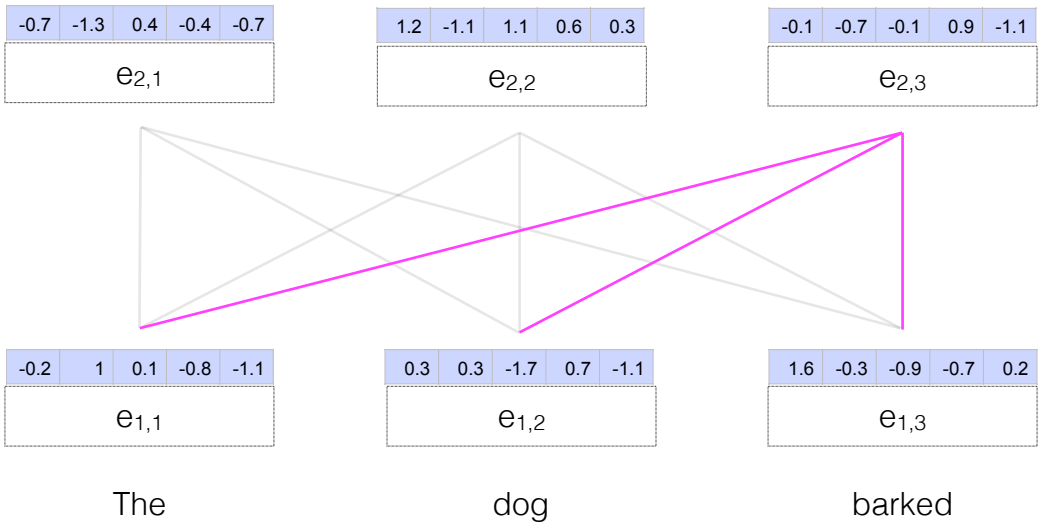


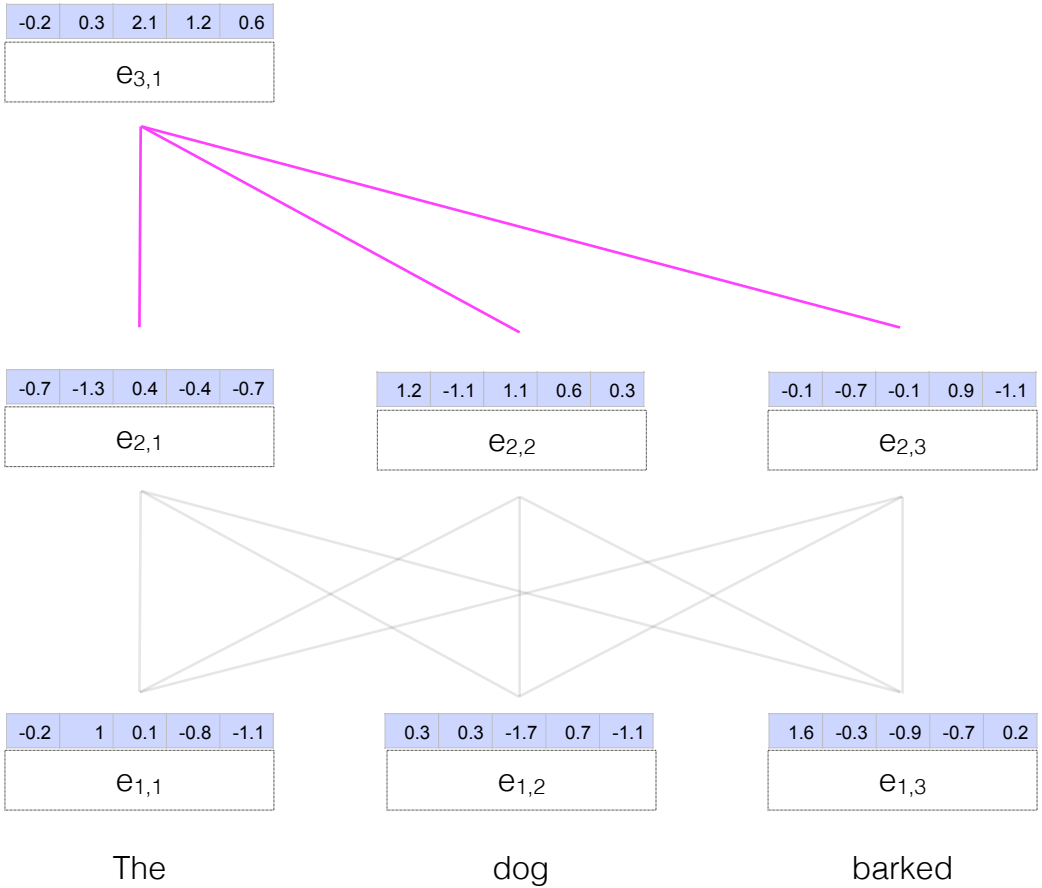

Each token in the input starts out represented by token and position embeddings
-0.2 1 0.1 -0.8 -1.1
e1,1
0.3 0.3 -1.7 0.7 -1.1
e1,2
1.6 -0.3 -0.9 -0.7 0.2
e1,3
The
dog
barked

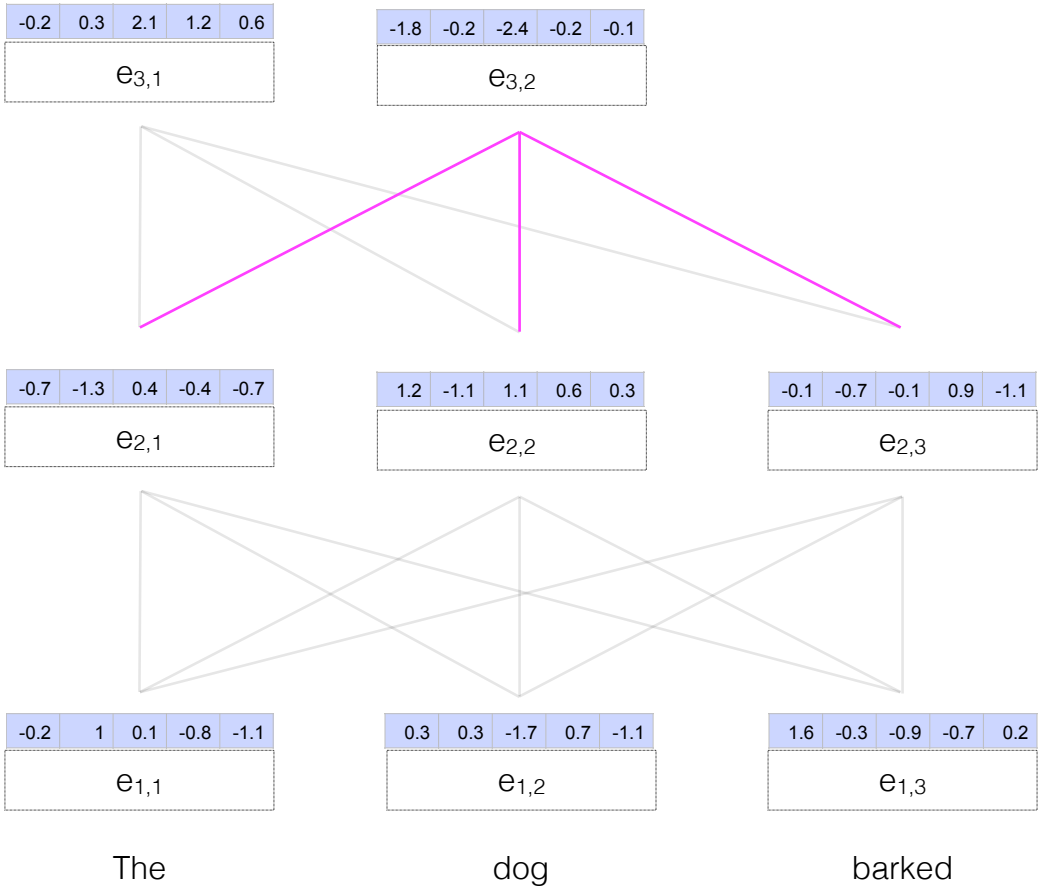

The value for time step j at layer i is the result of attention over all time steps in the previous layer i-1
-0.7 -1.3 0.4 -0.4 -0.7
e2,1
-0.2 1 0.1 -0.8 -1.1
e1,1
0.3 0.3 -1.7 0.7 -1.1
e1,2
1.6 -0.3 -0.9 -0.7 0.2
e1,3
The
dog
barked

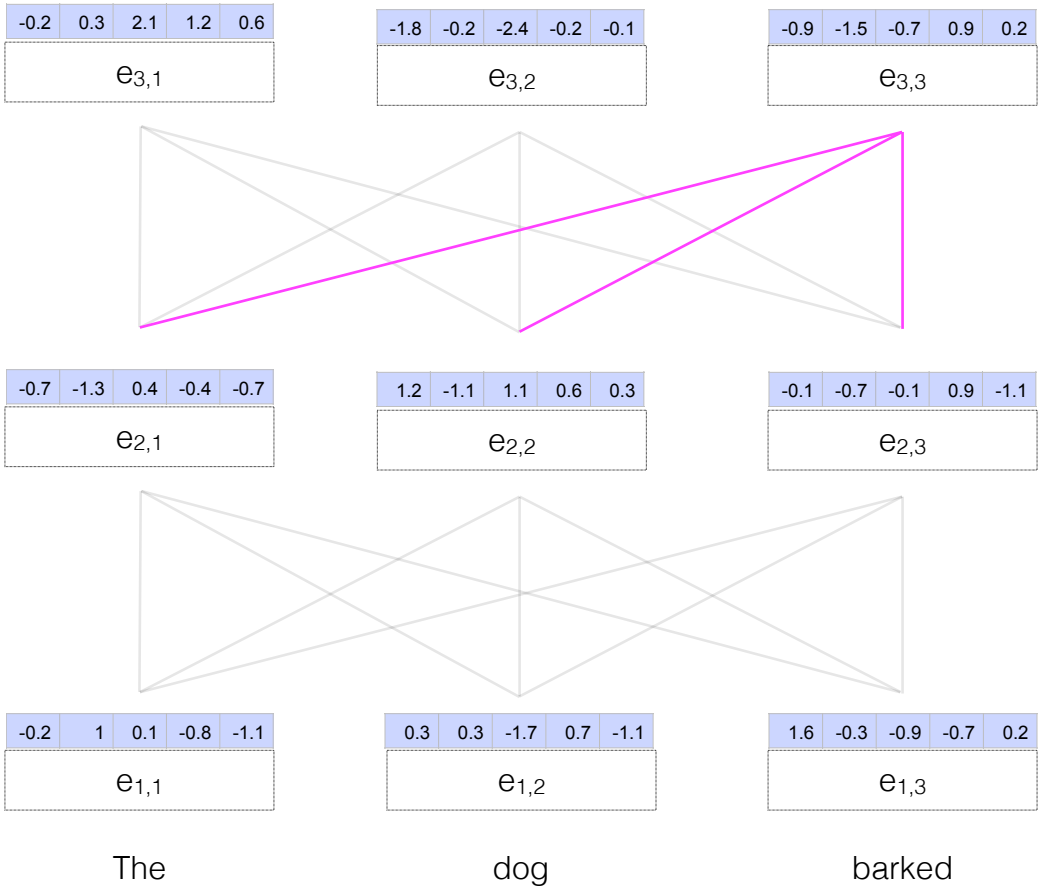

-0.7 -1.3 0.4 -0.4 -0.7
e2,1
-0.2 1 0.1 -0.8 -1.1
e1,1
0.3 0.3 -1.7 0.7 -1.1
e1,2
1.6 -0.3 -0.9 -0.7 0.2
e1,3
The
dog
barked


-0.7 -1.3 0.4 -0.4 -0.7
e2,1
1.2 -1.1 1.1 0.6 0.3
e2,2
-0.2 1 0.1 -0.8 -1.1
e1,1
0.3 0.3 -1.7 0.7 -1.1
e1,2
1.6 -0.3 -0.9 -0.7 0.2
e1,3
The
dog
barked

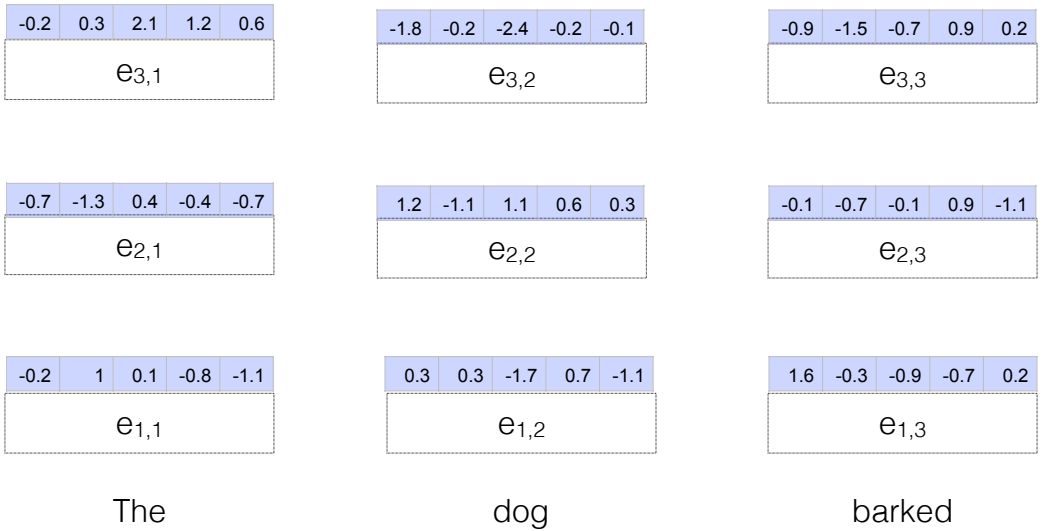

-0.7 -1.3 0.4 -0.4 -0.7
e2,1
1.2 -1.1 1.1 0.6 0.3
e2,2
-0.1 -0.7 -0.1 0.9 -1.1
e2,3
-0.2 1 0.1 -0.8 -1.1
e1,1
0.3 0.3 -1.7 0.7 -1.1
e1,2
1.6 -0.3 -0.9 -0.7 0.2
e1,3
The
dog
barked


-0.2 0.3 2.1 1.2 0.6
e3,1
-0.7 -1.3 0.4 -0.4 -0.7
e2,1
1.2 -1.1 1.1 0.6 0.3
e2,2
-0.1 -0.7 -0.1 0.9 -1.1
e2,3
-0.2 1 0.1 -0.8 -1.1
e1,1
0.3 0.3 -1.7 0.7 -1.1
e1,2
1.6 -0.3 -0.9 -0.7 0.2
e1,3
The
dog
barked


-0.2 0.3 2.1 1.2 0.6
e3,1
-1.8 -0.2 -2.4 -0.2 -0.1
e3,2
-0.7 -1.3 0.4 -0.4 -0.7
e2,1
1.2 -1.1 1.1 0.6 0.3
e2,2
-0.1 -0.7 -0.1 0.9 -1.1
e2,3
-0.2 1 0.1 -0.8 -1.1
e1,1
0.3 0.3 -1.7 0.7 -1.1
e1,2
1.6 -0.3 -0.9 -0.7 0.2
e1,3
The
dog
barked


-0.2 0.3 2.1 1.2 0.6
e3,1
-1.8 -0.2 -2.4 -0.2 -0.1
e3,2
-0.9 -1.5 -0.7 0.9 0.2
e3,3
-0.7 -1.3 0.4 -0.4 -0.7
e2,1
1.2 -1.1 1.1 0.6 0.3
e2,2
-0.1 -0.7 -0.1 0.9 -1.1
e2,3
-0.2 1 0.1 -0.8 -1.1
e1,1
0.3 0.3 -1.7 0.7 -1.1
e1,2
1.6 -0.3 -0.9 -0.7 0.2
e1,3
The
dog
barked

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



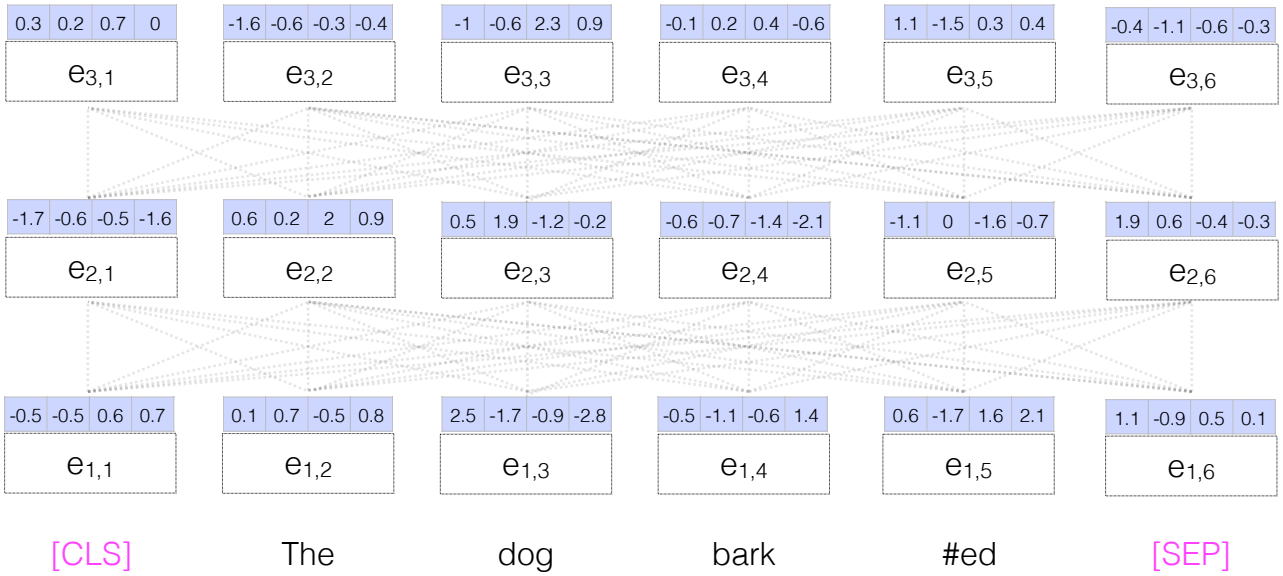
WordPiece

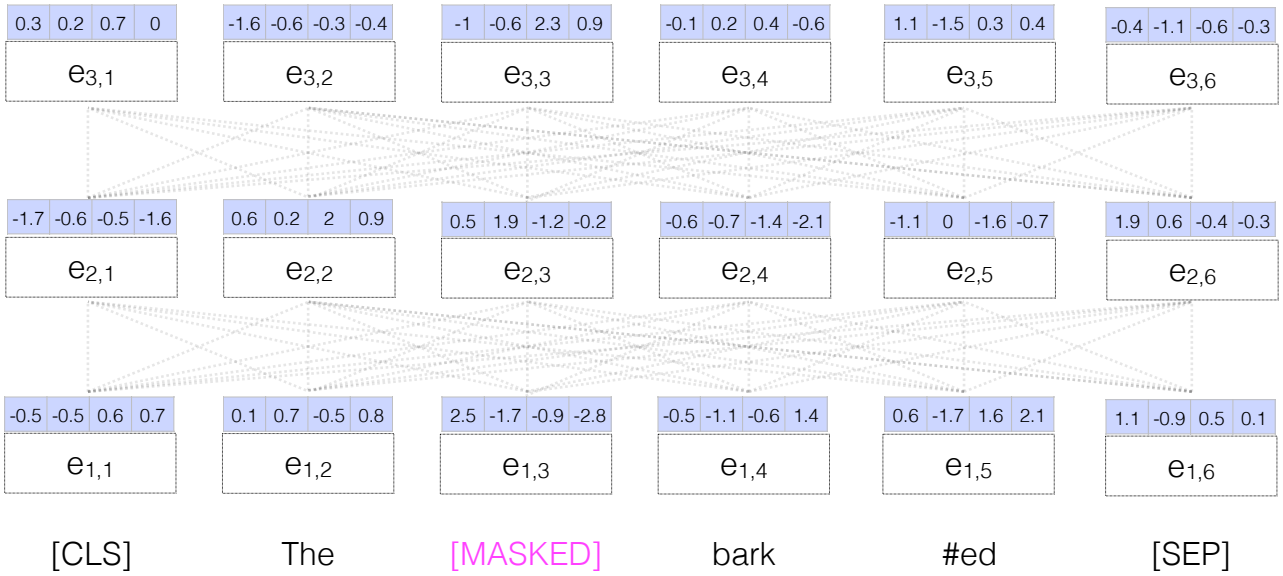
- BERT uses WordPiece tokenization, which segments some morphological structure of tokens
- Vocabulary size: 30,000

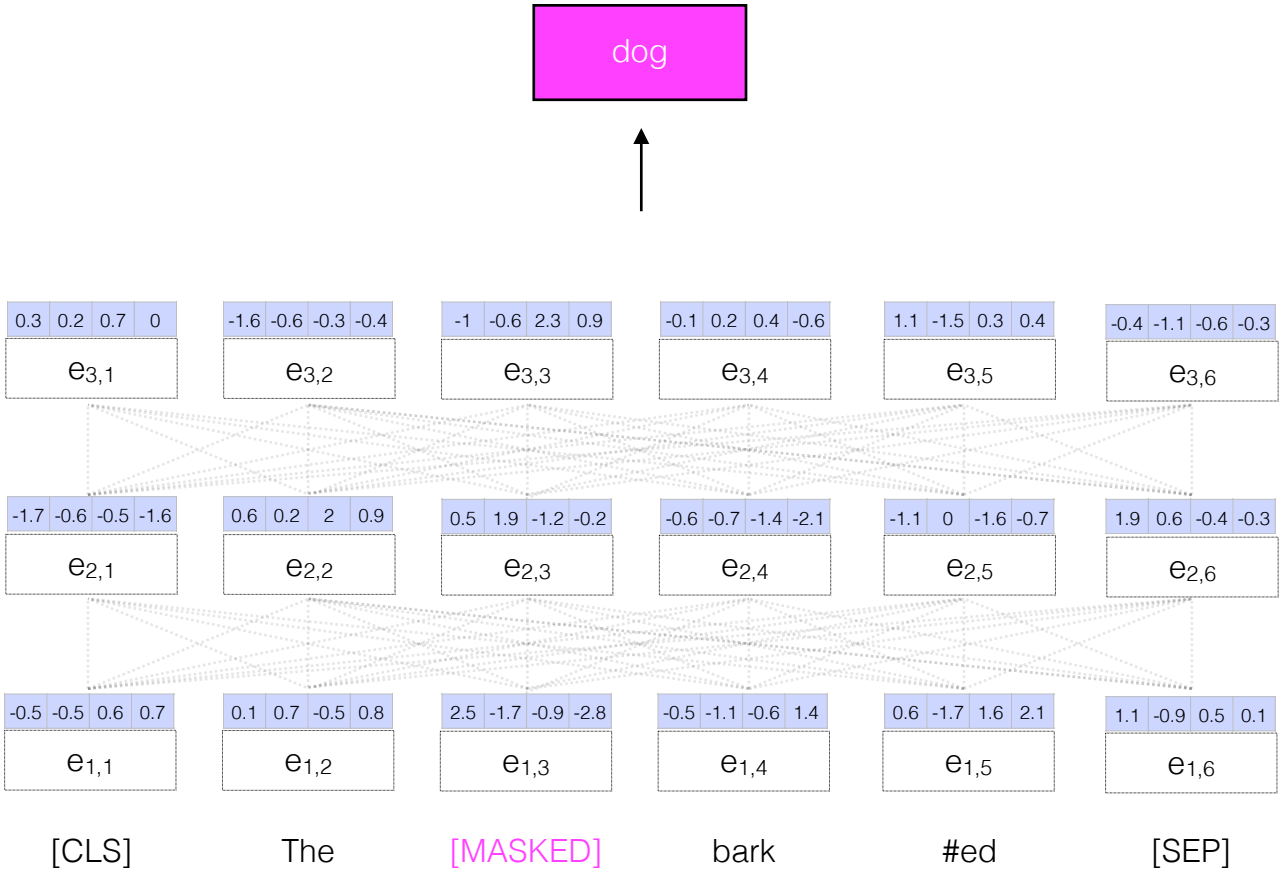
The	The
dog	dog
barked	bark #ed

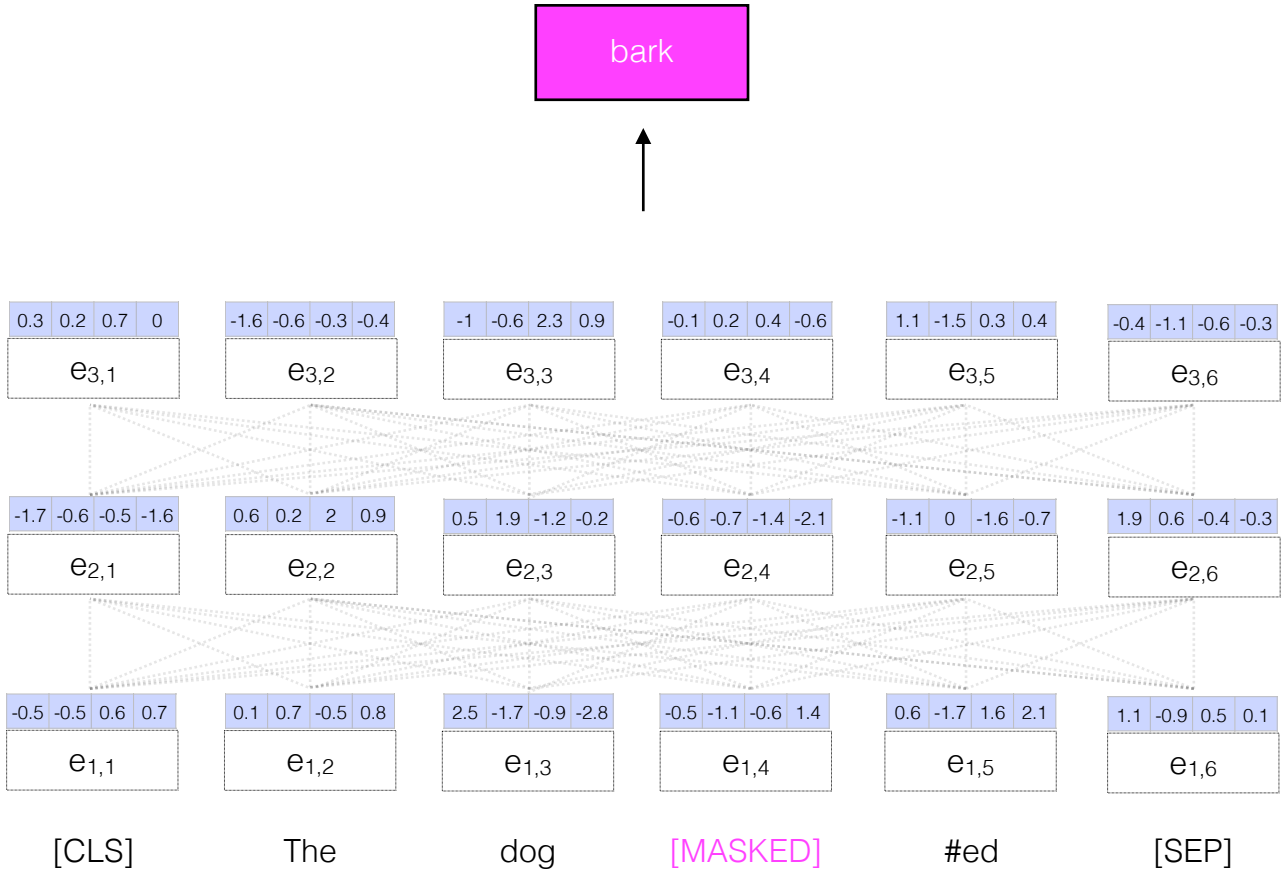
- BERT also encodes each sentence by appending a special token to the beginning ([CLS]) and end ([SEP]) of each sequence.

- This helps provides a single token that can be optimized to represent the entire sequence (e.g., for document classification)



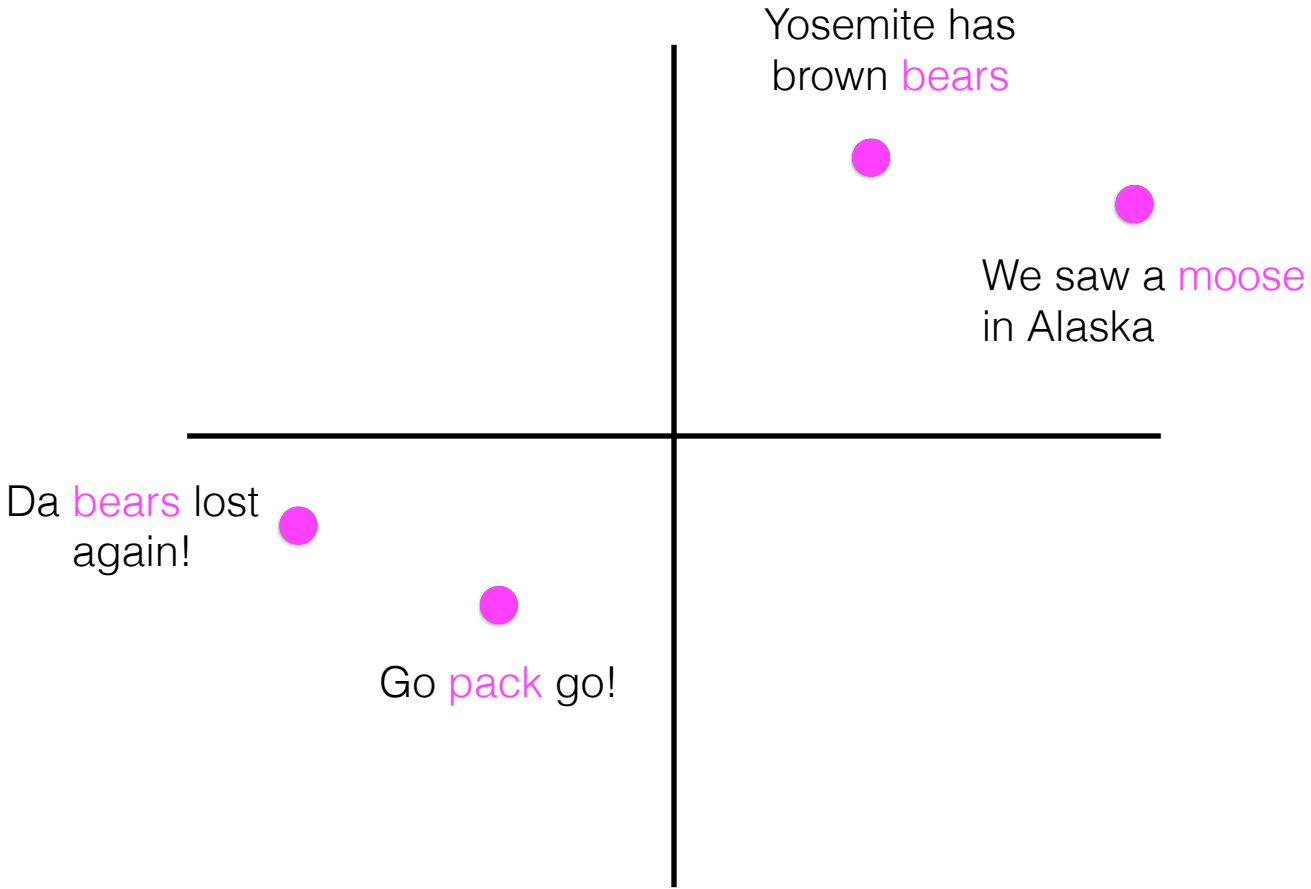

0.3 0.2 0.7 0
-1.6 -0.6 -0.3 -0.4
-1 -0.6 2.3 0.9
-0.1 0.2 0.4 -0.6
1.1 -1.5 0.3 0.4
-0.4 -1.1 -0.6 -0.3
e3,1
e3,2
e3,3
e3,4
e3,5
e3,6
-1.7 -0.6 -0.5 -1.6
0.6 0.2 2 0.9
0.5 1.9 -1.2 -0.2
-0.6 -0.7 -1.4 -2.1
-1.1 0 -1.6 -0.7
1.9 0.6 -0.4 -0.3
e2,1
e2,2
e2,3
e2,4
e2,5
e2,6
-0.5 -0.5 0.6 0.7
0.1 0.7 -0.5 0.8
2.5 -1.7 -0.9 -2.8
-0.5 -1.1 -0.6 1.4
0.6 -1.7 1.6 2.1
1.1 -0.9 0.5 0.1
e1,1
e1,2
e1,3
e1,4
e1,5
e1,6
[CLS]
The
[MASKED]
bark
#ed
[SEP]


dog
0.3 0.2 0.7 0
e3,1
-1.6 -0.6 -0.3 -0.4
e3,2
-1 -0.6 2.3 0.9
e3,3
-0.1 0.2 0.4 -0.6
e3,4
1.1 -1.5 0.3 0.4
e3,5
-0.4 -1.1 -0.6 -0.3
e3,6
-1.7 -0.6 -0.5 -1.6
e2,1
0.6 0.2 2 0.9
e2,2
0.5 1.9 -1.2 -0.2
e2,3
-0.6 -0.7 -1.4 -2.1
e2,4
-1.1 0 -1.6 -0.7
e2,5
1.9 0.6 -0.4 -0.3
e2,6
-0.5 -0.5 0.6 0.7
e1,1
0.1 0.7 -0.5 0.8
e1,2
2.5 -1.7 -0.9 -2.8
e1,3
-0.5 -1.1 -0.6 1.4
e1,4
0.6 -1.7 1.6 2.1
e1,5
1.1 -0.9 0.5 0.1
e1,6
[CLS]
The
[MASKED]
bark
#ed
[SEP]


bark
0.3 0.2 0.7 0
e3,1
-1.6 -0.6 -0.3 -0.4
e3,2
-1 -0.6 2.3 0.9
e3,3
-0.1 0.2 0.4 -0.6
e3,4
1.1 -1.5 0.3 0.4
e3,5
-0.4 -1.1 -0.6 -0.3
e3,6
-1.7 -0.6 -0.5 -1.6
e2,1
0.6 0.2 2 0.9
e2,2
0.5 1.9 -1.2 -0.2
e2,3
-0.6 -0.7 -1.4 -2.1
e2,4
-1.1 0 -1.6 -0.7
e2,5
1.9 0.6 -0.4 -0.3
e2,6
-0.5 -0.5 0.6 0.7
e1,1
0.1 0.7 -0.5 0.8
e1,2
2.5 -1.7 -0.9 -2.8
e1,3
-0.5 -1.1 -0.6 1.4
e1,4
0.6 -1.7 1.6 2.1
e1,5
1.1 -0.9 0.5 0.1
e1,6
[CLS]
The
dog
[MASKED]
#ed
[SEP]

BERT

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


Yosemite has
brown bears
We saw a moose
in Alaska
Da bears lost
again!
Go pack go!

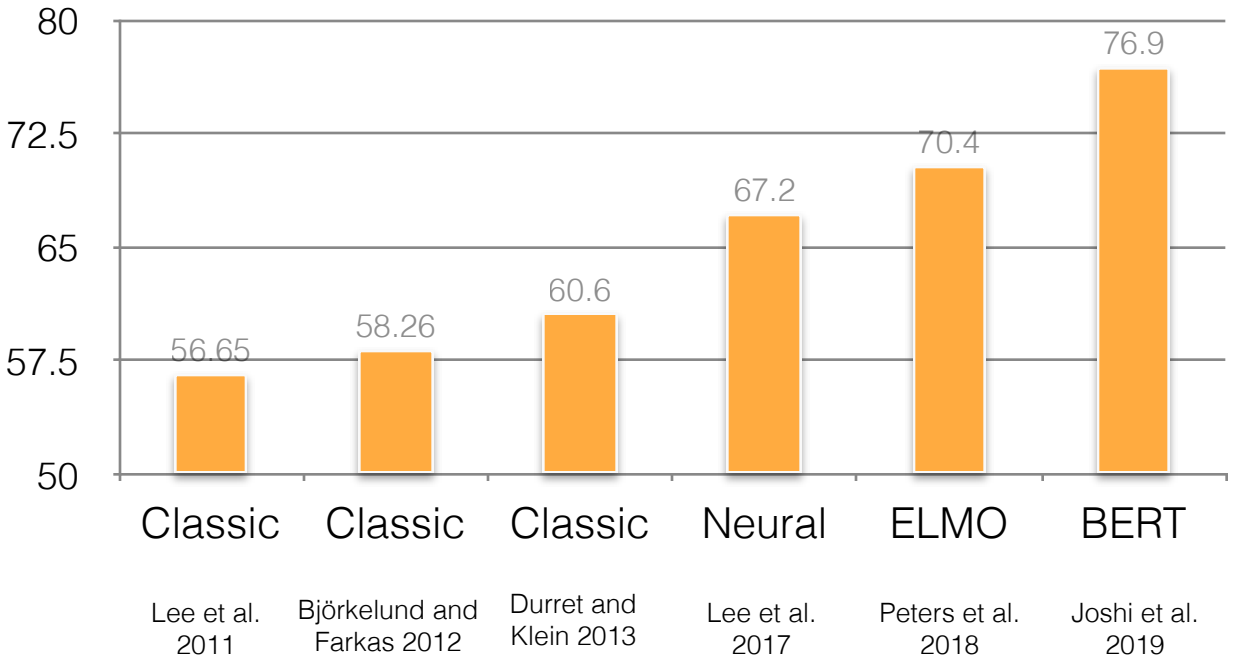
When plugged into NLP systems, contextual embeddings generally raise the state of the art.

ELMo

TASK	PREVIOUS SOTA		OUR BASELINE	ELMO + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

Table 1: Test set comparison of ELMo enhanced neural models with state-of-the-art single model baselines across six benchmark NLP tasks. The performance metric varies across tasks – accuracy for SNLI and SST-5; F_1 for SQuAD, SRL and NER; average F_1 for Coref. Due to the small test sizes for NER and SST-5, we report the mean and standard deviation across five runs with different random seeds. The "increase" column lists both the absolute and relative improvements over our baseline.

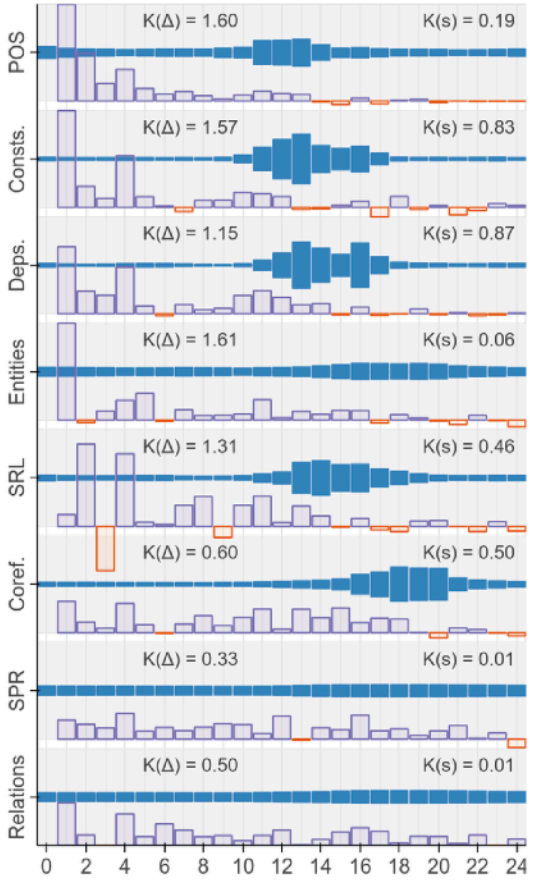
Progress — Coreference resolution



Probing

- Even though BERT is mainly trained on a language modeling objective, it learns a lot about the structure of language — even without direct training data for specific linguistic tasks.

- Probing experiments uncover what—-and where (in what layers)—-pretrained BERT encodes this information.



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

What can we use contextual embeddings for?

Contextual Nearest Neighbors

audentes fortuna iuvat

0.926	audentes forsque deusque iuvat.	Ov., Fast.
0.864	audentis fortuna iuvat, piger ipse sibi opstat.	Sen., Ep.
0.846	audentes in tela ruunt.	Vida, Scacchia Ludus
0.840	audentes facit amissae spes lapsa salutis, succurruntque duci	Vida, Scacchia Ludus
0.837	... audentis fortuna iuuat.' haec ait, et cum se uersat ...	Verg., Aen.

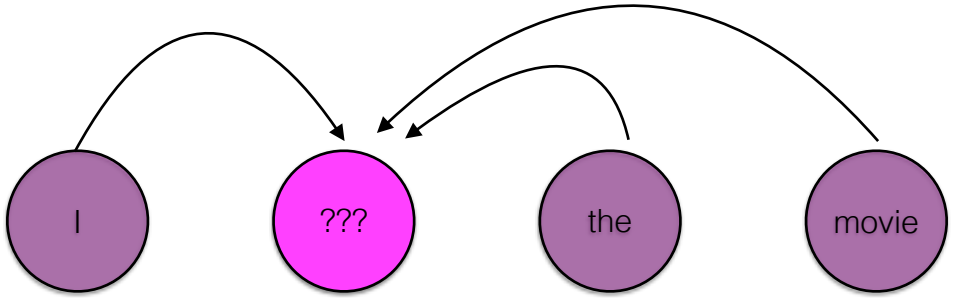
Bamman and Burns (2020), “Latin BERT: A Contextual Language Model for Classical Philology”

Textual reconstruction

Assael et al. 2022, "Restoring and attributing ancient texts using deep neural networks," *Nature*

Masked language model

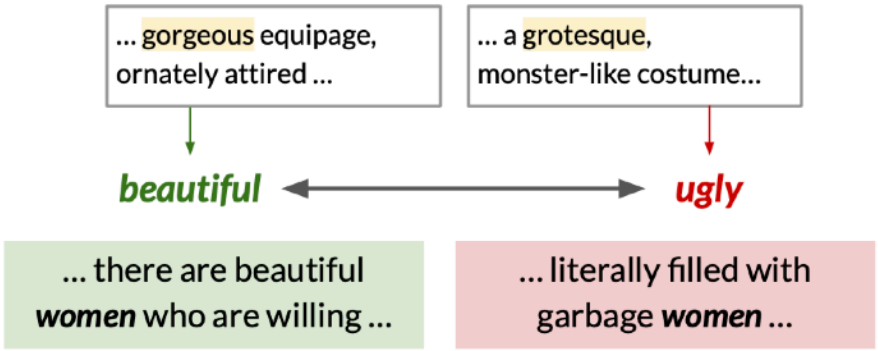
Use any context (left or right) to predict a masked word



$$P\left(w_t \mid w_{\neg t}\right)$$

Contextual SemAxis

- Creating a contextual semantic axis from token representations allows situating specific mentions along that axis (not just word types).



Lucy et al. 2022, “Discovering Differences in the Representation of People using Contextualized Semantic Axes”

- Ryan Heuser (2017), “Word Vectors in the Eighteenth Century” *DH*

(Static) SemAxis

- Define a set of terms that comprise the endpoints of an axis of interest and average them up to form axis endpoint vectors.

$$S^- = \{v_1^-, \ldots, v_n^-\}$$

{man, he, mr.}

$$V^- = \frac{1}{n} \sum_{1}^{N} v_i^-$$

$$S^+ = \{v_1^+, \ldots, v_m^+\}$$

{woman, she, miss, mrs.}

$$V^+ = \frac{1}{M} \sum_{1}^{M} v_i^+$$

An et al. 2018, “SemAxis: A Lightweight Framework to Characterize Domain-Specific Word Semantics Beyond Sentiment”

(Static) SemAxis

- The axis vector is then the difference between the two endpoint vectors

{man, he, mr.}

$$V^- = \frac{1}{n}\sum_{1}^{N} v_i^-$$

{woman, she, miss, mrs.}

$$V^+ = \frac{1}{M}\sum_{1}^{M} v_i^+$$

$$V_{\text{axis}} = V^+ - V^-$$

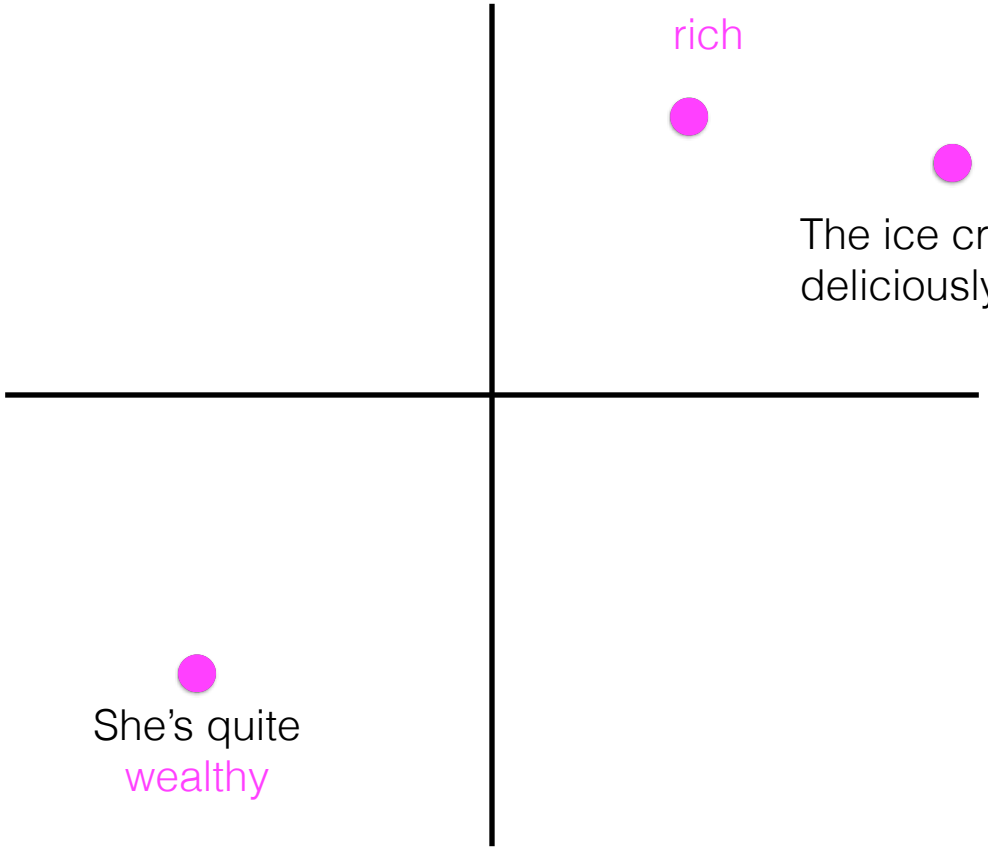
An et al. 2018, “SemAxis: A Lightweight Framework to Characterize Domain-Specific Word Semantics Beyond Sentiment”

(Static) SemAxis

- For any vector, we can find its position along this axis by taking the cosine similarity with it (or dot product if all the vectors are normalized to unit length)

$$\text{Semaxis score} = \cos\left(\text{football}, V_{\text{axis}}\right)$$

An et al. 2018, “SemAxis: A Lightweight Framework to Characterize Domain-Specific Word Semantics Beyond Sentiment”


This cake is too rich
The ice cream is deliciously sweet
Victoria Beckham drove a Rolls Royce to school (her family was rich).
She's quite wealthy

- Define a set of BERT word vectors that comprise the endpoints of an axis of interest and average them up to form axis endpoint vectors.

$$V^- = \frac{1}{n}\sum_{1}^{N} v_i^-$$

$$V^+ = \frac{1}{M}\sum_{1}^{M} v_i^+$$

- She's quite wealthy
- Victoria Beckham drove a Rolls Royce to school (her family was rich).
- John was rich and Bill was poor.

- He lost his job and is now poor.
- He lives in poverty.
- John was rich and Bill was poor.

(Contextual) SemAxis

- The axis vector is then the difference between the two endpoint vectors

$$V^- = \frac{1}{n}\sum_{1}^{N} v_i^-$$

$$V^+ = \frac{1}{M}\sum_{1}^{M} v_i^+$$

$$V_{\text{axis}} = V^+ - V^-$$

(Contextual) SemAxis

- For any BERT word embedding, we can find its position along this axis by taking the cosine similarity with it.

$$\text{Semaxis score} = \cos\left(\text{golf}_{\text{(He plays golf on the weekend)}}, V_{\text{axis}}\right)$$

$$\text{Semaxis score} = \cos\left(\text{golf}_{\text{(He drives a VW golf)}}, V_{\text{axis}}\right)$$

Activity

`3.embeddings/BERT.ipynb`

- Explore BERT through the huggingface transformers library and use it to find contextual nearest neighbors.
- What else could we use contextual embeddings for?