Data Wrangling and Data Analysis
Text mining #2
Sentiment analysis & Embeddings

DL Oberski

Dept of Methodology & Statistics
Utrecht University

With slides by **Dr. Dong Nguyen**

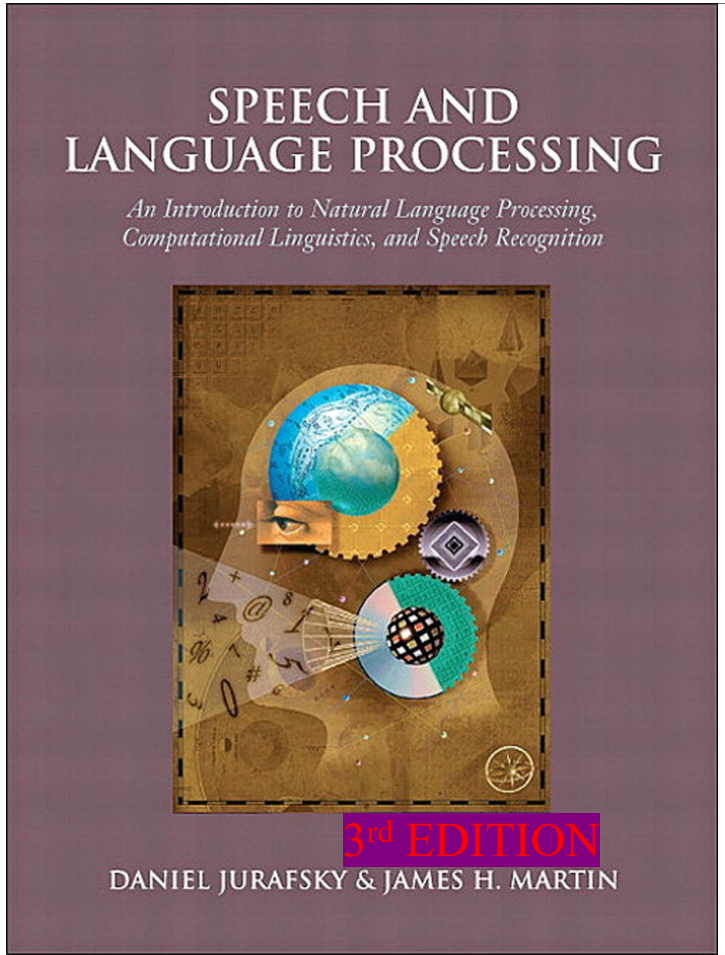
Dept of Computing Sciences
Utrecht University

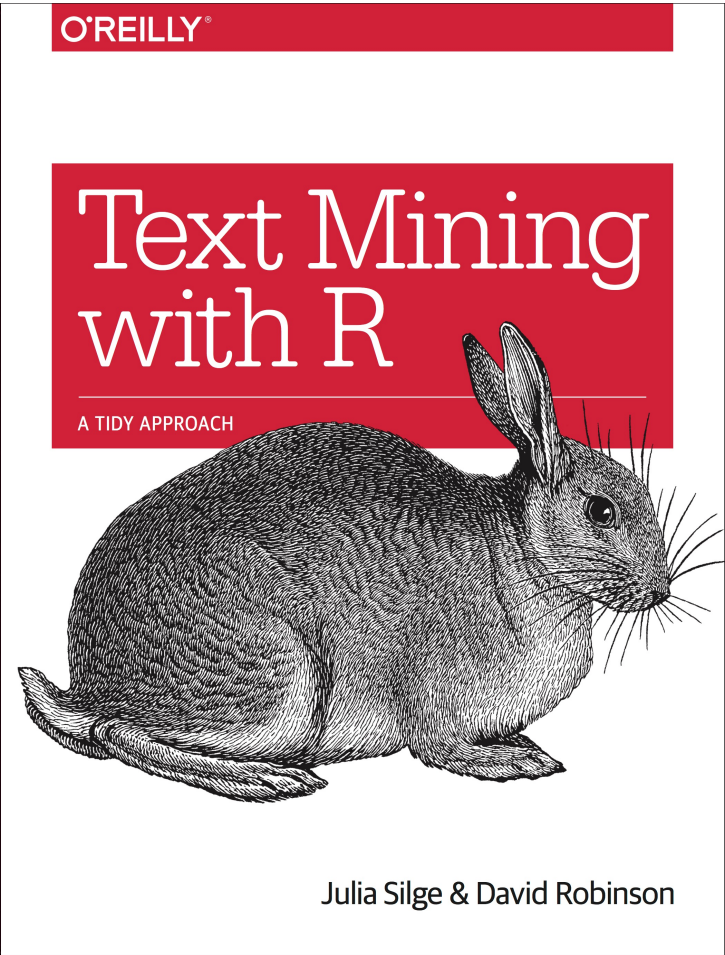
This week

- Day 1: Clustering #2: Model-based clustering
- Day 2: Text mining #1: regular expressions, BoW, TF-IDF
- **Day 3: Text mining #2: sentiment analysis, embeddings**

Readings about text mining

- Jurafsky & Martin (2021). *Speech and language processing (3rd ed draft)*
- https://web.stanford.edu/~jurafsky/slp3/ed3bookfeb3_2024.pdf
  - Sections
  - 2.1, 2.4, (regular expressions)
  - 6.2, 6.3, 6.4, 6.5, 6.8
- *Silge & Robinson (2021). Text mining with R: A tidy approach.* https://www.tidytextmining.com/
  - Chapter 3
- More accessible (?) intro to regular expressions:
  - **R4 data science** ch. 14
  - https://r4ds.had.co.nz/strings.html#matching-patterns-with-regular-expressions





Word embeddings

Word representations

How can we represent the *meaning* of words?

Word representations

How can we represent the *meaning* of words?

So we can ask:

- How similar is *cat* to *dog*, or *Paris* to *London*?
- How similar is *document A* to *document B*?

Word representations

How can we represent the *meaning* of words?

So we can ask:

- How similar is *cat* to *dog*, or *Paris* to *London*?
- How similar is *document A* to *document B*?

And use such representations for:

- various NLP tasks: translation, classification, etc.
- studying linguistic questions

Word as vectors

Key idea: Can we represent words as vectors?

The vector representations should:

- capture semantics
  - similar words should be close to each other in the vector space
  - relation between two vectors should reflect the relationship between the two words
- be efficient (vectors with fewer dimensions are easier to work with)
- be interpretable

Word as vectors

Key idea: Can we represent words as vectors?

The vector representations should:

- capture semantics
  - similar words should be close to each other in the vector space
  - relation between two vectors should reflect the relationship between the two words
- be efficient (vectors with fewer dimensions are easier to work with)
- be interpretable

How similar are *smart* and *intelligent?* (not similar 0–10 very similar):
How similar are *easy* and *big* (not similar 0–10 very similar):

Word as vectors

Key idea: Can we represent words as vectors?

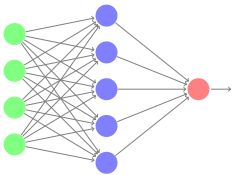
The vector representations should:

- capture semantics
  - similar words should be close to each other in the vector space
  - relation between two vectors should reflect the relationship between the two words
- be efficient (vectors with fewer dimensions are easier to work with)
- be interpretable

How similar are *smart* and *intelligent?* (not similar 0–10 very similar): 9.2
How similar are *easy* and *big* (not similar 0–10 very similar): 1.12
(*SimLex-999 dataset*)

How are they used?

How are they used?



In neural networks (text classification, sequence tagging, etc..)

cat	0.52	0.48	-0.01	···	0.28
dog	0.32	0.42	-0.09	···	0.78

As research objects

Properties

We can use cosine similarity to find similar words in the vector space.

- **dog**: *dogs, cat, man, cow, horse*
- **car**: *driver, cars, automobile, vehicle, race*
- **amsterdam**: *netherlands, rotterdam, dutch, centraal, paris*
- **chocolate**: *candy, beans, caramel, butter, liquor*

Exercise (5 min)

- Go to https://projector.tensorflow.org/. The site should load 'Word2Vec 10K' vectors by default (see left panel).
- What are the 5 nearest words to '*cat*'?
- What are the 5 nearest words to '*computer*'?

Words as vectors

One hot encoding

Map each word to a unique identifier
e.g. *cat (3)* and *dog (5)*.
→ Vector representation: all zeros, except 1 at the ID

cat	0	0	1	0	0	0	0
dog	0	0	0	0	1	0	0
car	0	0	0	0	0	0	1

One hot encoding

Map each word to a unique identifier
e.g. *cat (3)* and *dog (5)*.
→ Vector representation: all zeros, except 1 at the ID

cat	0	0	1	0	0	0	0
dog	0	0	0	0	1	0	0
car	0	0	0	0	0	0	1

What are limitations of one hot encodings?

One hot encoding

Map each word to a unique identifier
e.g. *cat (3)* and *dog (5)*.
→ Vector representation: all zeros, except 1 at the ID

cat	0	0	1	0	0	0	0
dog	0	0	0	0	1	0	0
car	0	0	0	0	0	0	1

Even related words have distinct vectors!

High number of dimensions

Distributional hypothesis

some believe that	**wampos**	scales have medicinal qualities
approach to fighting	**wampos**	(and general wildlife) trafficking
Even though	**wampos**	scales are made of exactly the

Distributional hypothesis

some believe that	**wampos**	scales have medicinal qualities
approach to fighting	**wampos**	(and general wildlife) trafficking
Even though	**wampos**	scales are made of exactly the

What is a **wampos**?

Distributional hypothesis

Figure: Photo by Piekfrosch; CC-BY-SA-3.0

some believe that	**wampos**	scales have medicinal qualities
approach to fighting	**wampos**	(and general wildlife) trafficking
Even though	**wampos**	scales are made of exactly the

wampos = pangolin

You shall know a word by the company it keeps (Firth, J. R. 1957:11)

Distributional hypothesis

Figure: Photo by Piekfrosch; CC-BY-SA-3.0

some believe that **wampos** scales have medicinal qualities
approach to fighting **wampos** (and general wildlife) trafficking
Even though **wampos** scales are made of exactly the

wampos = pangolin

You shall know a word by the company it keeps (Firth, J. R. 1957:11)

The distributional hypothesis: Words that occur in similar contexts tend to have similar meanings

Word vectors based on co-occurrences

documents as context
word-document matrix

	doc_1	doc_2	doc_3	doc_4	doc_5	doc_6	doc_7
cat	5	2	0	1	4	0	0
dog	7	3	1	0	2	0	0
car	0	0	1	3	2	1	1

Word vectors based on co-occurrences

documents as context
word-document matrix

	doc_1	doc_2	doc_3	doc_4	doc_5	doc_6	doc_7
cat	5	2	0	1	4	0	0
dog	7	3	1	0	2	0	0
car	0	0	1	3	2	1	1

neighboring words as context
word-word matrix

	cat	dog	car	bike	book	house	tree
cat	0	3	1	1	1	2	3
dog	3	0	2	1	1	3	1
car	0	0	1	3	2	1	1

Word vectors based on co-occurrences

There are many variants:

- Context (words, documents, which window size, etc.)
- Weighting (raw frequency, etc.)

Vectors are sparse: Many zero entries.
Therefore: Dimensionality reduction is often used (e.g., SVD)

These methods are sometimes called **count-based** methods as they work directly on **co-occurrence** counts.

Word embeddings

Word embeddings

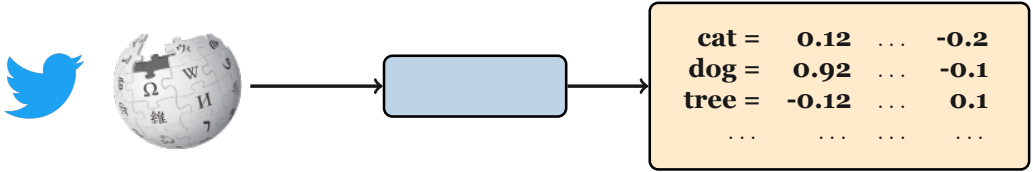
Word embeddings:

- Vectors are short; typically 50-1024 dimensions ☺
- Vectors are dense (mostly non-zero values)
- Very effective for many NLP tasks ☺
- Individual dimensions are less interpretable ☹

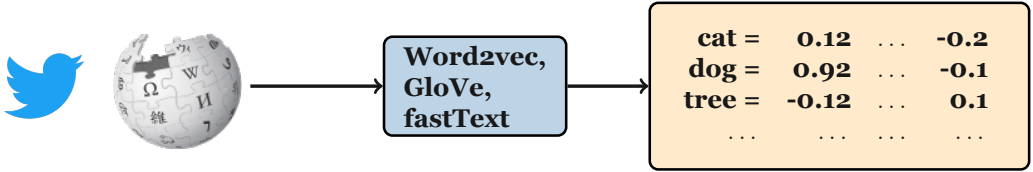
cat	0.52	0.48	-0.01	···	0.28
dog	0.32	0.42	-0.09	···	0.78

How do we learn word embeddings?

Learning word embeddings



Learning word embeddings



Training data

How can we train a model to learn the meaning of words?
Which data can we use for supervised learning?

Training data

How can we train a model to learn the meaning of words?
Which data can we use for supervised learning?

Key idea:

Use text itself as training data for the model!

A form of *self-supervision*.

Training data

How can we train a model to learn the meaning of words?
Which data can we use for supervised learning?

Example: Train a neural network to predict the next word given previous words.

Key idea:

Use text itself as training data for the model!

A form of *self-supervision*.

A neural probabilistic language model. Bengio et al. (2003), JMLR [url]

Natural language processing (almost) from scratch, Collobert et al. (2011), JMLR, [url]

Exercise: Word prediction task

`yesterday I went to the ?`

`A new study has highlighted the positive ?`

Which word comes next?

Word2Vec

The domestic **cat** is a small, typically furry carnivorous mammal

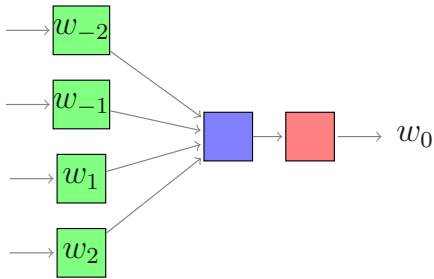
w_{-2} w_{-1} w_0 w_1 w_2 w_3 w_4 w_5

We have **target** words (*cat*) and **context** words (here: window=5).

Remember: distributional hypothesis

Word2Vec

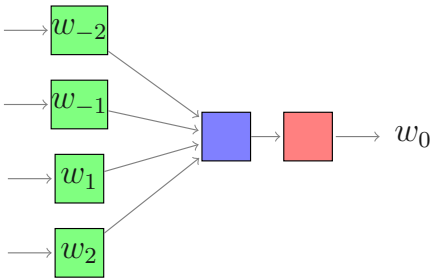
Continuous Bag-Of-Words (CBOW)



`one snowy ? she went`

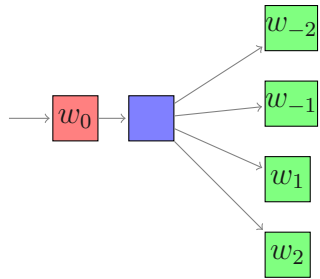
Word2Vec

Continuous Bag-Of-Words (CBOW)



one snowy **?** she went

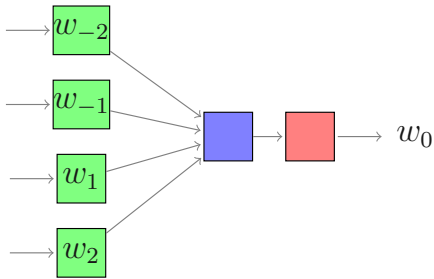
skipgram



? **?** day **?** **?**

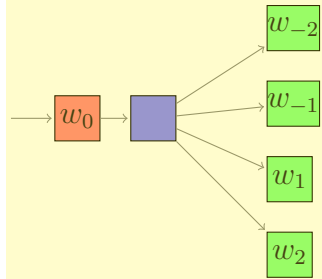
Word2Vec

Continuous Bag-Of-Words (CBOW)



one snowy **?** she went

skipgram



? **?** day **?** **?**

Word2Vec: skipgram overview

The domestic **cat** is a small, typically furry carnivorous mammal

word (w)	context (c)	label
cat	small	1
cat	furry	1
cat	car	0
...	...	...

Word2Vec: skipgram overview

The domestic **cat** is a small, typically furry carnivorous mammal

word (w)	context (c)	label
cat	small	1
cat	furry	1
cat	car	0
...	...	...

1. **Create examples**
   - Positive examples: Target word and neighboring context
   - Negative examples: Target word and randomly sampled words from the lexicon (*negative sampling)*
2. Train a **logistic regression** model to distinguish between the positive and negative examples
3. The resulting **weights** are the embeddings!

Word2Vec: skipgram overview

The domestic **cat** is a small, typically furry carnivorous mammal

word (w)	context (c)	label
cat	small	1
cat	furry	1
cat	car	0
…	…	…

Embedding vectors are essentially a byproduct!

1. **Create examples**
   - Positive examples: Target word and neighboring context
   - Negative examples: Target word and randomly sampled words from the lexicon (*negative sampling)*
2. Train a **logistic regression** model to distinguish between the positive and negative examples
3. The resulting **weights** are the embeddings!

Word2Vec: skipgram

The domestic **cat** is a small, typically furry carnivorous mammal
$c1$ $c2$ w $c3$ $c4$ $c5$ $c6$ $c7$

We have **target** words (*cat*) and **context** words (here: window=5).

The probability that *c* is a real context word:

$$P(+|w, c)$$

The probability that *c* is not a real context word:

$$P(-|w, c)$$

See also: 6.8 of Speech and Language Processing (3rd ed. draft), Dan Jurafsky and James H. Martin
https://web.stanford.edu/~jurafsky/slp3/

Word2Vec: skipgram

Intuition: A word c is likely to occur near the target if its embedding is similar to the target embedding.

$$\approx w \cdot c$$

Turn this into a probability using the sigmoid function

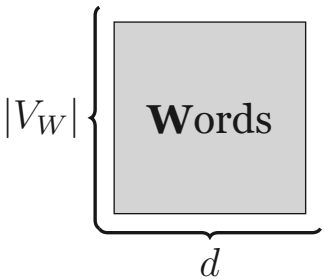
$$P(+|w, c) = \frac{1}{1 + e^{-w \cdot c}}$$

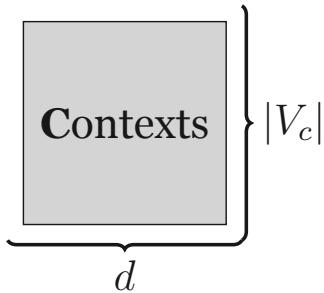
See also: 6.8 of Speech and Language Processing (3rd ed. draft), Dan Jurafsky and James H. Martin
https://web.stanford.edu/~jurafsky/slp3/

Word2Vec

Words:
Each word w is represented as a d-dimensional vector.

Contexts:
Each word w is represented as a d-dimensional vector.





All vectors are initialized with random weights.

Word2vec: skipgram (learning)

We **start** with random embedding vectors.

Word2vec: skipgram (learning)

We **start** with random embedding vectors.

During training:

- *Maximize* the similarity between the embeddings of the target word and context words from the positive examples
- *Minimize* the similarity between the embeddings of the target word and context words from the negative examples

Word2vec: skipgram (learning)

We **start** with random embedding vectors.

During training:

- *Maximize* the similarity between the embeddings of the target word and context words from the positive examples
- *Minimize* the similarity between the embeddings of the target word and context words from the negative examples

After training:

- frequent word-context pairs in data: $w \cdot c$ high
- not word-context pairs in data: $w \cdot c$ low

So: Words occurring in same contexts are close to each other

Word2vec: skipgram (learning)

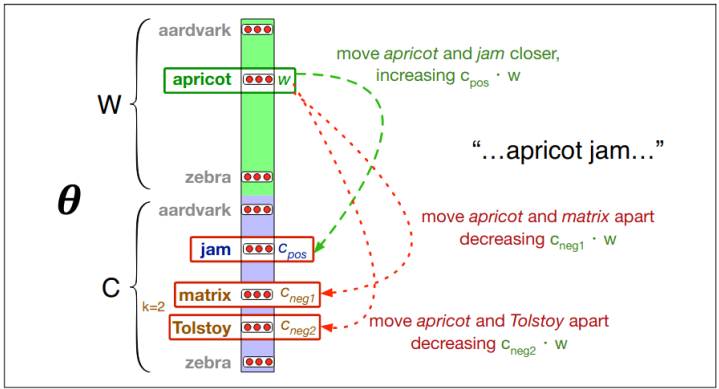


Figure: Figure 6.14 from Speech and Language Processing (3rd ed. draft), Dan Jurafsky and James H. Martin
https://web.stanford.edu/~jurafsky/slp3/

Pre-trained embeddings

- I want to build a system to solve a task (e.g. sentiment analysis)
  - Use pre-trained embedddings. Should I fine-tune?
    - Lots of data: yes
    - Just a small dataset: no
- Analysis (e.g. bias, semantic change)
  - Train embeddings from scratch

Properties of word embeddings

Properties of word embeddings: analogies

We can look at analogies in the vector space, for example:
king - man + woman ≈ queen

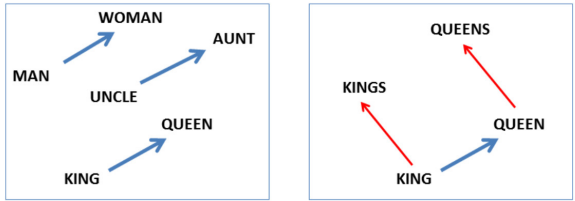
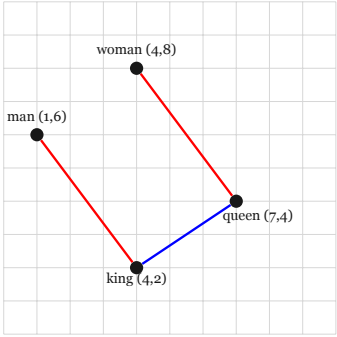


Figure: Figure 2 from Linguistic Regularities in Continuous Space Word Representations, Mikolov et al. NAACL 2013 [url]

Properties of word embeddings: analogies

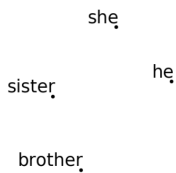
We can look at analogies in the vector space, for example:
king - man + woman ≈ queen




```
king-man = [4,2] - [1,6] = [3,-4]
king-man + woman = [3,-4] + [4,8] = [7,4]
```

Biases in word embeddings

Biases in word embeddings



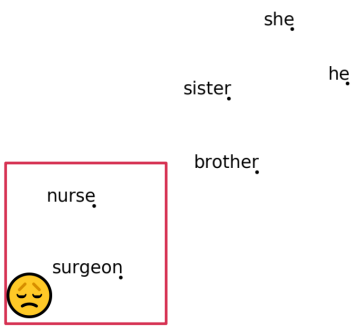
Measuring gender bias:

- To assess NLP models and investigate the impact of 'bias mitigation' techniques
- To study societal trends

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings, Bolukbasi, et al. NIPS 2016 [url]

Semantics derived automatically from language corpora contain human-like biases, Caliskan, Bryson, Narayanan, Science 2017 [url]

Biases in word embeddings



Pre-trained GloVe model on Twitter

Measuring gender bias:

- To assess NLP models and investigate the impact of 'bias mitigation' techniques
- To study societal trends

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings, Bolukbasi, et al. NIPS 2016 [url]

Semantics derived automatically from language corpora contain human-like biases, Caliskan, Bryson, Narayanan, Science 2017 [url]

Biases reflected in analogy tasks

Biases reflected in analogy tasks:

man is to *computer programmer* as *woman* is to ? : x = homemaker
father is to *doctor* as *mother* is to ? : x = nurse

Note: Input words are excluded as possible answers! (see also Nissim et al. 2020 [url])

Compare: gender-specific words (e.g., *brother, businesswoman*) vs. *gender-neutral* words (e.g. *nurse*, *teacher*).

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings, Bolukbasi, et al. NIPS 2016 [url]

Perpetuation of bias in sentiment analysis

> *"I had tried building an algorithm for sentiment analysis based on word embeddings [..]. When I applied it to restaurant reviews, I found it was ranking Mexican restaurants lower. The reason was not reflected in the star ratings or actual text of the reviews. It's not that people don't like Mexican food.* ***The reason was that the system had learned the word "Mexican" from reading the Web****."*

(emphasis mine)

http://blog.conceptnet.io/posts/2017/conceptnet-numberbatch-17-04-better-less-stereotyped-word-vectors/

The problem with static embeddings

- They are static! The embedding for a word doesn't reflect how its meaning changes in context.

  The chicken didn't cross the road because **it** was too tired

- What is the meaning represented in the static embedding for "it"?

Source: Jurafsky & Martin (2025)

Contextual embeddings

- Intuition: a representation of meaning of a word should be different in different contexts.

- **Contextual Embedding**: each word has a different vector that expresses different meanings depending on the surrounding words

- How to compute contextual embeddings?
  - **Attention**

Source: Jurafsky & Martin (2025)

Computer Science > Computation and Language

[Submitted on 12 Jun 2017 (v1), last revised 2 Aug 2023 (this version, v7)]

Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks in an encoder-decoder configuration. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Contextual embeddings

```
The chicken didn't cross the road because it
```

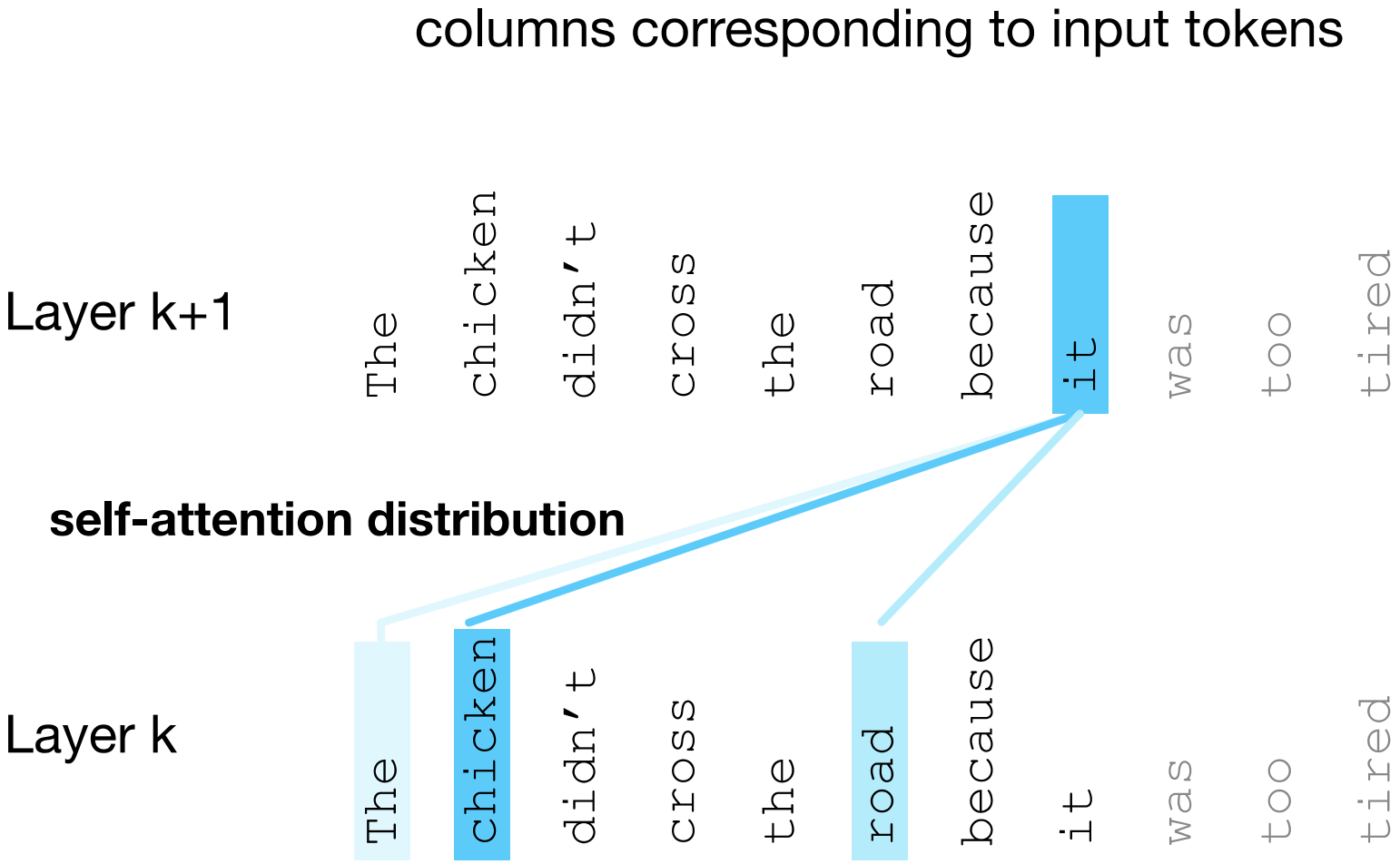
What should be the properties of "it"?

```
The chicken didn't cross the road because it was too tired
The chicken didn't cross the road because it was too wide
```

At this point in the sentence, it's probably referring to either the chicken or the street

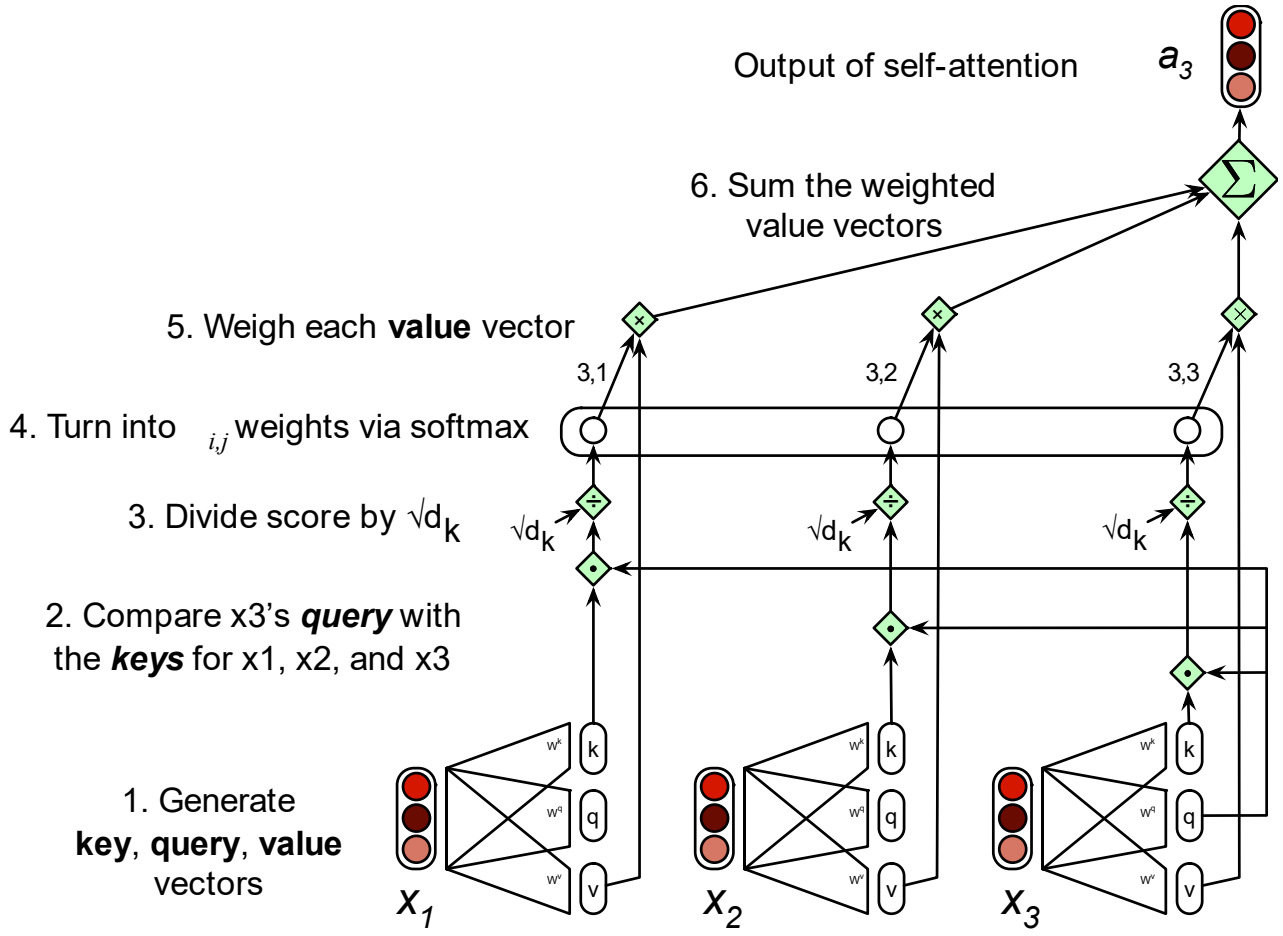
Source: Jurafsky & Martin (2025)

Contextual embeddings



Source: Jurafsky & Martin (2025)

One “attention head”



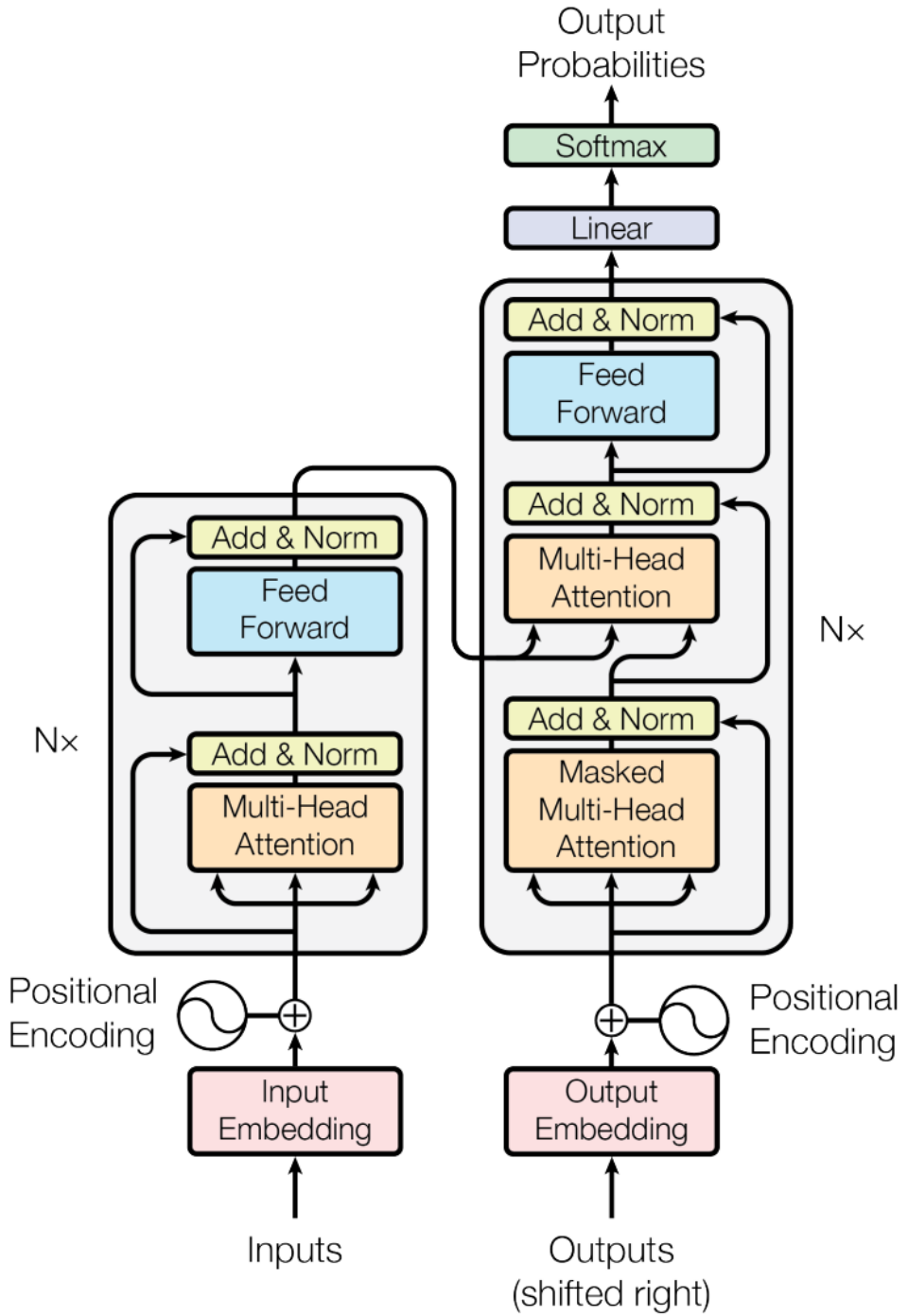
Source: Jurafsky & Martin (2025)

Transformers

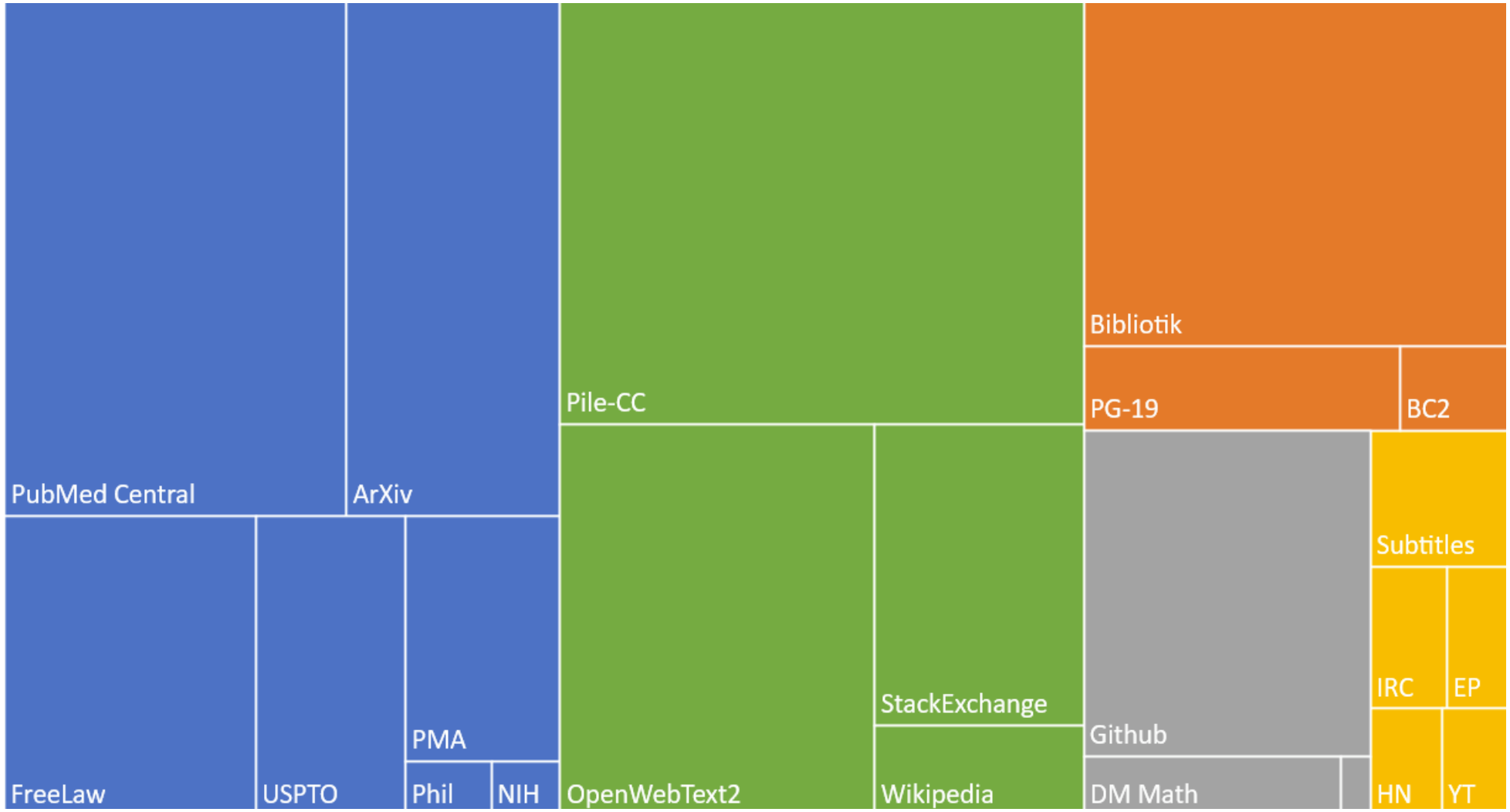
Llama 3.1 Example:

- Word embedding dimension is 16,384
- And positional embedding dim = 500,000
  - In each of the 126 layers…
    - Which each contain 128 attention heads…
      - With 8 key/value heads each.
- Followed by FF neural net of width 53,248

Total number of parameters: 405 billion



The Pile: a pretraining corpus



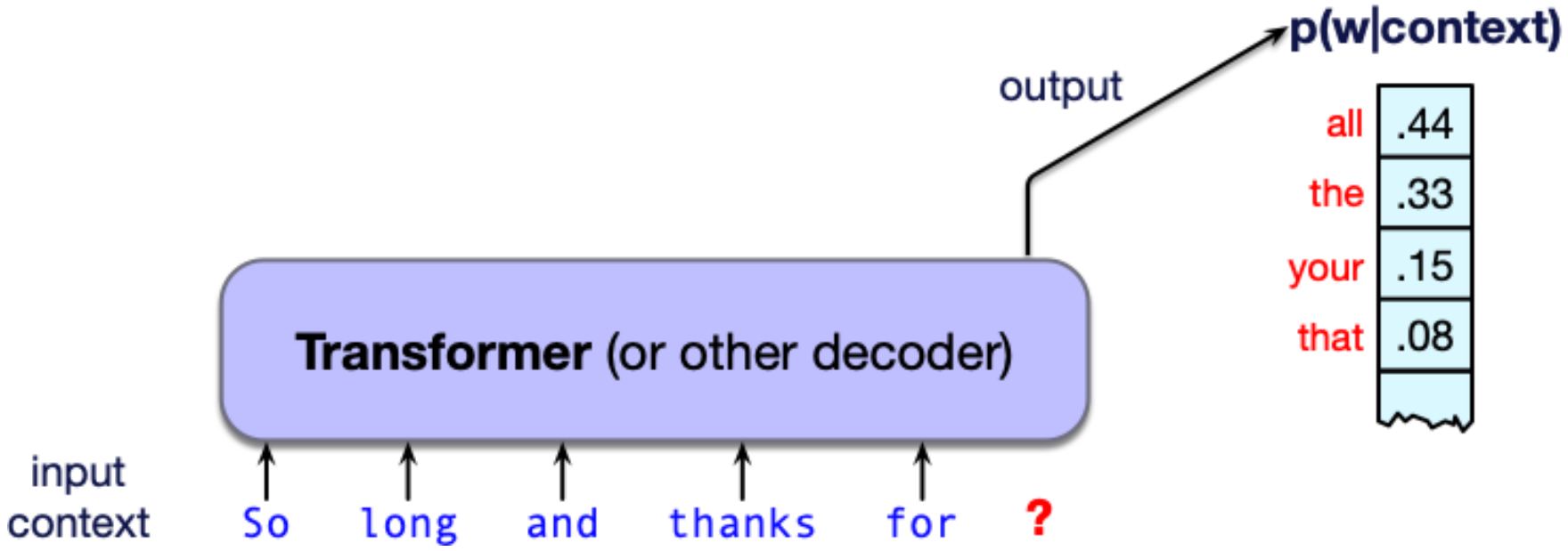
Source: Jurafsky & Martin (2025)

What is a large language model?

A neural network with:

Input: a context or prefix,

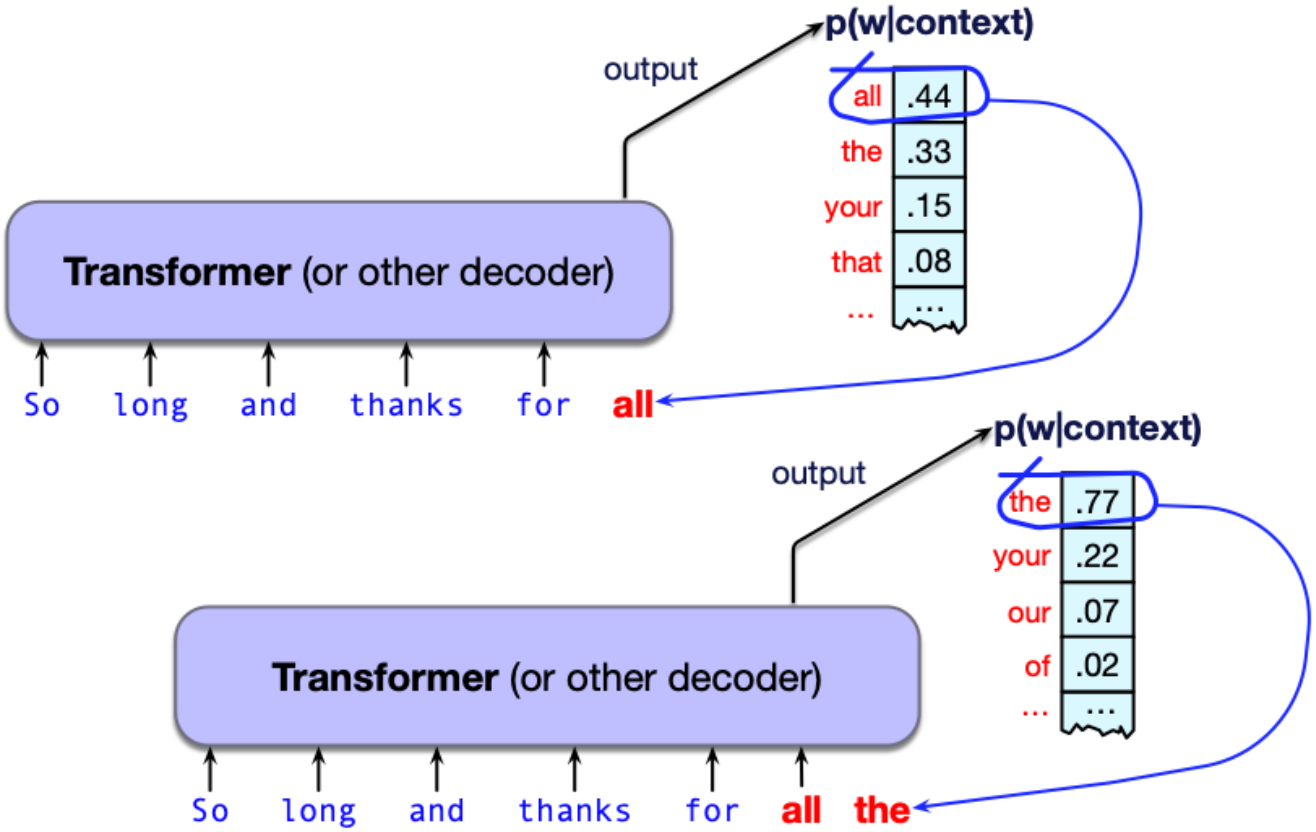
Output: a distribution over possible next words



Source: Jurafsky & Martin (2025)

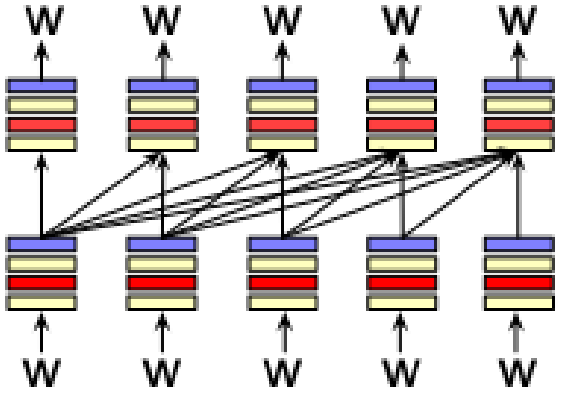
LLMs can generate

A model that gives a probability distribution over next words can generate by repeatedly sampling from the distribution

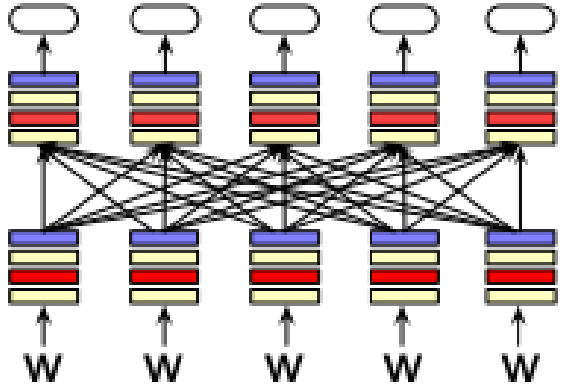


Source: Jurafsky & Martin (2025)

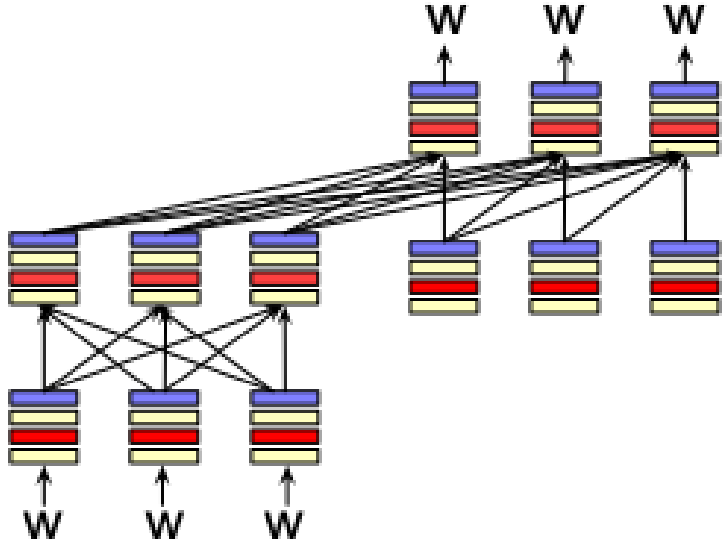
Three architectures for LLMs



Decoders
- GPT, Claude, Llama, Mixtral, Deepseek



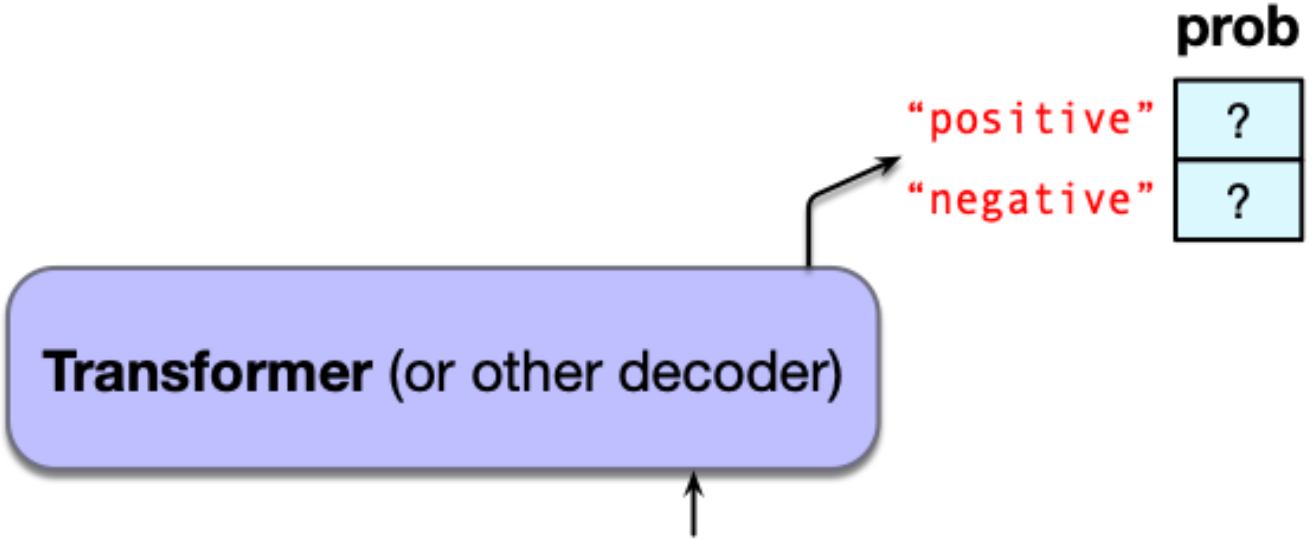
Encoders
- word2vec
- BERT family



Encoder-decoder
- Whisper

Source: Jurafsky & Martin (2025)

Lots of tasks can be turned into tasks of predicting words



$P(\text{positive} \mid \text{The sentiment of the sentence ``I like Jackie Chan" is:})$

$P(\text{negative} \mid \text{The sentiment of the sentence ``I like Jackie Chan" is:})$

Source: Jurafsky & Martin (2025)

Conclusion

- Core idea: “distributional hypothesis”: “you will know a word by the company it keeps”;
- Embeddings → Map each word into a "low"-dimensional vector space
- Or, in English: assign a bunch of numbers to each word, in such a way that “similar” words are closer together;
- **Contextual** embeddings are popular now (transformers/ LLMs), who knows what is next!