

DEEP LEARNING

Lecture 8: Language Model

Dr. Yang Lu

Department of Computer Science and Technology

luyang@xmu.edu.cn



NLP Tasks using Language Model

■ Question answering

Input

Example 1

Question: what color was john wilkes booth's hair

Wikipedia Page: John_Wilkes_Booth

Long answer: Some critics called Booth “the handsomest man in America” and a “natural genius”, and noted his having an “astounding memory”; others were mixed in their estimation of his acting. He stood 5 feet 8 inches (1.73 m) tall, had jet-black hair, and was lean and athletic. Noted Civil War reporter George Alfred Townsend described him as a “muscular, perfect man” with “curling hair, like a Corinthian capital”.

Short answer: jet-black

Example 2

Question: can you make and receive calls in airplane mode

Wikipedia Page: Airplane_mode

Long answer: Airplane mode, aeroplane mode, flight mode, offline mode, or standalone mode is a setting available on many smartphones, portable computers, and other electronic devices that, when activated, suspends radio-frequency signal transmission by the device, thereby disabling Bluetooth, telephony, and Wi-Fi. GPS may or may not be disabled, because it does not involve transmitting radio waves.

Short answer: BOOLEAN:NO

Prediction



NLP Tasks using Language Model

■ Textual Entailment (TE) or Natural Language Inference (NLI)

Relationship	Premise & Hypothesis
Entailment	<p>Premise: This church choir sings to the masses as they sing joyous songs from the book at a church.</p> <p>Hypothesis: The church is filled with song.</p>
Neutral	<p>Premise: This church choir sings to the masses as they sing joyous songs from the book at a church.</p> <p>Hypothesis: The church has cracks in the ceiling.</p>
Contradict	<p>Premise: This church choir sings to the masses as they sing joyous songs from the book at a church.</p> <p>Hypothesis: A choir singing at a baseball game.</p>



NLP Tasks using Language Model

■ Sentiment analysis

GT: 4 Prediction: 4

pork belly = delicious .
scallops ?
i do n't .
even .
like .
scallops , and these were a-m-a-z-i-n-g .
fun and tasty cocktails .
next time i 'm in phoenix , i will go
back here .
highly recommend .

GT: 0 Prediction: 0

terrible value .
ordered pasta entree .
.
\$ 16.95 good taste but size was an
appetizer size .
.
no salad , no bread no vegetable .
this was .
our and tasty cocktails .
our second visit .
i will not go back .

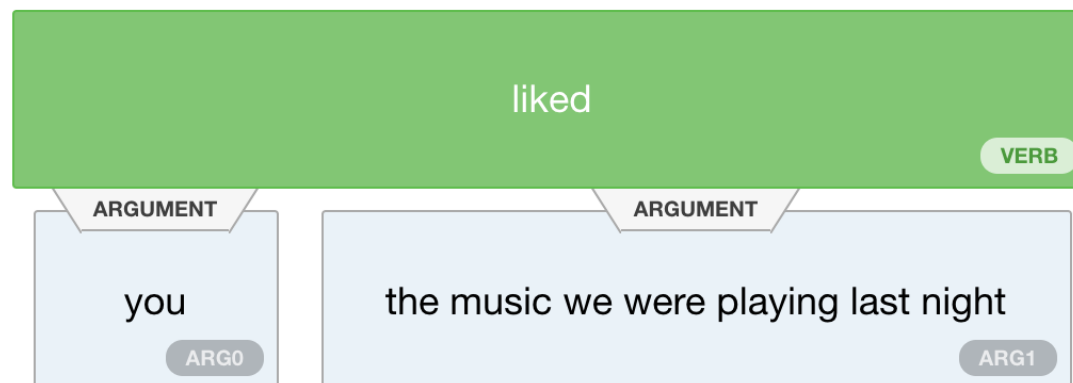


NLP Tasks using Language Model

■ Semantic role labeling

< > Verb 1 of 4: liked

If you liked the music we were playing last night , you will absolutely love what we 're playing tomorrow !

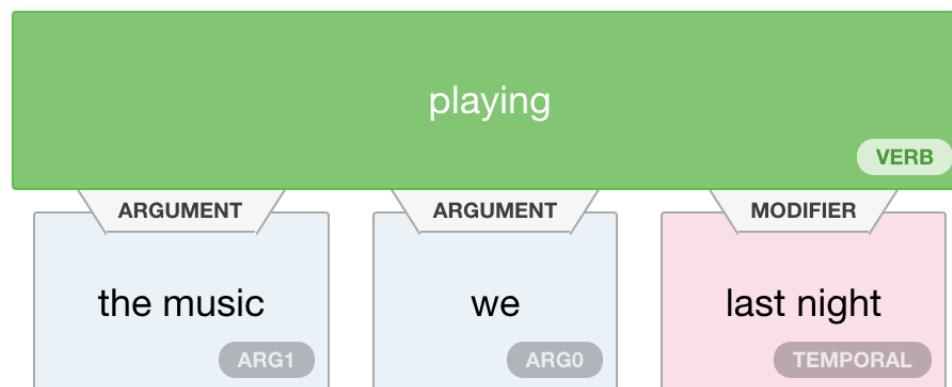


NLP Tasks using Language Model

■ Semantic role labeling

< > Verb 2 of 4: playing

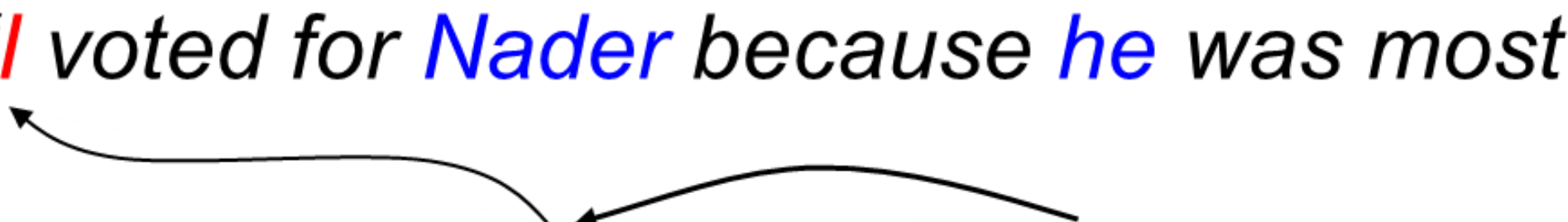
If you liked the music we were **playing** last night , you will absolutely love what we 're playing tomorrow !



NLP Tasks using Language Model

- Coreference resolution

“I voted for Nader because he was most aligned with my values,” she said.



The diagram shows three curved arrows indicating coreference resolution. One arrow points from the word 'Nader' in the first sentence to the word 'he' in the same sentence. A second arrow points from the word 'she' in the second sentence to the word 'I' in the same sentence. A third arrow points from the word 'she' in the second sentence to the word 'I' in the first sentence.



NLP Tasks using Language Model

Named entity recognition (NER)

contentSkip to site indexPoliticsSubscribeLog InSubscribeLog InToday's **PaperAdvertisementSupported** **ORG** by **F.B.I. Agent** **Peter Strzok** **PERSON** ,
Who Criticized Trump **PERSON** in Texts, Is FiredImagePeter Strzok, a top **F.B.I.** **GPE** counterintelligence agent who was taken off the special counsel
investigation after his disparaging texts about President **Trump** **PERSON** were uncovered, was fired. **CreditT.J. Kirkpatrick** **PERSON** for **The New York**
TimesBy Adam Goldman **ORG** and **Michael S. SchmidtAug** **PERSON** . **13** **CARDINAL** , **2018WASHINGTON** **CARDINAL** — **Peter Strzok**
PERSON , the **F.B.I.** **GPE** senior counterintelligence agent who disparaged President **Trump** **PERSON** in inflammatory text messages and helped
oversee the **Hillary Clinton** **PERSON** email and **Russia** **GPE** investigations, has been fired for violating bureau policies, Mr. **Strzok** **PERSON** 's lawyer
said **Monday** **DATE** .Mr. Trump and his allies seized on the texts — exchanged during the **2016** **DATE** campaign with a former **F.B.I.** **GPE** lawyer,
Lisa Page — in **PERSON** assailing the **Russia** **GPE** investigation as an illegitimate “witch hunt.” Mr. **Strzok** **PERSON** , who rose over **20** **years**
DATE at the **F.B.I.** **GPE** to become one of its most experienced counterintelligence agents, was a key figure in **the early months** **DATE** of the
inquiry.Along with writing the texts, Mr. **Strzok** **PERSON** was accused of sending a highly sensitive search warrant to his personal email account.The
F.B.I. **GPE** had been under immense political pressure by Mr. **Trump** **PERSON** to dismiss Mr. **Strzok** **PERSON** , who was removed **last summer**
DATE from the staff of the special counsel, **Robert S. Mueller III** **PERSON** . The president has repeatedly denounced Mr. **Strzok** **PERSON** in posts on

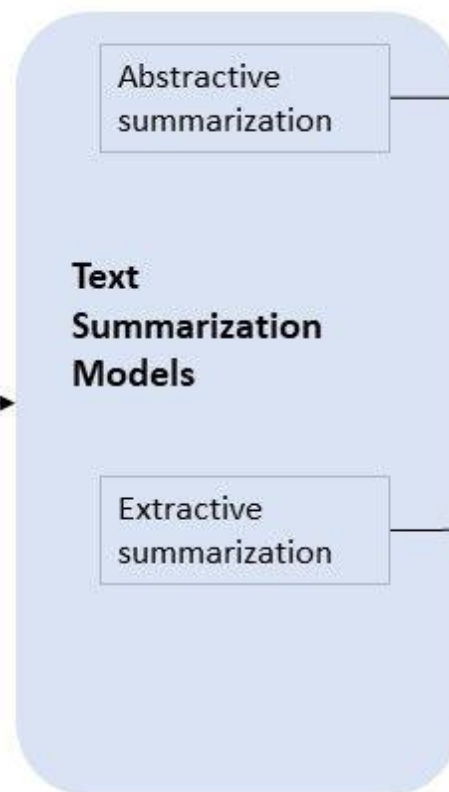


NLP Tasks using Language Model

■ Text summarization

Input Article

Marseille, France (CNN) The French prosecutor leading an investigation into the crash of Germanwings Flight 9525 insisted Wednesday that he was not aware of any video footage from on board the plane. Marseille prosecutor Brice Robin told CNN that " so far no videos were used in the crash investigation . " He added, " A person who has such a video needs to immediately give it to the investigators . " Robin\'s comments follow claims by two magazines, German daily Bild and French Paris Match, of a cell phone video showing the harrowing final seconds from on board Germanwings Flight 9525 as it crashed into the French Alps . All 150 on board were killed. Paris Match and Bild reported that the video was recovered from a phone at the wreckage site. ...



Generated summary

Prosecutor : " So far no videos were used in the crash investigation "

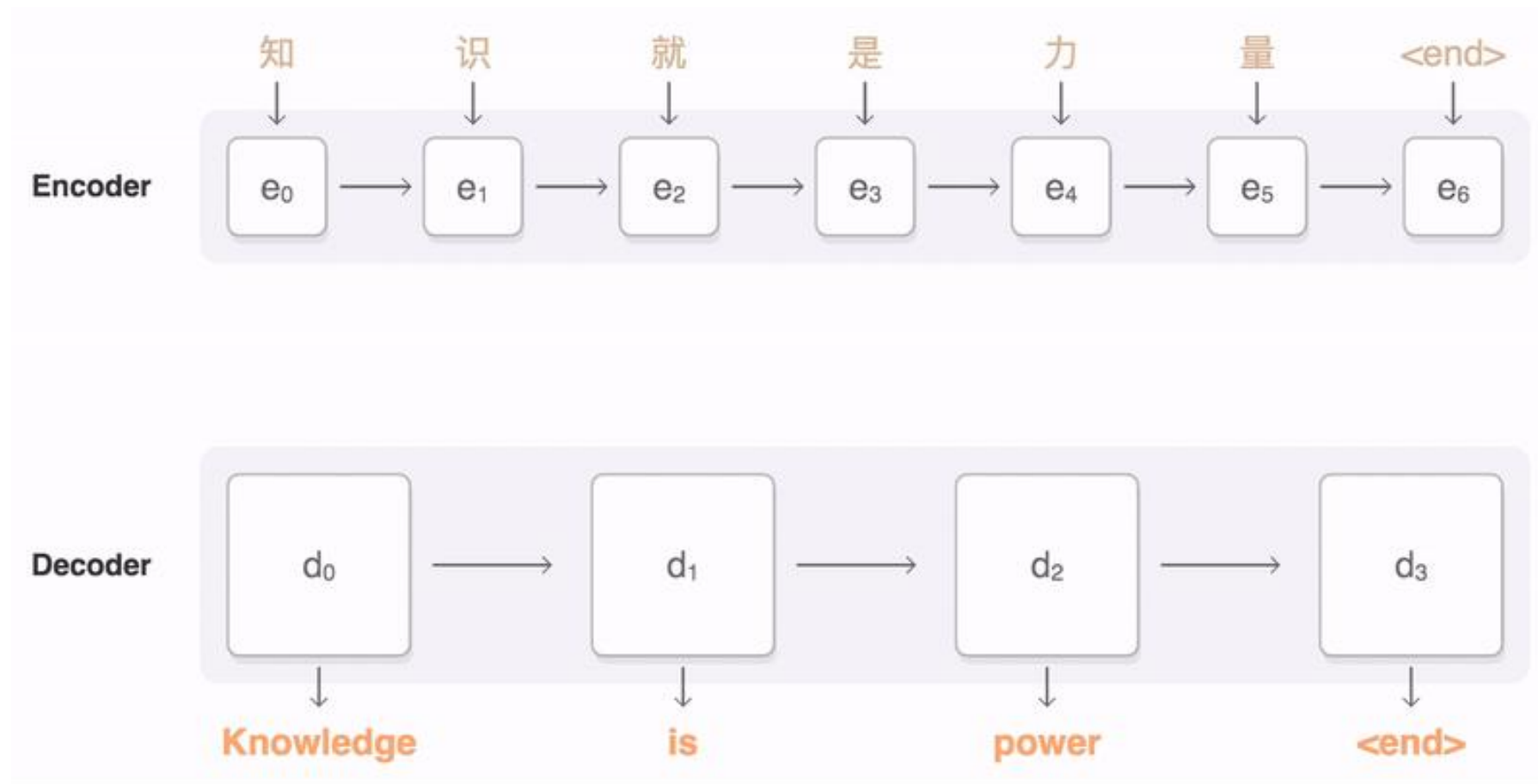
Extractive summary

marseille prosecutor brice robin told cnn that " so far no videos were used in the crash investigation . " robin \'s comments follow claims by two magazines , german daily bild and french paris match , of a cell phone video showing the harrowing final seconds from on board germanwings flight 9525 as it crashed into the french alps . paris match and bild reported that the video was recovered from a phone at the wreckage site .



NLP Tasks using Language Model

Machine translation



- “NLP is the crown jewel of Artificial Intelligence”.
- It is very hard to make AI understand underlying meaning of human language.
 - Among lots of problems, **ambiguity** is one of NLP’s nightmares.



Outlines

- Word2vec
- Transformer
- BERT
- GPT





WORD2VEC

Meaning of a Word

- How can computer know the meaning of a word?
- Use e.g. WordNet, a thesaurus containing lists of synonym sets and hypernyms (“is a” relationships).

Synonym (同义词) of “good”

```
from nltk.corpus import wordnet as wn
poses = { 'n': 'noun', 'v': 'verb', 's': 'adj (s)', 'a': 'adj', 'r': 'adv' }
for synset in wn.synsets("good"):
    print("{}: {}".format(poses[synset.pos()],
                          ", ".join([l.name() for l in synset.lemmas()])))
```

```
noun: good
noun: good, goodness
noun: good, goodness
noun: commodity, trade_good, good
adj: good
adj (sat): full, good
adj: good
adj (sat): estimable, good, honorable, respectable
adj (sat): beneficial, good
adj (sat): good
adj (sat): good, just, upright
...
adverb: well, good
adverb: thoroughly, soundly, good
```

```
from nltk.corpus import wordnet as wn
panda = wn.synset("panda.n.01")
hyper = lambda s: s.hypernyms()
list(panda.closure(hyper))
```

```
[Synset('procyonid.n.01'),
Synset('carnivore.n.01'),
Synset('placental.n.01'),
Synset('mammal.n.01'),
Synset('vertebrate.n.01'),
Synset('chordate.n.01'),
Synset('animal.n.01'),
Synset('organism.n.01'),
Synset('living_thing.n.01'),
Synset('whole.n.02'),
Synset('object.n.01'),
Synset('physical_entity.n.01'),
Synset('entity.n.01')]
```

Hypernyms (上位词) of “panda”



Meaning of a Word

Problems of using dictionary library:

- Great as a resource but missing slight difference between words.
 - e.g. “proficient” is listed as a synonym for “good”. This is only correct in some contexts.
- Different meanings depending on the **context**.
- Missing new meanings of words, or new created words.
 - e.g., badass, lmao, skr, kiki...
 - Impossible to keep up-to-date!
- Requires human labor to create and adapt.

Meaning of a Word

- In traditional NLP, we regard words as discrete symbols.

- Words can be represented by **one-hot vectors**:

Cat: $[0,0,0,0,0,0,\dots,1,0,0]$

Dog: $[1,0,0,0,0,0,\dots,0,0,0]$

Car: $[0,0,0,0,1,0,\dots,0,0,0]$

- The length of the vector equals to the size of the corpus (e.g. 500,000).
- Problem: the distance between any pair of words are 1, except itself.
 - There is no natural notion of **similarity** for one-hot vectors.
- Solution: learn to encode similarity in the vectors themselves.



Word Vectors

- Build a dense vector for each word, chosen so that it is similar to vectors of words that appear in similar contexts.
 - **Word vectors** are sometimes called **word embeddings** or **word representations**. They are a **distributed representation**.
- Word vectors with small distance have the close meaning.

Cat: [0.1,0.7,0.9,0.1,0.1]

Dog: [0.2,0.7,0.8,0.2,0.1]

Car: [0.9,0.1,0.5,0.6,0.8]

- Usually hundreds of dimensions.
- However, there is **no label to train** these word embeddings in a supervised manner.
 - It is impossible to label the similarity between any two words.

Contextual Information

- Distributional semantics: words that are used and occur in the **same contexts** tend to purport **similar meanings**.
 - “A word is characterized by the company it keeps” was popularized by J. R. Firth, an English linguist, in the 1950s.
- When a word w appears in a text, its context is the set of words that appear nearby (within a fixed-size window).
- Use the many contexts of w to build up a representation of w .

*...government debt problems turning into **banking** crises as happened in 2009...*

*...saying that Europe needs unified **banking** regulation to replace the hodgepodge...*

*...India has just given its **banking** system a shot in the arm...*

“Banking” is represented by its context words



Word2vec

Distributed representations of words and phrases and their compositionality

T Mikolov, I Sutskever, K Chen... - Advances in neural ..., 2013 - proceedings.neurips.cc

... We show how to train **distributed representations** of **words** and phrases with the Skip-gram model and demonstrate that these **representations** exhibit linear structure that makes precise ...

☆ Save 77 Cite Cited by 41093 Related articles All 53 versions ⇔

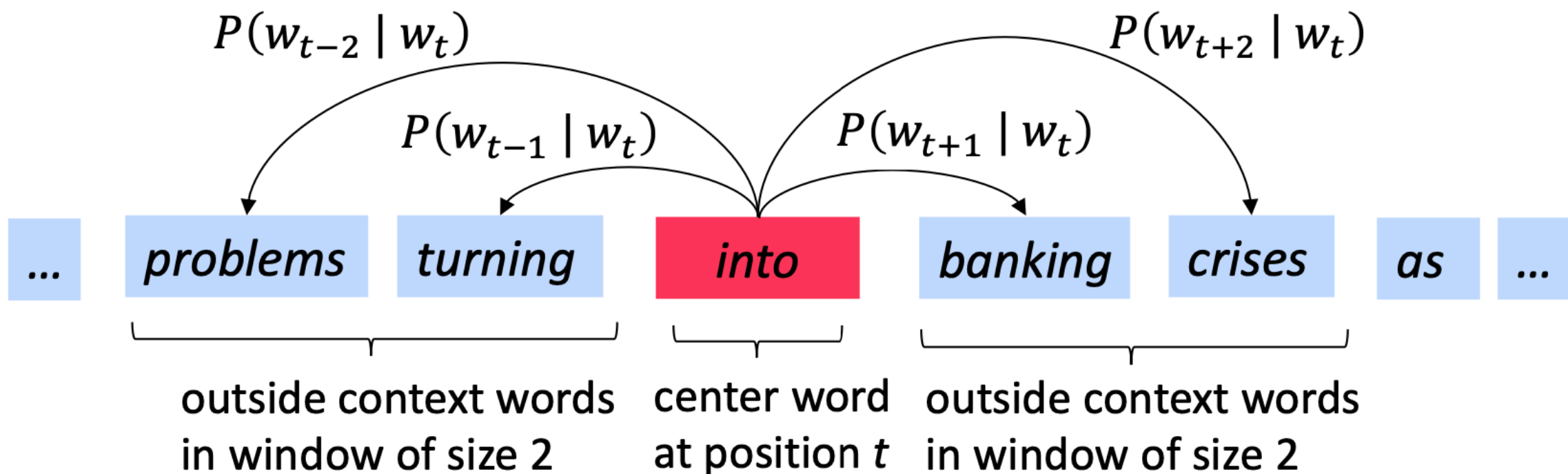
Idea:

- Every word in a fixed vocabulary is represented by a dense vector.
- Go through each position t in the text, which has
 - a **center word** c ,
 - **context words** o .
- Use the similarity of the word vectors for c and o to calculate the probability of c given o (or vice versa).
- Keep adjusting the word vectors to maximize this probability.



Word2vec

- The authors proposed **Skip-gram model** to train word vectors.
- Given the **center word c** , predict the **context words o** .



- The objective function $J(\theta)$ is the negative log likelihood:

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t; \theta).$$

- The probability is calculated by:

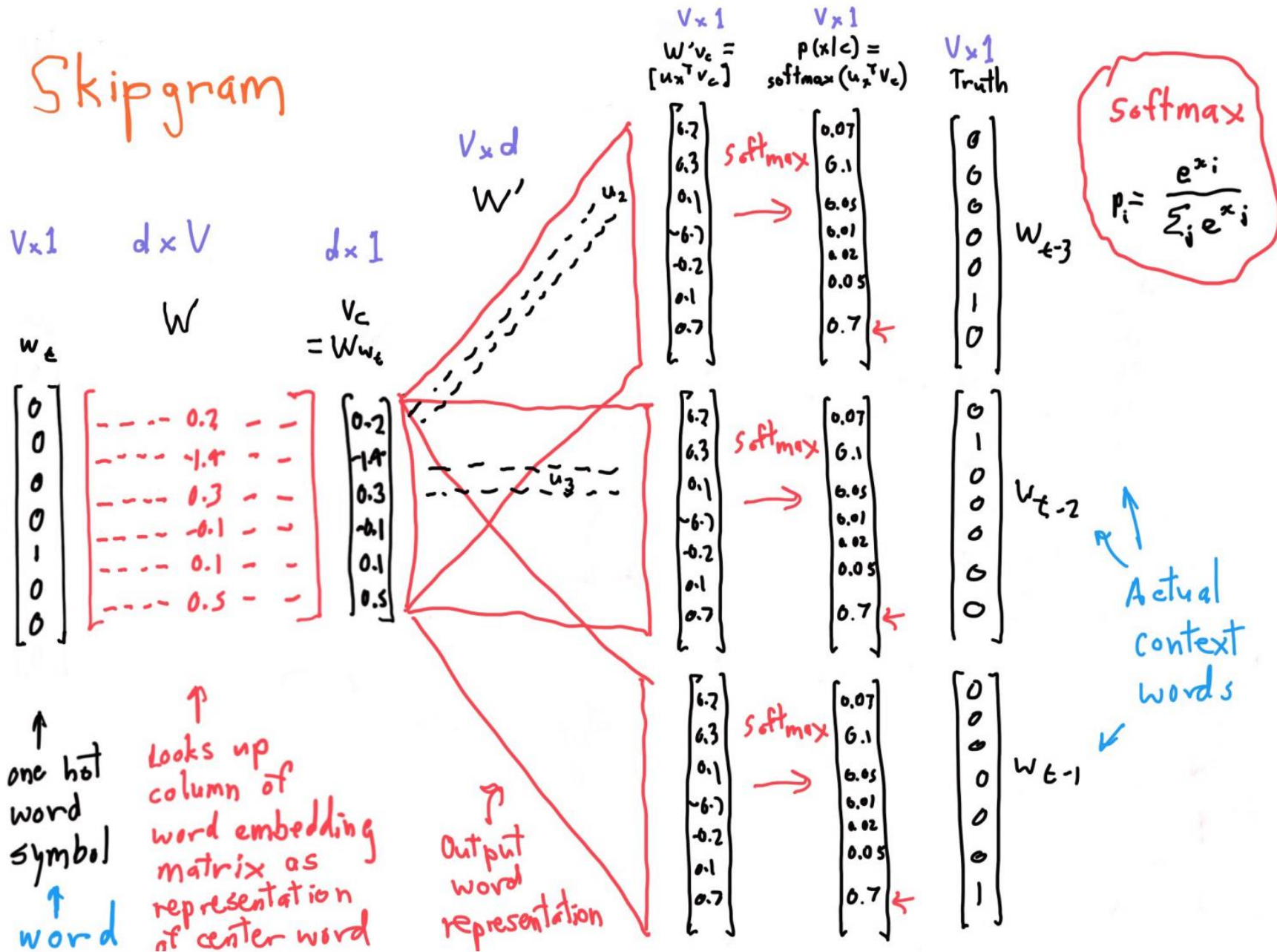
$$p(o|c) = \frac{\exp(\mathbf{w}'_o{}^T \mathbf{w}_c)}{\sum_{u \in V} \exp(\mathbf{w}'_u{}^T \mathbf{w}_c)}.$$

Predict context

Given center

It is nothing but inner product with softmax.

Skipgram



- The learnable representation is called **embedding**.
- What is the difference between embedding and feature/representation?
 - Feature / representation is produced by learnable parameters, but embeddings themselves are learnable parameters.

Negative Sampling

- The probability is calculated by:

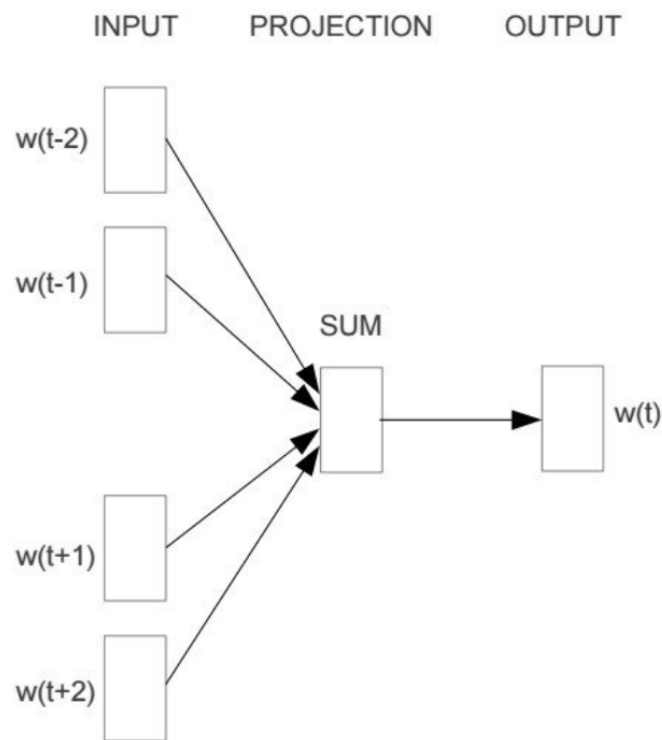
$$p(o|c) = \frac{\exp(\mathbf{w}'_o{}^T \mathbf{w}_c)}{\sum_{u \in V} \exp(\mathbf{w}'_u{}^T \mathbf{w}_c)}.$$

- Every time, we calculate the similarity between word embedding of c and all $u \in V$.
 - It is computational cost is very high.
- We can simply sample a few random samples as the negative samples for training.

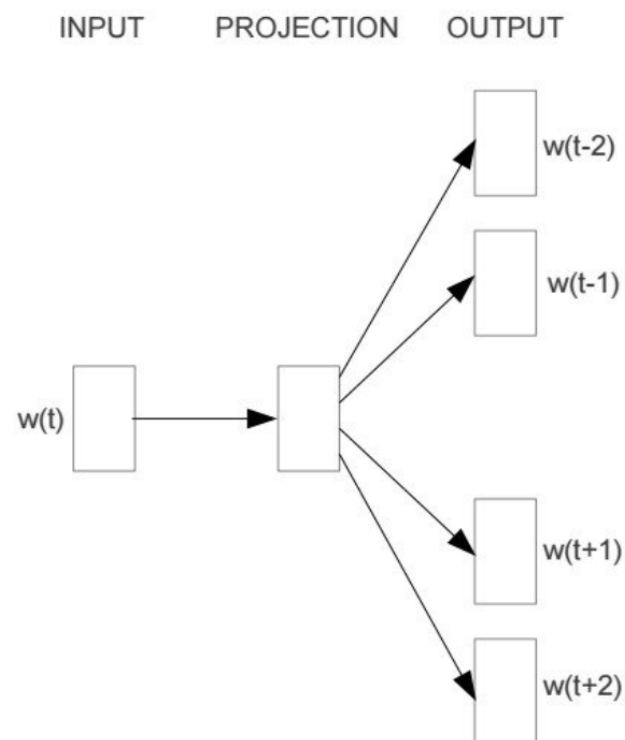


Word2vec

- We can also use the context words to predict the center word. This model is called CBOW (Continuous Bag of Words).



CBOW



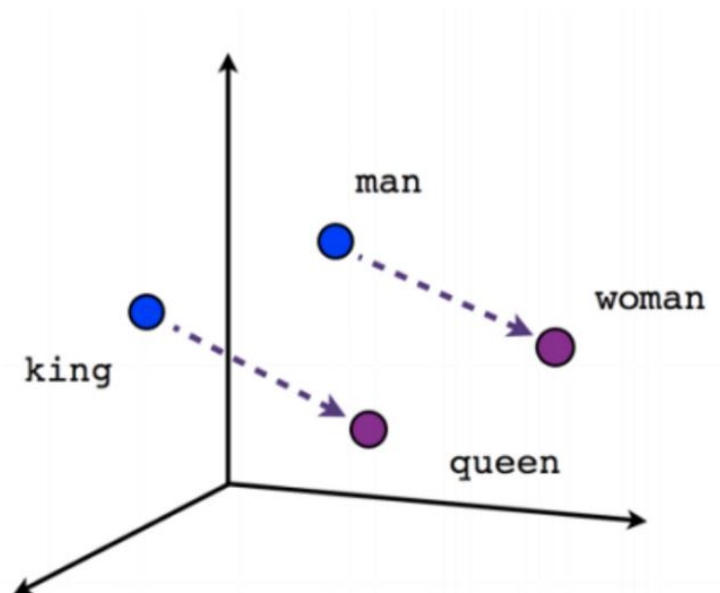
Skip-gram



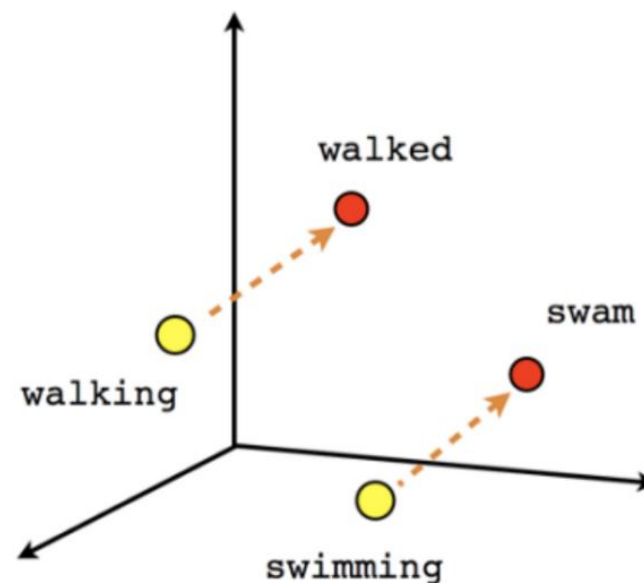
Word Vectors

- By using word vectors, we can “calculate their meaning”:

$$w[\text{'king'}] \approx w[\text{'queen'}] - w[\text{'woman'}] + w[\text{'man'}]$$

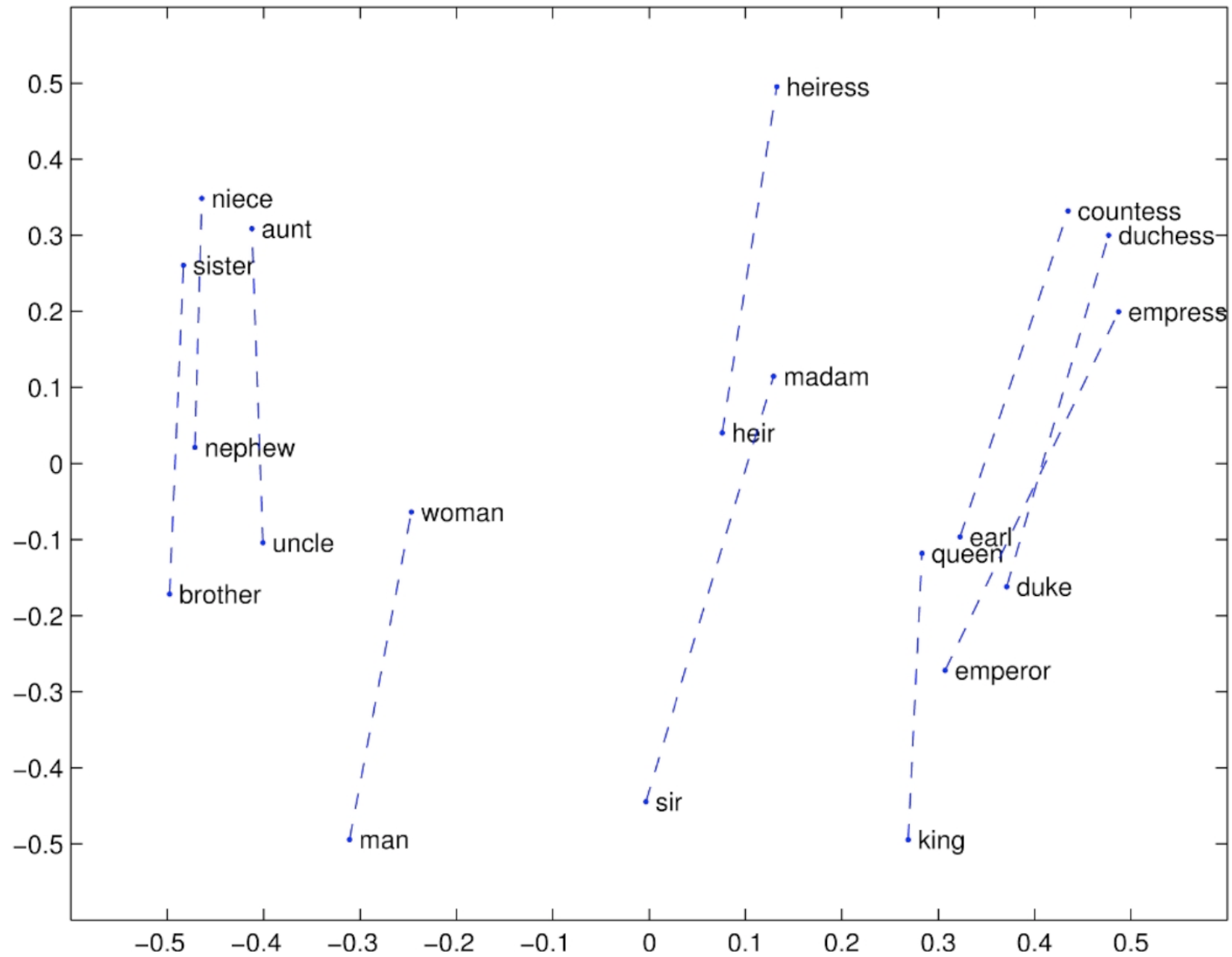


Male-Female



Verb tense





Embedding Projector

DATA

    | Points: 10000 | Dimension: 200



5 tensors found

Word2Vec 10K

Label by

word

Color by

No color map

Edit by

word

Tag selection as

Load

Publish

Download

Label

Sphereize data

Checkpoint: Demo datasets

Metadata: oss_data/word2vec_10000_200d_labels.tsv

UMAP

T-SNE

PCA

CUSTOM

X

Component #1

Y

Component #2

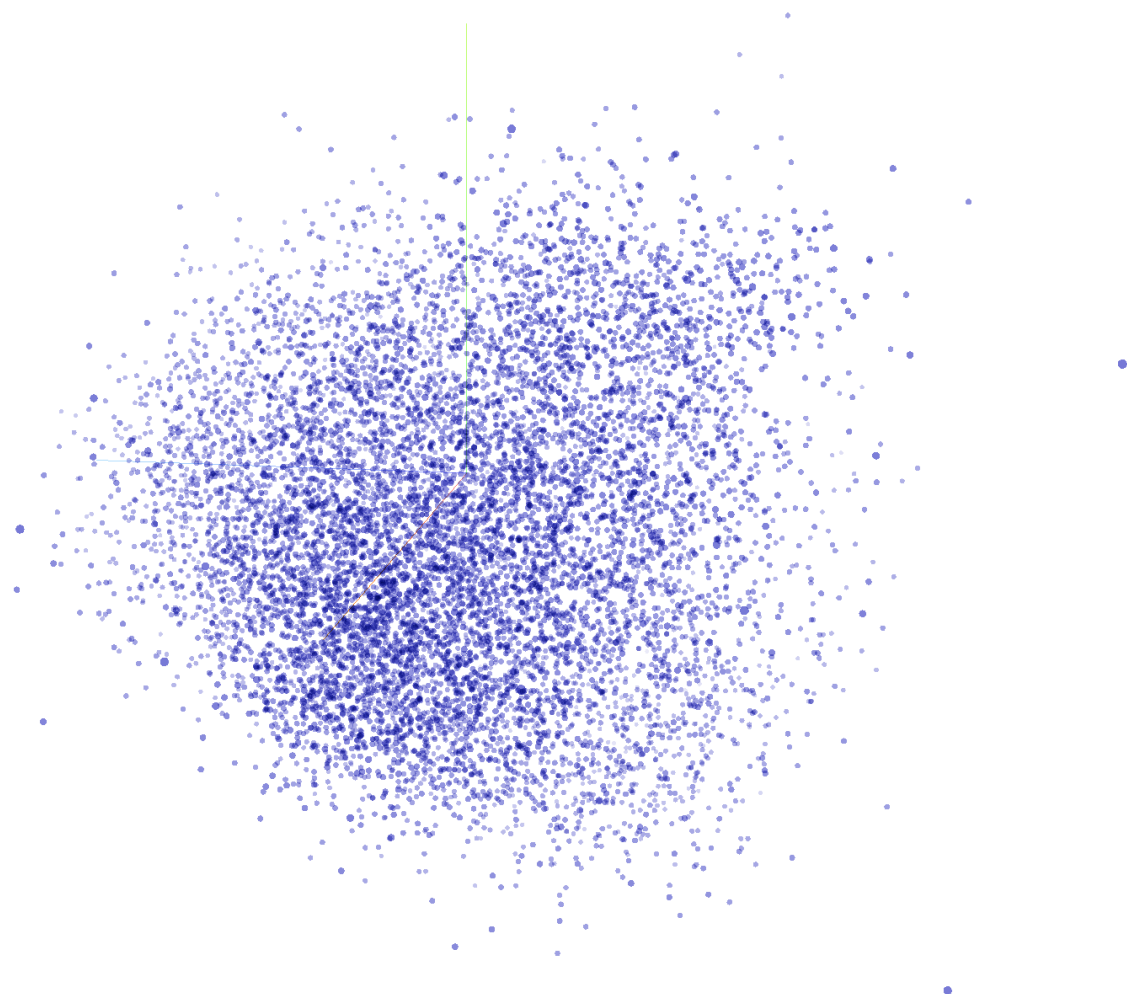
Z

Component #3



PCA is approximate.

Total variance described: 37.5%.



Word2vec

- In essence, Word2vec uses supervised manner to train word vectors.
 - The center word is the input, the context words are its labels.

Source Text	Training Samples			
<table border="1"><tr><td>The</td><td>quick</td><td>brown</td></tr></table> fox jumps over the lazy dog. →	The	quick	brown	(the, quick) (the, brown)
The	quick	brown		
The <table border="1"><tr><td>quick</td><td>brown</td><td>fox</td></tr></table> jumps over the lazy dog. →	quick	brown	fox	(quick, the) (quick, brown) (quick, fox)
quick	brown	fox		
The quick <table border="1"><tr><td>brown</td><td>fox</td><td>jumps</td></tr></table> over the lazy dog. →	brown	fox	jumps	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
brown	fox	jumps		
The quick brown <table border="1"><tr><td>fox</td><td>jumps</td><td>over</td></tr></table> the lazy dog. →	fox	jumps	over	(fox, quick) (fox, brown) (fox, jumps) (fox, over)
fox	jumps	over		

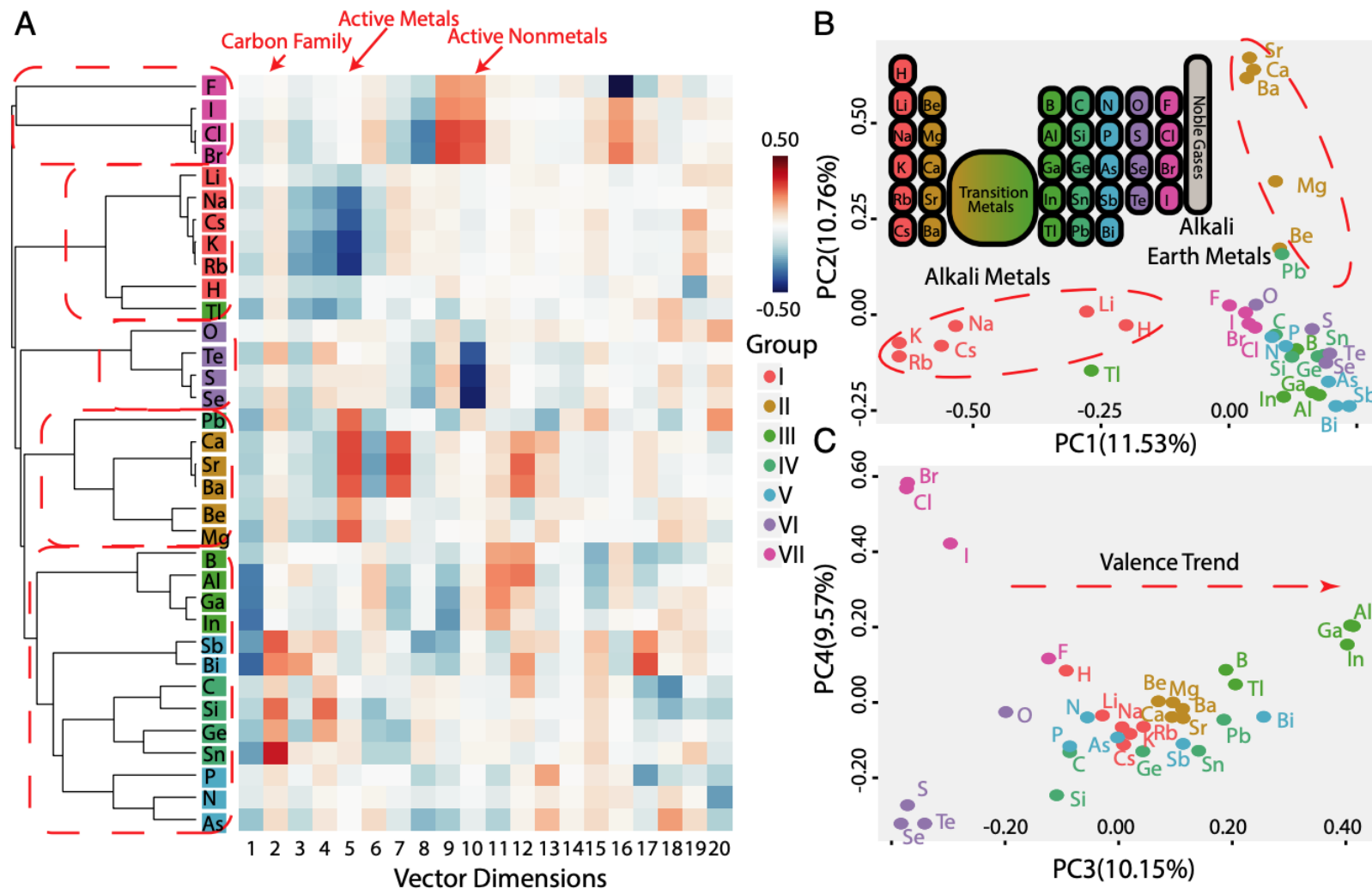


XXX2vec

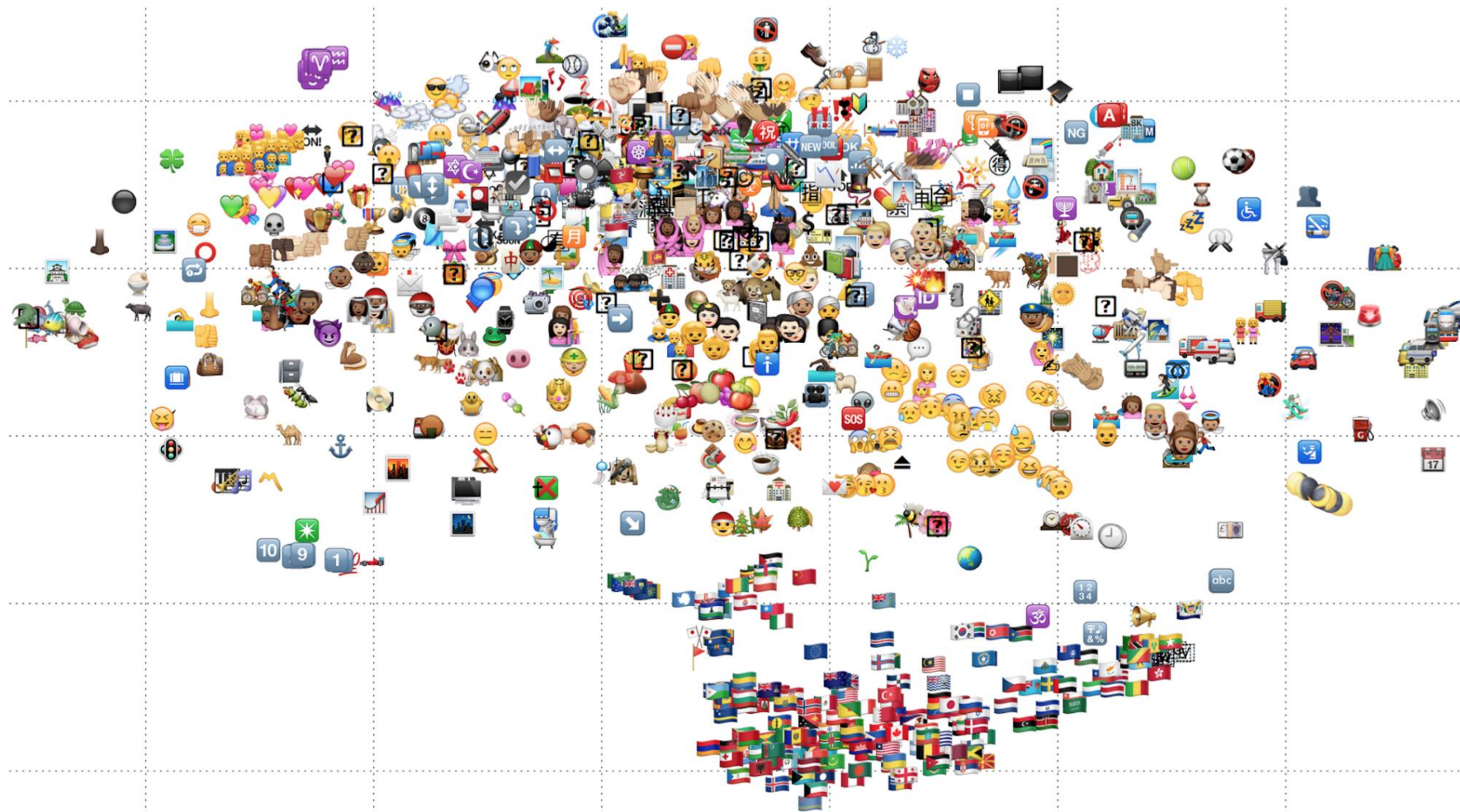
- Follow this idea, any pair frequently occur in a set can be represented by a vector:
 - Recommender system: item2vec, user2vec.
 - Graph: node2vec, edge2vec.
 - Social media: tweet2vec, emoji2vec.
 - Bioinformatics: protein2vec, dna2vec.
 - Chemistry: molecule2vec, atom2vec.
 - Finance: stock2vec, fund2vec, company2vec.
- For more xxx2vec, check [here](#).



Atom2vec



Emoji2vec



廈門大學信息學院 (特色化示范性软件学院)

School of Informatics Xiamen University (National Characteristic Demonstration Software School)



廈門大學 计算机科学与技术系

Department of Computer Science and Technology, Xiamen University

Train Word2vec by PyTorch

- Use `nn.Embedding` for embedding loop-up.

```
word_to_ix = {"hello": 0, "world": 1}
embeds = nn.Embedding(2, 5)  # 2 words in vocab, 5 dimensional embeddings
lookup_tensor = torch.tensor([word_to_ix["hello"]], dtype=torch.long)
hello_embed = embeds(lookup_tensor)
print(hello_embed)

tensor([[ 0.6614,  0.2669,  0.0617,  0.6213, -0.4519]],
        grad_fn=<EmbeddingBackward>)
```



```

EMBEDDING_DIM = 10
# We will use Shakespeare Sonnet 2
test_sentence = ""When forty winters shall besiege thy brow,
And dig deep trenches in thy beauty's field,
Thy youth's proud livery so gazed on now,
Will be a totter'd weed of small worth held:
Then being asked, where all thy beauty lies,
Where all the treasure of thy lusty days;
To say, within thine own deep sunken eyes,
Were an all-eating shame, and thriftless praise.
How much more praise deserv'd thy beauty's use,
If thou couldst answer 'This fair child of mine
Shall sum my count, and make my old excuse,'
Proving his beauty by succession thine!
This were to be new made when thou art old,
And see thy blood warm when thou feel'st it cold.""".split()

# we should tokenize the input, but we will ignore that for now
# build a list of tuples.
# Each tuple is ([ word_i-2, word_i-1, word_i+1, word_i+2 ], target word)

context_tuple_list = [(test_sentence[i + 2],
                       [test_sentence[i], test_sentence[i + 1],
                        test_sentence[i + 3], test_sentence[i + 4]])
                       for i in range(len(test_sentence) - 4)]

# print the first 3, just so you can see what they look like
print(context_tuple_list[:3])

vocab = set(test_sentence)
word_to_ix = {word: i for i, word in enumerate(vocab)}

[('winters', ['When', 'forty', 'shall', 'besiege']), ('shall', ['forty',
'winters', 'besiege', 'thy']), ('besiege', ['winters', 'shall', 'thy', 'b
row,'])]

```

```
class SkipGramLanguageModeler(nn.Module):

    def __init__(self, vocab_size, embedding_dim):
        super(SkipGramLanguageModeler, self).__init__()
        self.embeddings = nn.Embedding(vocab_size, embedding_dim)
        self.linear = nn.Linear(embedding_dim, vocab_size)

    def forward(self, inputs):
        embeds = self.embeddings(inputs).view((1, -1))
        out = self.linear(embeds)
        log_probs = F.log_softmax(out, dim=1)
        return log_probs

losses = []
loss_function = nn.NLLLoss()
model = SkipGramLanguageModeler(len(vocab), EMBEDDING_DIM)
optimizer = optim.SGD(model.parameters(), lr=0.001)
```

```

for epoch in range(10):
    total_loss = 0
    for target, context_list in context_tuple_list:
        for context in context_list:

            # Step 1. Prepare the inputs to be passed to the model (i.e, t
            # into integer indices and wrap them in tensors)
            context_idxs = torch.tensor(word_to_ix[context], dtype=torch.l

            # Step 2. Recall that torch *accumulates* gradients. Before pa
            # new instance, you need to zero out the gradients from the ol
            # instance
            model.zero_grad()

            # Step 3. Run the forward pass, getting log probabilities over
            # words
            log_probs = model(context_idxs)

            # Step 4. Compute your loss function. (Again, Torch wants the
            # word wrapped in a tensor)
            loss = loss_function(log_probs, torch.tensor([word_to_ix[targe

            # Step 5. Do the backward pass and update the gradient
            loss.backward()
            optimizer.step()

            # Get the Python number from a 1-element Tensor by calling ten
            total_loss += loss.item()
    print(total_loss)
    losses.append(total_loss)

```

```

2106.0324614048004
2099.963498353958
2093.969718694687
2088.050463914871
2082.2050607204437
2076.4329063892365
2070.7334401607513
2065.106065750122
2059.5502502918243
2054.065470457077

```

Word Representation

- Originally, we basically had **only one representation** of words:
 - E.g. Word2vec, GloVe, fastText.
- These have two problems:
 - Always the same representation for a word **regardless of the context** in which a word token occurs.
 - We just have one representation for a word, but words have different aspects, including semantics, syntactic behavior, and register /connotations.

Deep contextualized word representations

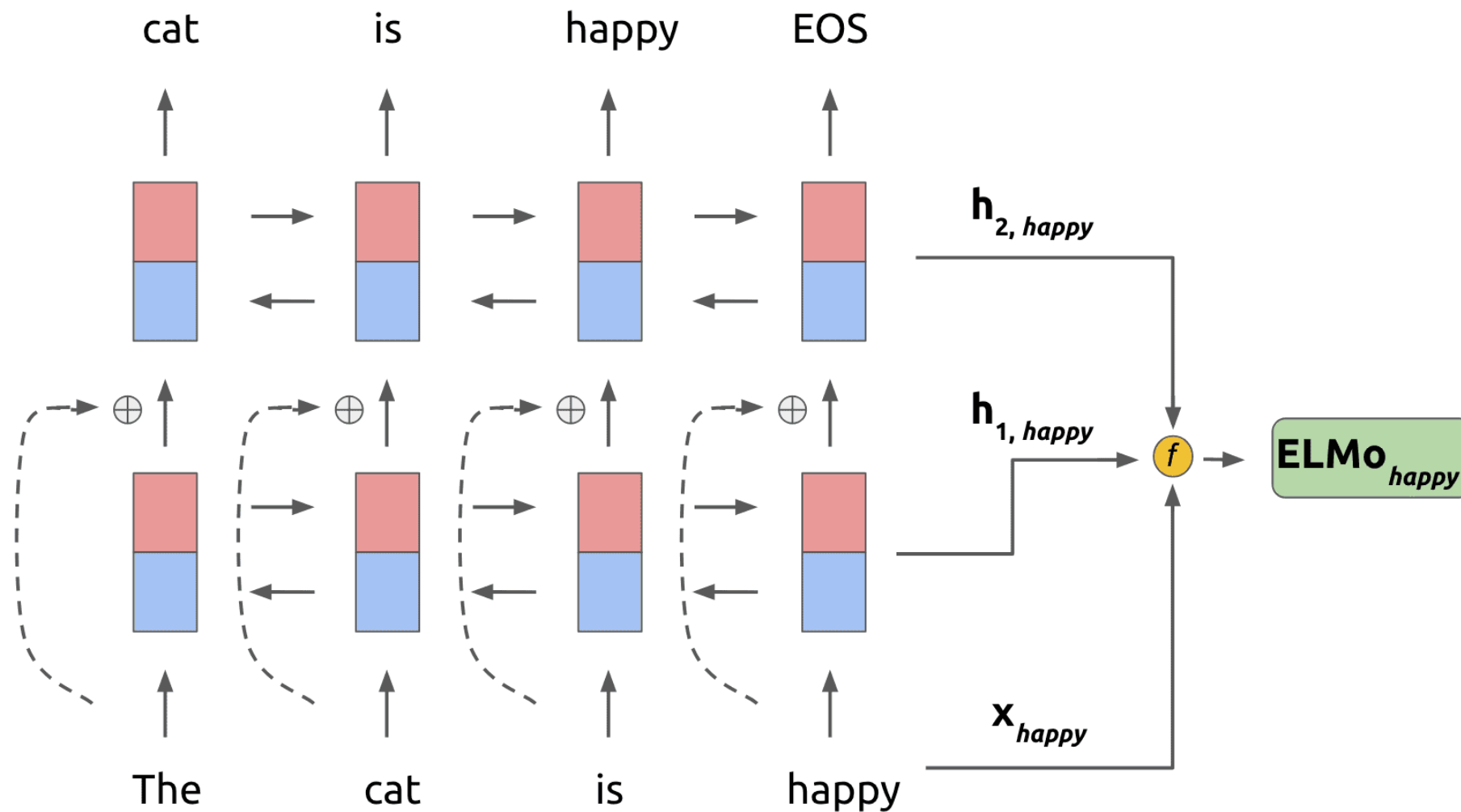
[ME Peters](#), [M Neumann](#), [M Iyyer](#), [M Gardner](#)... - arXiv preprint arXiv ..., 2018 - arxiv.org

We introduce a new type of deep contextualized word representation that models both (1) complex characteristics of word use (eg, syntax and semantics), and (2) how these uses vary across linguistic contexts (ie, to model polysemy). Our word vectors are learned functions of ...

☆  Cited by 4902 [Related articles](#) [All 18 versions](#) 

- Combine pre-trained word token vectors or contextual word vectors.
- Learn word token vectors using long contexts not context windows (here, whole sentence, could be longer).
- Learn a deep bidirectional language model (biLM) and use all its layers in prediction.

ELMo



TRANSFORMER



XXX is All You Need

Attention is all you need ????

[A Vaswani, N Shazeer, N Parmar...](#) - Advances in neural ..., 2017 - proceedings.neurips.cc

... to attend to **all** positions in the decoder up to and including that position. **We need** to prevent ... **We** implement this inside of scaled dot-product **attention** by masking out (setting to $-\infty$) ...

☆ Save 77 Cite Cited by 94235 Related articles All 62 versions 88

[PDF] Attention is all you need

[A Vaswani, N Shazeer, N Parmar...](#) - Advances in neural ..., 2017 - papers.nips.cc

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks in an encoder and decoder configuration. The best performing such models also connect the encoder and decoder through an attentionm ...

☆ 77 Cited by 30000 Related articles All 28 versions 88

Rezero is all you need: Fast convergence at large depth

[T Bachlechner, BP Majumder, HH Mao...](#) - arXiv preprint arXiv ..., 2020 - arxiv.org

Deep networks often suffer from vanishing or exploding gradients due to inefficient signal propagation, leading to long training times or convergence difficulties. Various architecture designs, sophisticated residual-style networks, and initialization schemes have been shown ...

☆ 77 Cited by 61 Related articles All 3 versions 88

Diversity is all you need: Learning skills without a reward function

[B Eysenbach, A Gupta, J Ibarz, S Levine](#) - arXiv preprint arXiv:1802.06070, 2018 - arxiv.org

Intelligent creatures can explore their environments and learn useful skills without supervision. In this paper, we propose DIAYN ('Diversity is All You Need'), a method for learning useful skills without a reward function. Our proposed method learns skills by ...

☆ 77 Cited by 398 Related articles All 4 versions 88

Hopfield networks is all you need

[H Ramsauer, B Schöfl, J Lehner, P Seidl...](#) - arXiv preprint arXiv ..., 2020 - arxiv.org

We introduce a modern Hopfield network with continuous states and a corresponding update rule. The new Hopfield network can store exponentially (with the dimension of the associative space) many patterns, retrieves the pattern with one update, and has ...

☆ 77 Cited by 70 Related articles All 7 versions 88

Proving the lottery ticket hypothesis: Pruning is all you need

[E Malach, G Yehudai...](#) - International ..., 2020 - proceedings.mlr.press

The lottery ticket hypothesis (Frankle and Carbin, 2018), states that a randomly-initialized network contains a small subnetwork such that, when trained in isolation, can compete with the performance of the original network. We prove an even stronger hypothesis (as was also ...

☆ 77 Cited by 60 Related articles All 4 versions 88

Rethinking few-shot image classification: a good embedding is all you need?

[Y Tian, Y Wang, D Krishnan, JB Tenenbaum...](#) - Computer Vision–ECCV ..., 2020 - Springer

The focus of recent meta-learning research has been on the development of learning algorithms that can quickly adapt to test time tasks with limited data and low computational cost. Few-shot learning is widely used as one of the standard benchmarks in meta-learning ...

☆ 77 Cited by 194 Related articles All 7 versions 88

Image augmentation is all you need: Regularizing deep reinforcement learning from pixels

[I Kostrikov, D Yarats, R Fergus](#) - arXiv preprint arXiv:2004.13649, 2020 - arxiv.org

We propose a simple data augmentation technique that can be applied to standard model-free reinforcement learning algorithms, enabling robust learning directly from pixels without the need for auxiliary losses or pre-training. The approach leverages input perturbations ...

☆ 77 Cited by 130 Related articles All 6 versions 88

15 keypoints is all you need

[M Snower, A Kadav, F Lai...](#) - Proceedings of the IEEE ..., 2020 - openaccess.thecvf.com

Pose-tracking is an important problem that requires identifying unique human pose-instances and matching them temporally across different frames in a video. However, existing pose-tracking methods are unable to accurately model temporal relationships and ...

☆ 77 Cited by 6 Related articles All 5 versions 88

Depthwise convolution is all you need for learning multiple visual domains

[Y Guo, Y Li, L Wang, T Rosing](#) - ... of the AAAI Conference on Artificial ..., 2019 - ojs.aaai.org

There is a growing interest in designing models that can deal with images from different visual domains. If there exists a universal structure in different visual domains that can be captured via a common parameterization, then we can use a single model for all domains ...

☆ 77 Cited by 39 Related articles All 11 versions 88

Diffusion is all you need for learning on surfaces

[N Sharp, S Attaiki, K Crane, M Ovsjanikov](#) - arXiv preprint arXiv ..., 2020 - arxiv.org

We introduce a new approach to deep learning on 3D surfaces such as meshes or point clouds. Our key insight is that a simple learned diffusion layer can spatially share data in a principled manner, replacing operations like convolution and pooling which are complicated ...

☆ 77 Cited by 6 Related articles All 3 versions 88



Transformer

- Recurrent models typically factor computation along the symbol positions of the input and output sequences.
 - i.e. either forward or backward.
- It brings two problems:
 - Preclude parallelization within training examples.
 - Difficult to learn dependencies between distant positions.

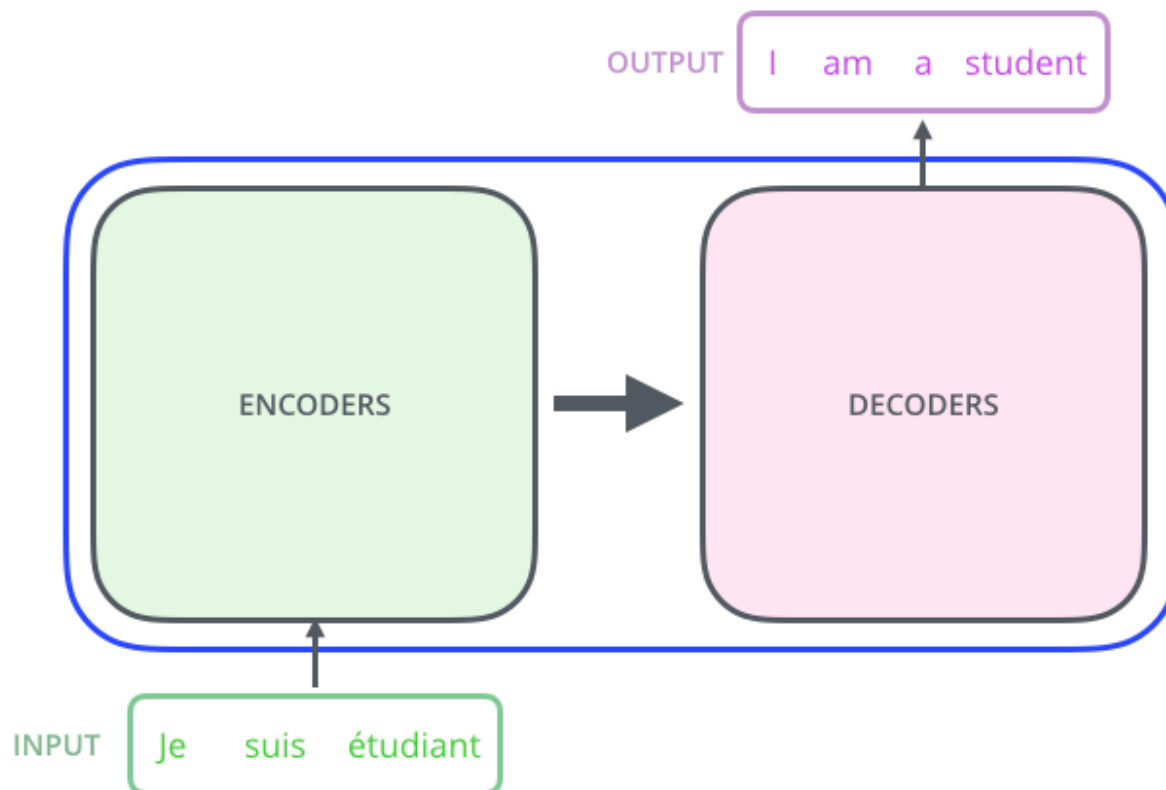
Transformer

- Thoroughly abandoned RNN or CNN architecture.
- **Only use self-attention** and feed forward neural network to model contextual information.
- Designed for machine translation by the encoder-decoder architecture, but now widely used as a basic component of many NLP and CV tasks.



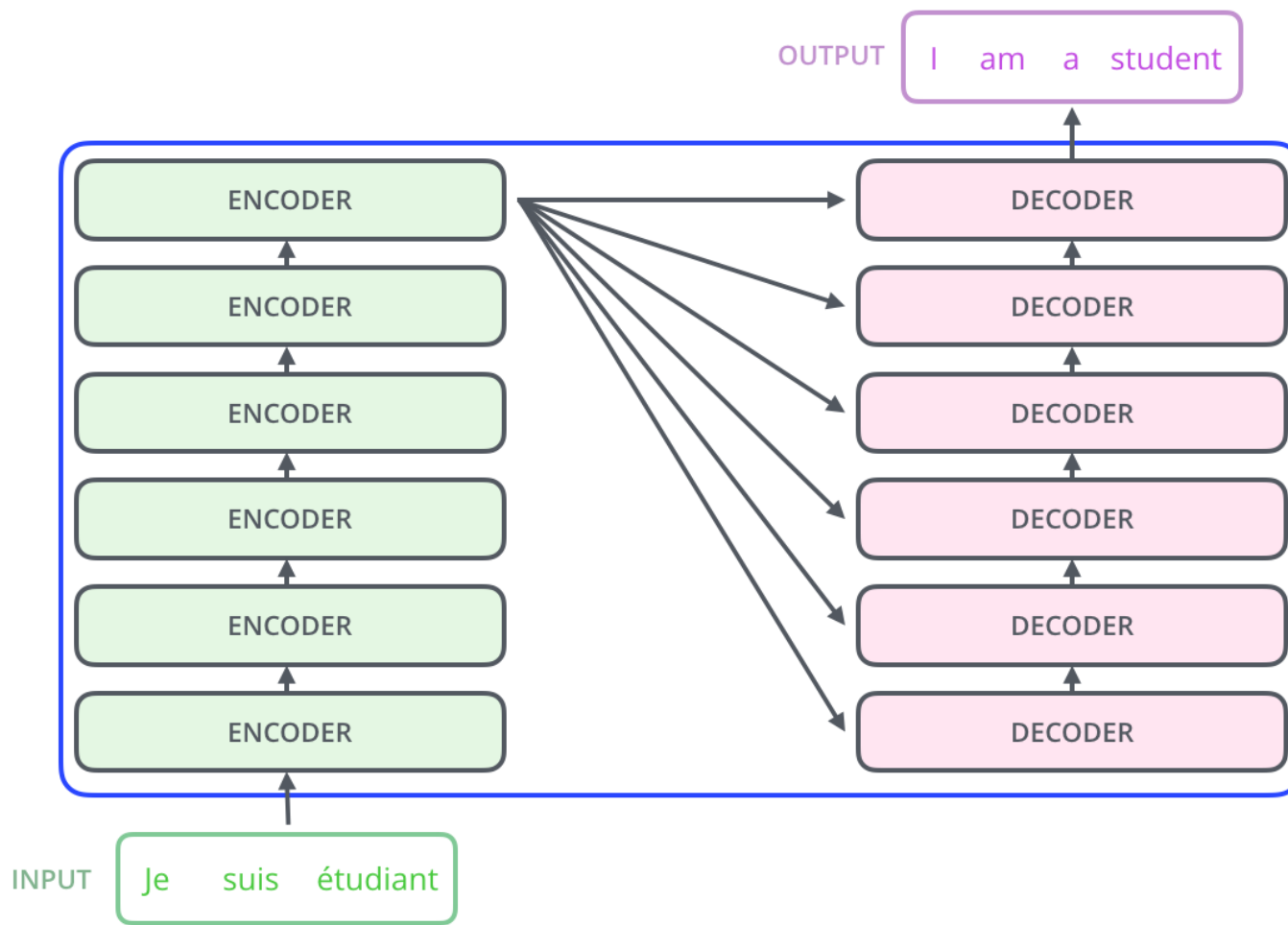
Transformer

- From a high-level look, it is nothing but an encoder-decoder network.



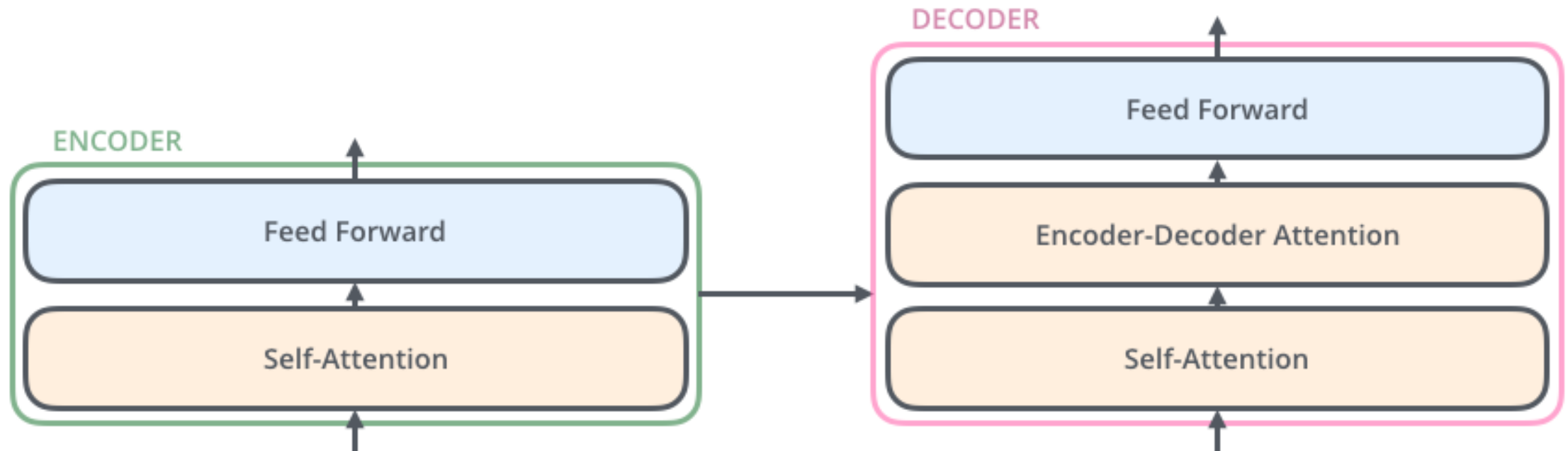
Transformer

- The encoding component is a stack of encoders.
 - In the paper, it is 6.
- The decoding component is a stack of decoders of the same number.



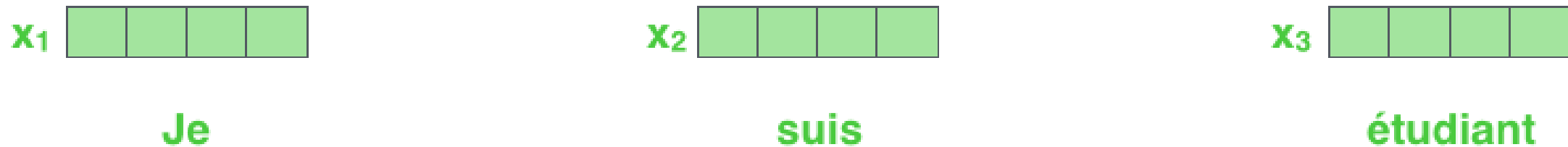
Transformer

- Transformer keeps the encoder-decoder attention, but replace RNN layer by self-attention layer.



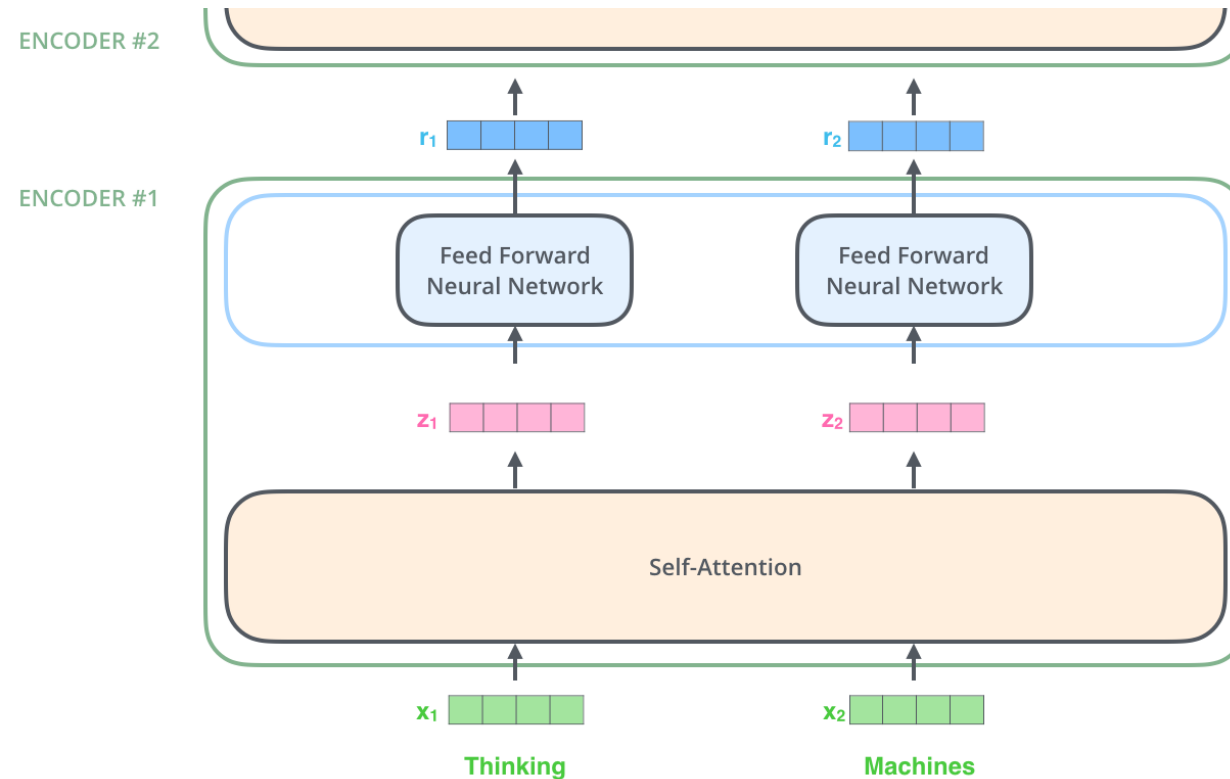
Self-Attention

- As is the case in NLP applications in general, we begin by turning each input word into a vector using an embedding algorithm.
 - e.g. each word is embedded into a vector of size 512.



Self-Attention

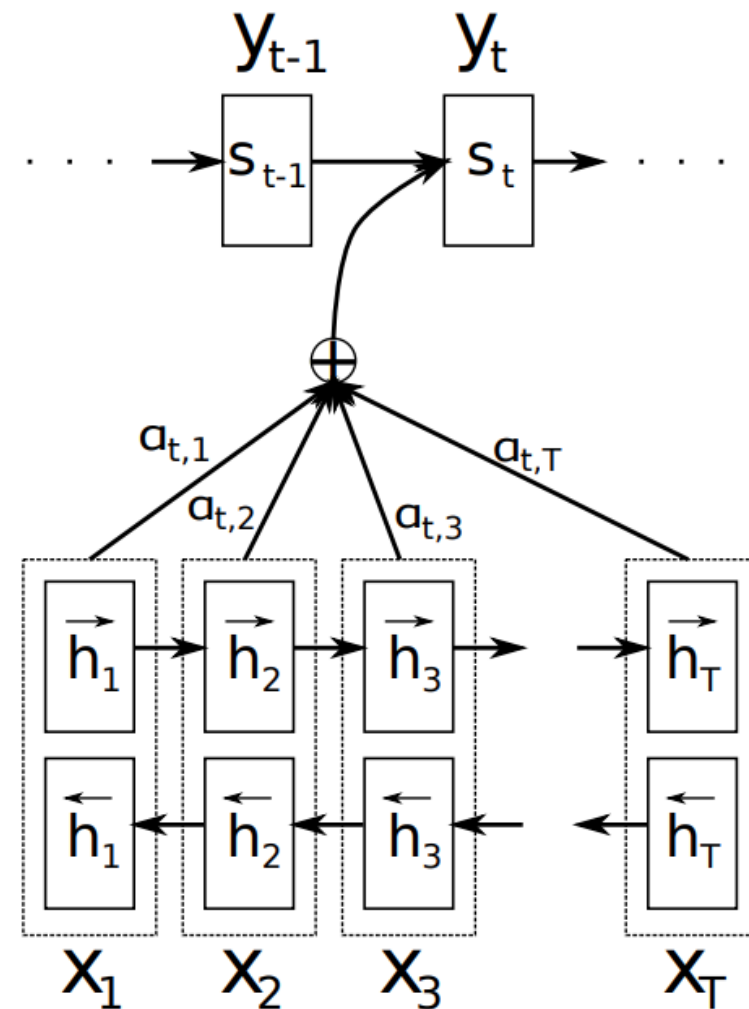
- Each embedding flows through each of the two layers of the encoder.
- There are dependencies between these paths in the self-attention layer.



Attention

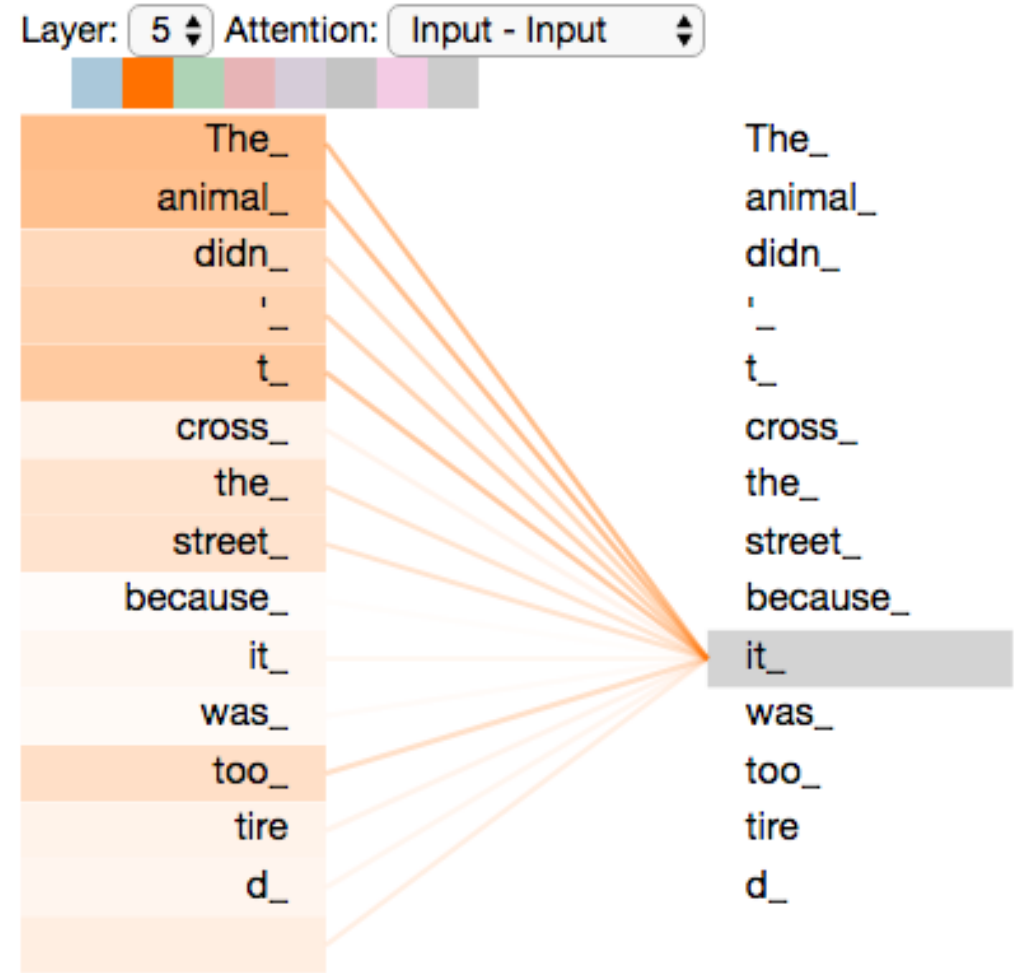
Recall the encoder-decoder attention architecture:

- Use RNN to capture context information.
- Use attention to assign weights from the encoder hidden states to the decoder.



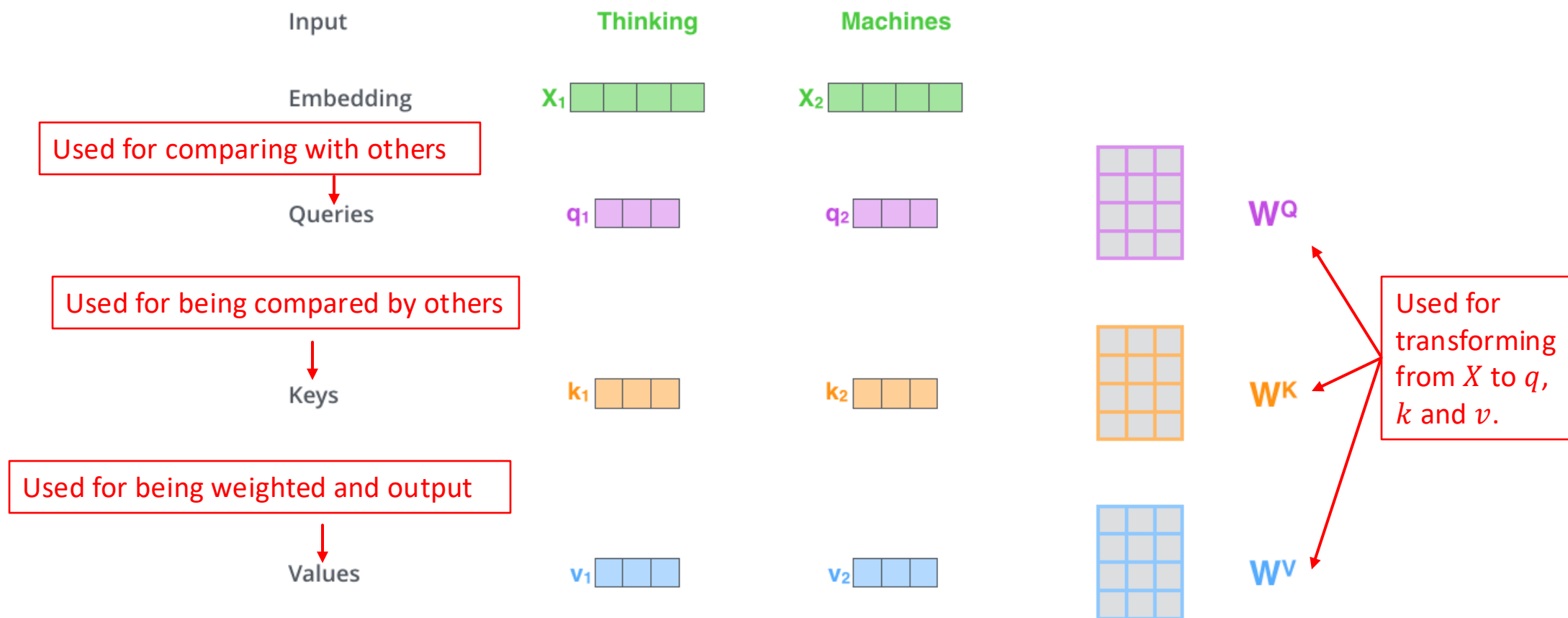
Self-Attention

- Self-attention is the weighted representation of the **target at place of itself**.
- When the model is processing the word “it”, self-attention allows it to associate “it” with “animal”.
- RNN can also do this job, but the correlation highly depends on the distance.



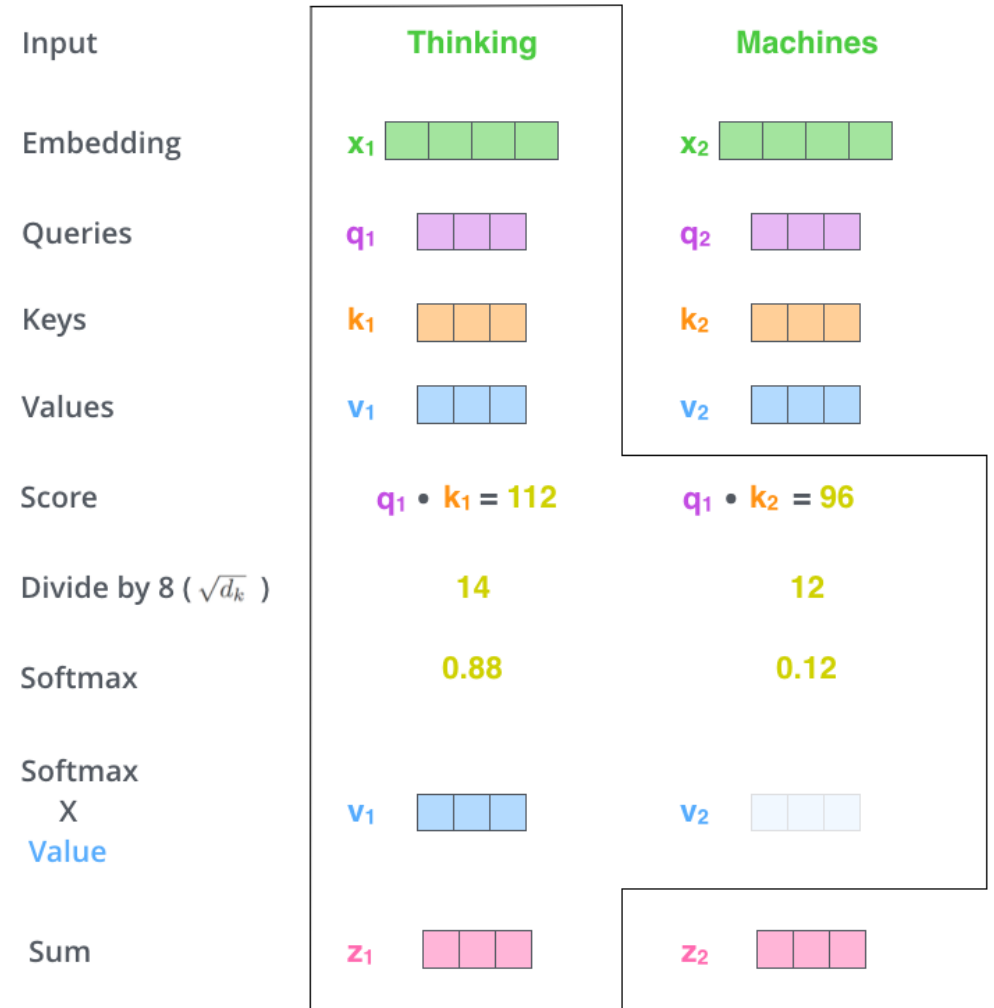
Self-Attention

- So for each word vector, we transform it into a **Query** vector, a **Key** vector, and a **Value** vector.



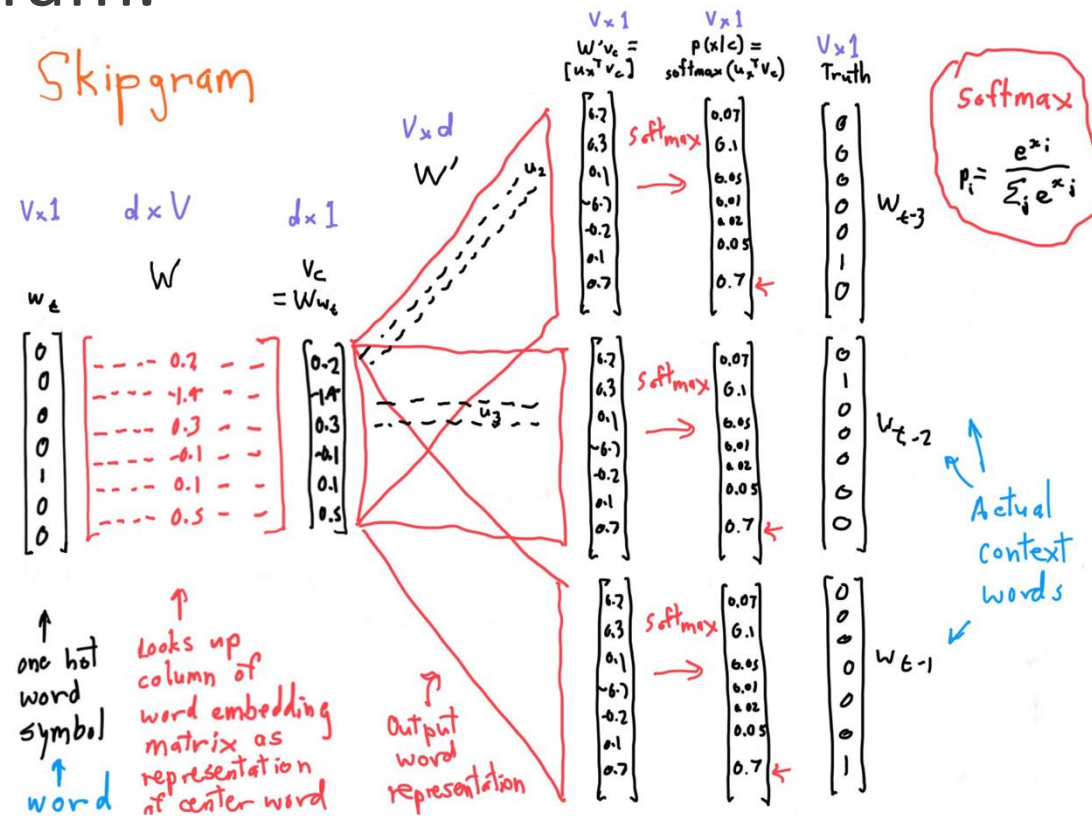
Self-Attention

- Q and K are used to calculate attention weights, and V is used to apply those weights.
- Q is the vector for itself, and K is the vector for others.

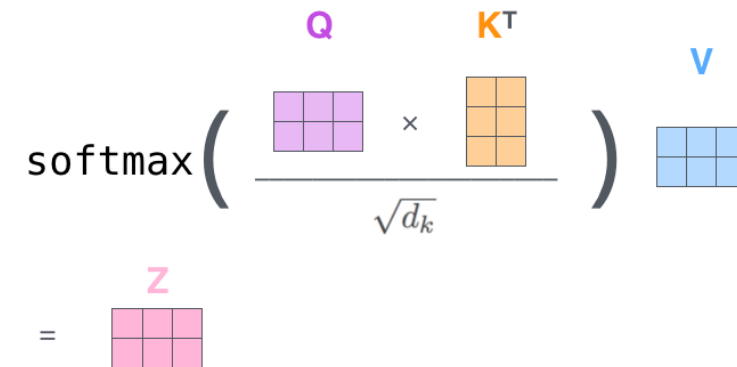


Self-Attention

- Q and K represent central and context, which is similar to W and W' in Skipgram.



Self-Attention



- It is also called dot-product attention:

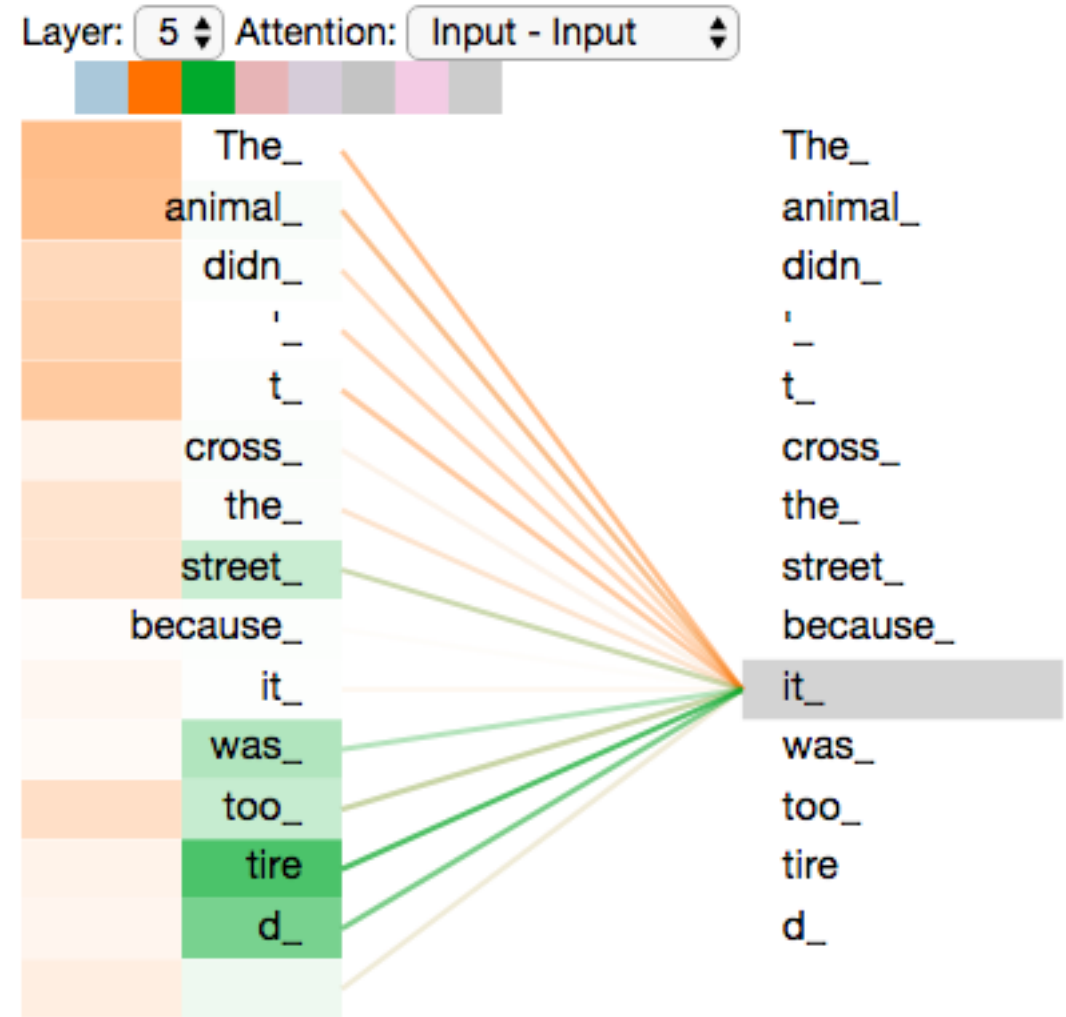
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- When we calculate the self-attention representation, we put all words into matrix:

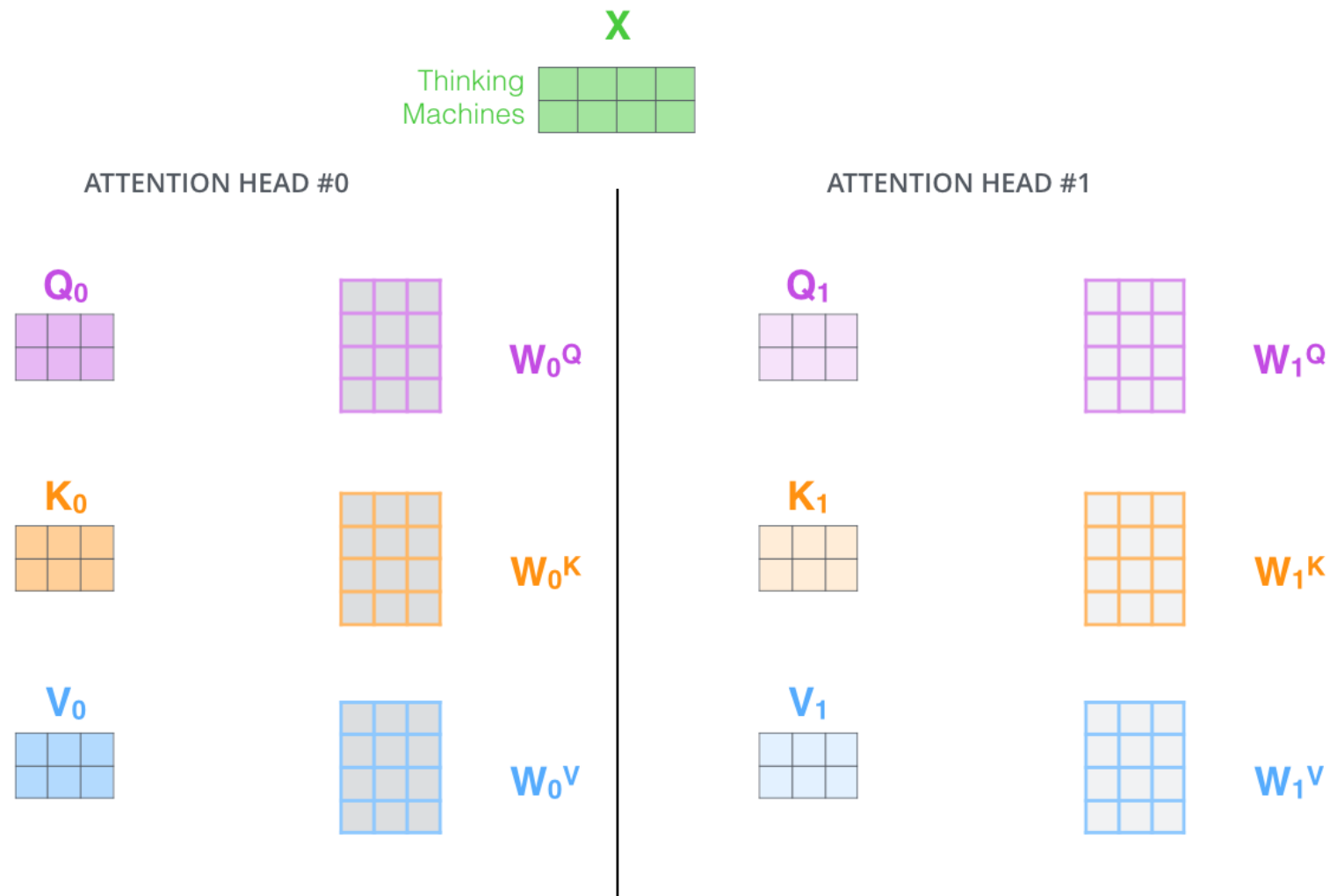


Multi-Head Attention

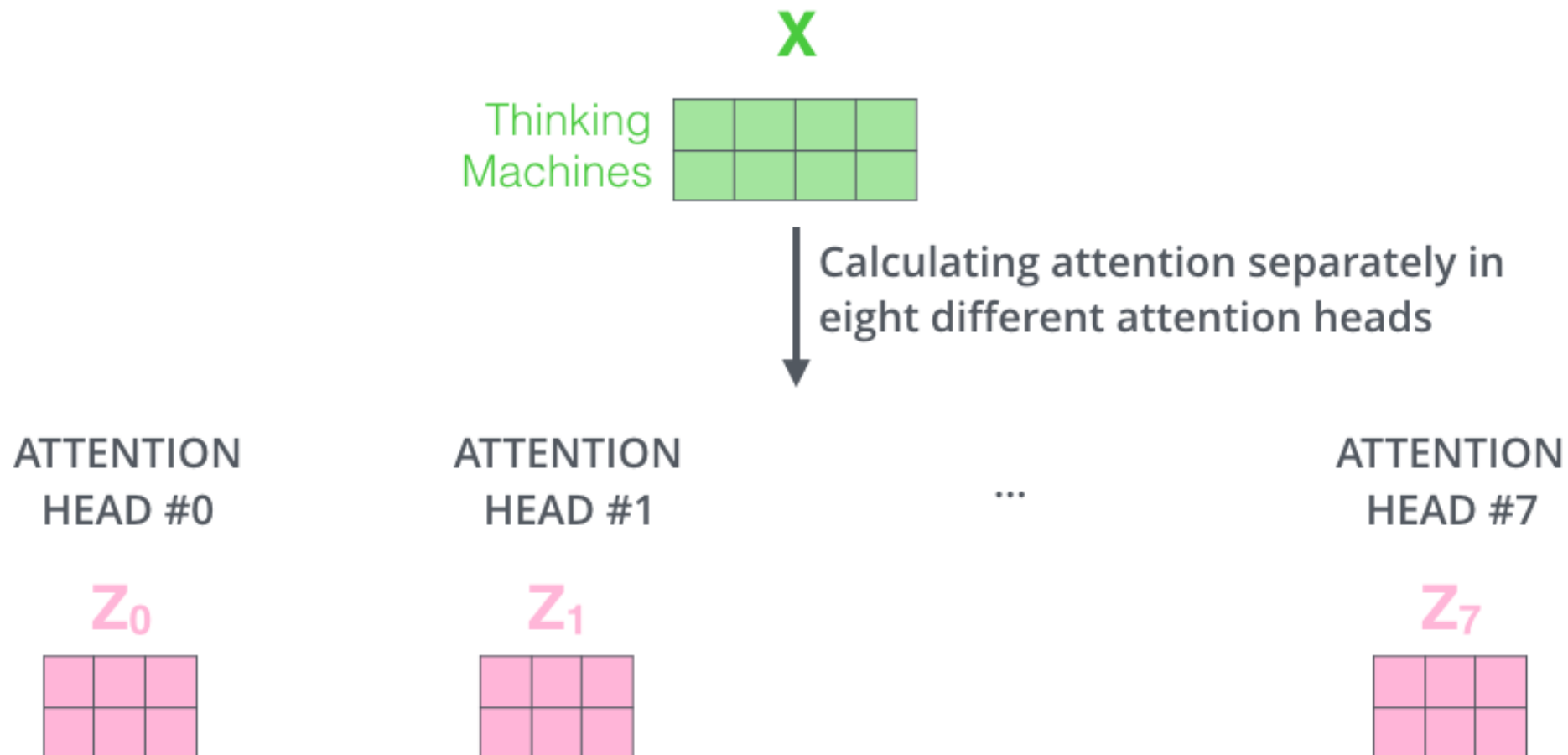
- Multi-head attention expands the model's ability to focus on different positions.
- Each head uses different W^Q , W^K and W^V , which are randomly initialized.
- Different attention heads can be trained in parallel.



Multi-Head Attention



Multi-Head Attention



Multi-Head Attention

1) This is our input sentence*

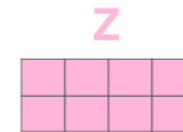
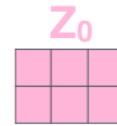
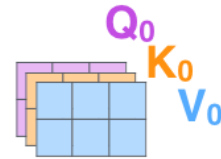
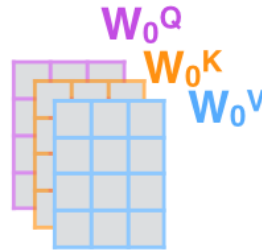
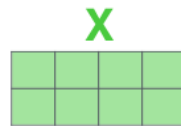
2) We embed each word*

3) Split into 8 heads. We multiply X or R with weight matrices

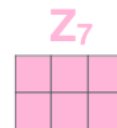
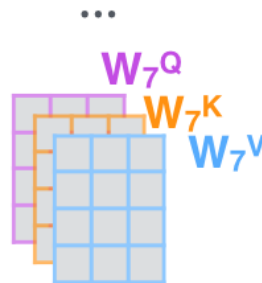
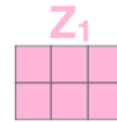
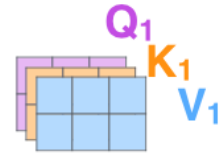
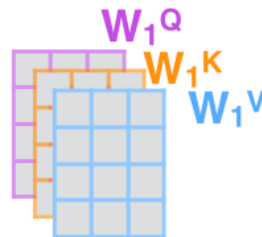
4) Calculate attention using the resulting $Q/K/V$ matrices

5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

Thinking
Machines

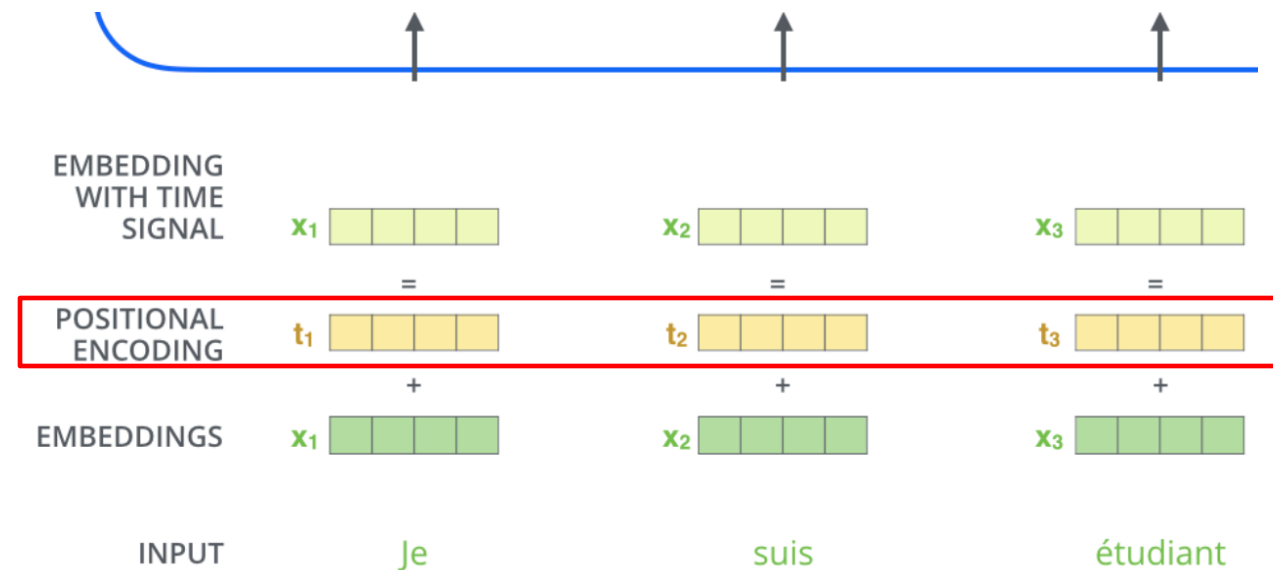


* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one



Positional Encoding

- Now, one problem is that we lose the information about the relative or absolute position of the tokens in the sequence.
 - He likes this movie because it doesn't have an overhead history. -> **Positive**.
 - He doesn't like this movie because it has an overhead history. -> **Negative**.
- Positional encoding helps the model determine the position of each word, or the distance between different words in the sequence.



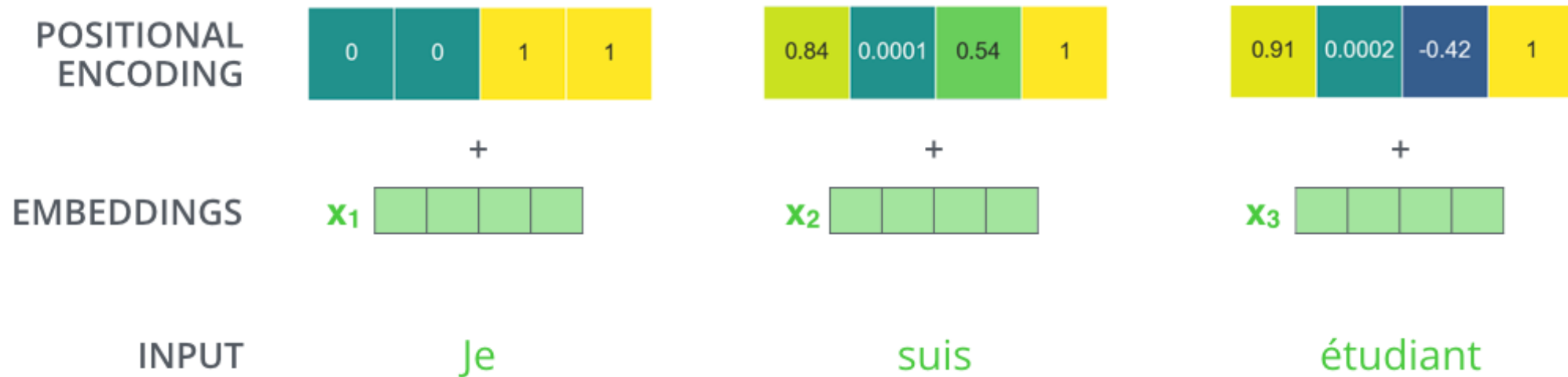
Positional Encoding

- Positional encoding is formulated as:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$

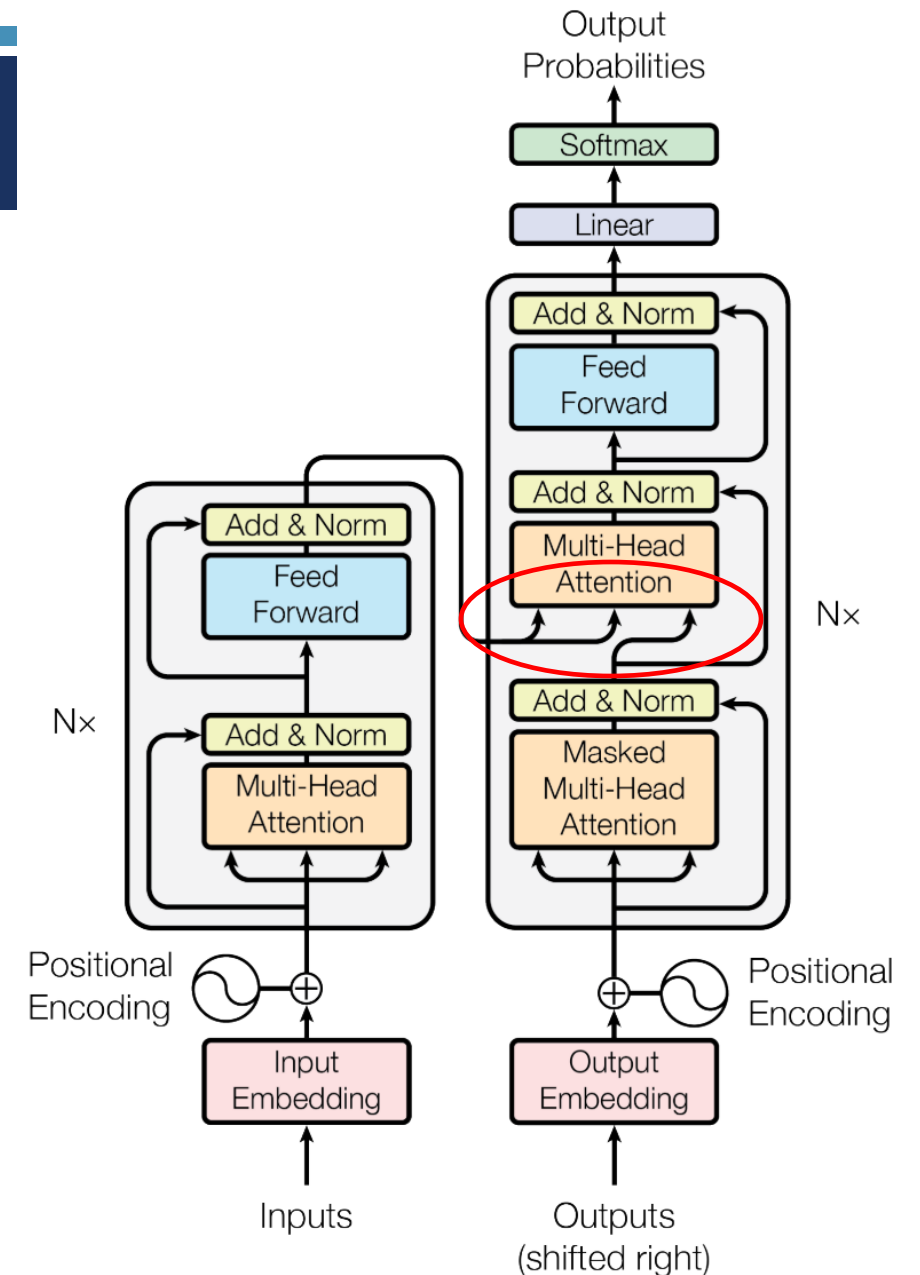
$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

where pos is the position, i is the dimension index, d_{model} is word embedding dimension.

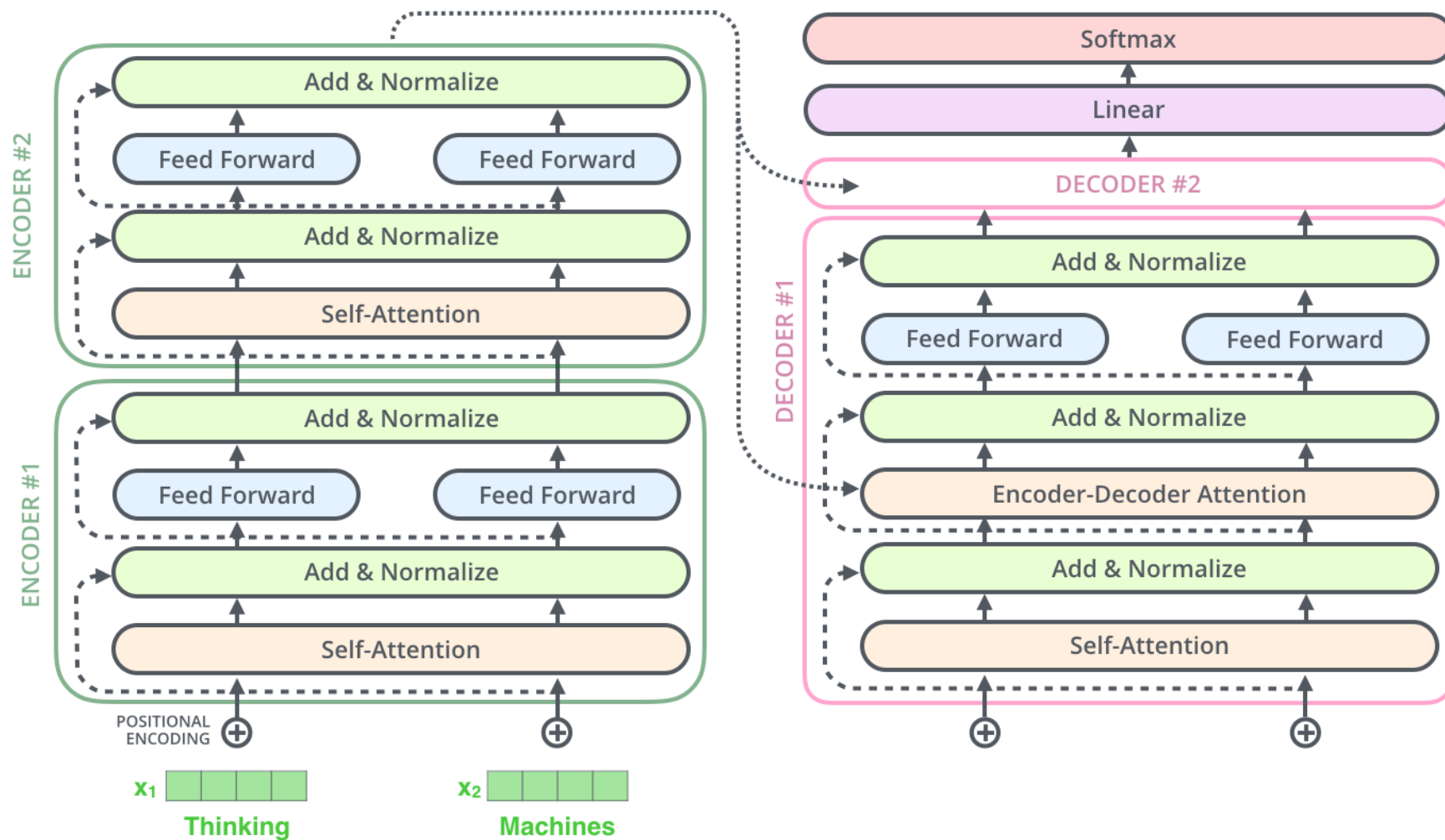


Encoder-Decoder Architecture

- Residual connections are used in both encoder and decoder.
- In the decoder, the self-attention layer is only allowed to attend to **earlier positions** in the output sequence, which is called **masked multi-head attention**.
- In the encoder-decoder attention, only K and V from the encoder are used.



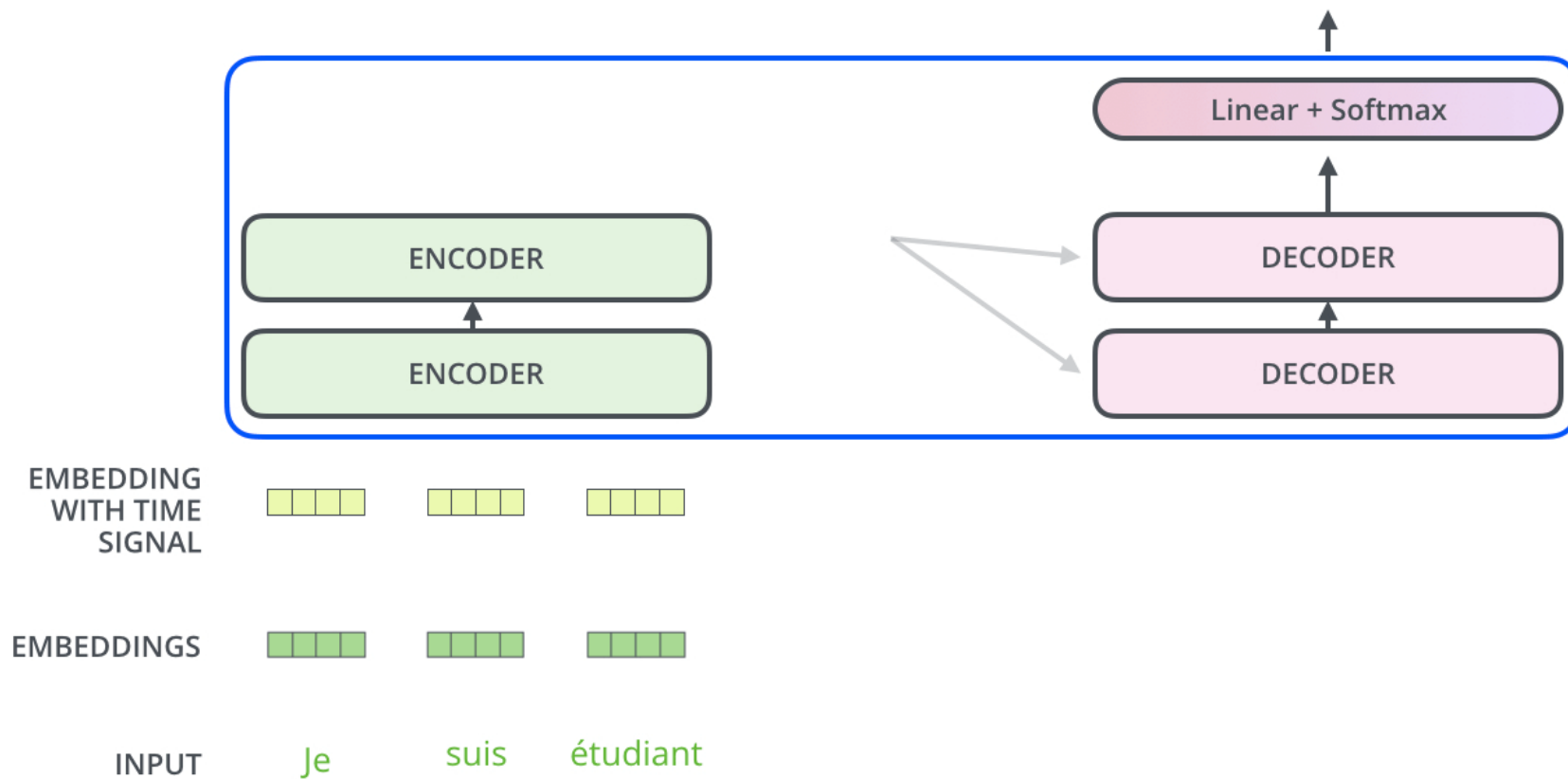
Encoder-Decoder Architecture



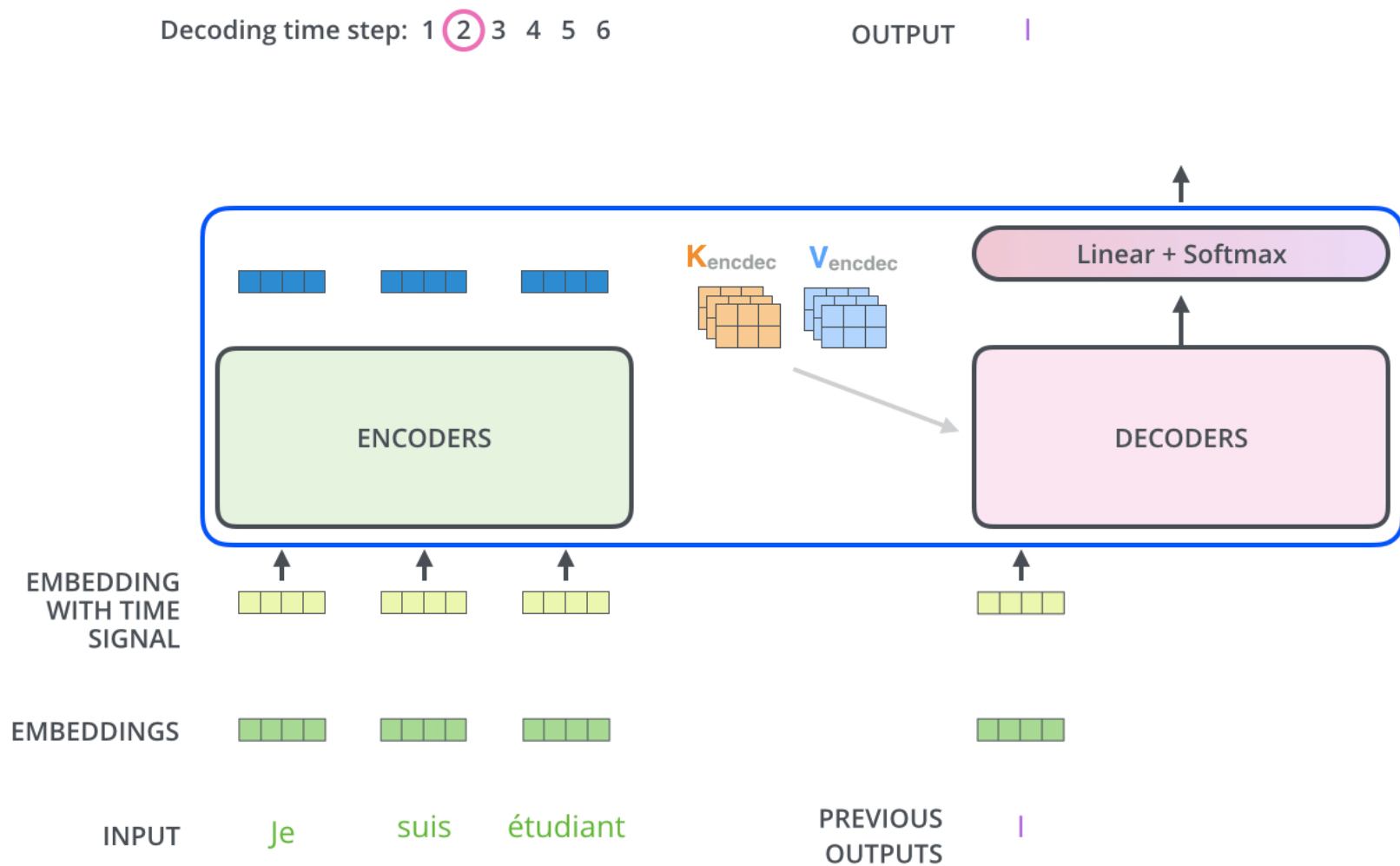
Encoder-Decoder Architecture

Decoding time step: ① 2 3 4 5 6

OUTPUT



Encoder-Decoder Architecture



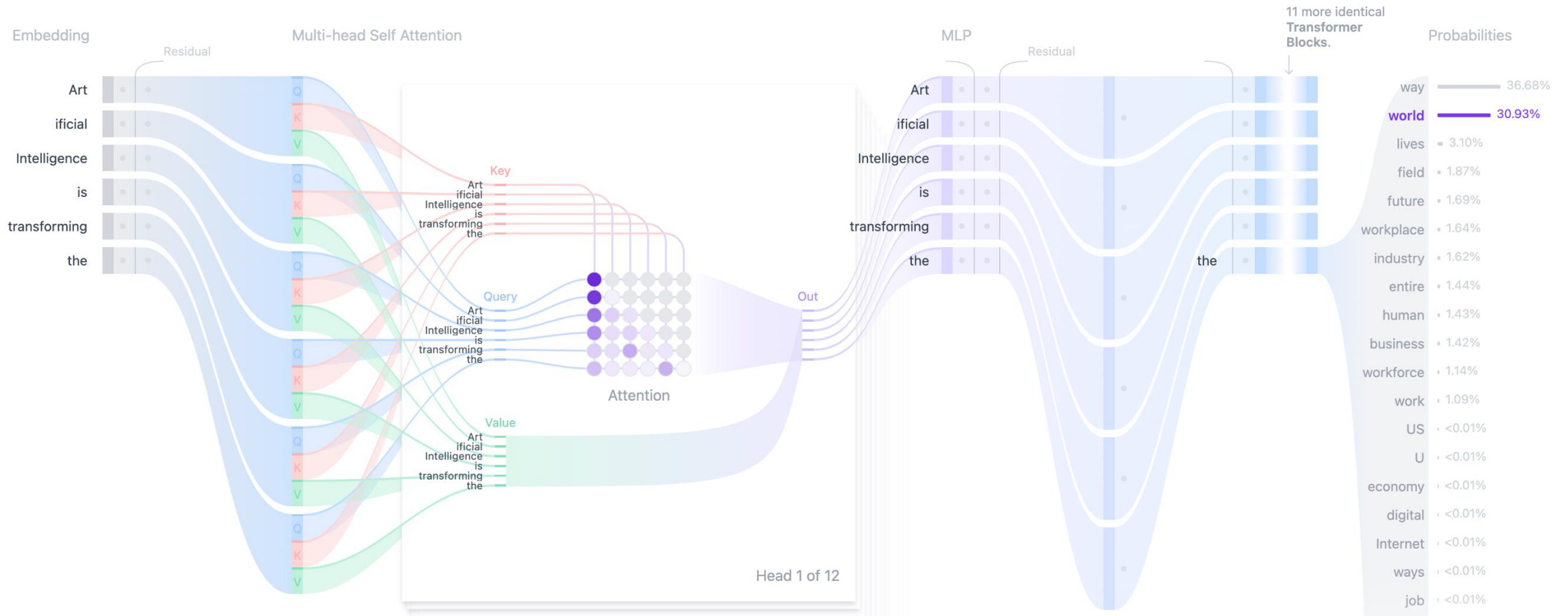
Transformer Explainer

TRANSFORMER EXPLAINER

Examples ▾ Artificial Intelligence is transforming the world

Generate

Temperature ? 1



廈門大學信息學院 (特色化示范性软件学院)

School of Informatics Xiamen University (National Characteristic Demonstration Software School)



廈門大學 计算机科学与技术系

Department of Computer Science and Technology, Xiamen University

Source: <https://poloclub.github.io/transformer-explainer/>

Vision Transformer

An **image** is worth 16x16 words: Transformers for **image** recognition at scale

[A Dosovitskiy, L Beyer, A Kolesnikov...](#) - arXiv preprint arXiv ..., 2020 - arxiv.org

... directly to **images**, with the fewest possible modifications. To do so, we split an **image** into patches ... only to small-resolution **images**, while we handle medium-resolution **images** as well. ...

☆ Save 77 Cite Cited by 23476 Related articles All 12 versions 88



The Vision Transformer treats an input image as a sequence of patches, akin to a series of word embeddings generated by an NLP Transformer.



廈門大學信息學院 (特色化示范性软件学院)

School of Informatics Xiamen University (National Characteristic Demonstration Software School)



廈門大學 计算机科学与技术系

Department of Computer Science and Technology, Xiamen University

Image source: <https://ai.googleblog.com/2020/12/transformers-for-image-recognition-at.html>

Swin Transformer

Swin transformer: Hierarchical vision transformer using shifted windows

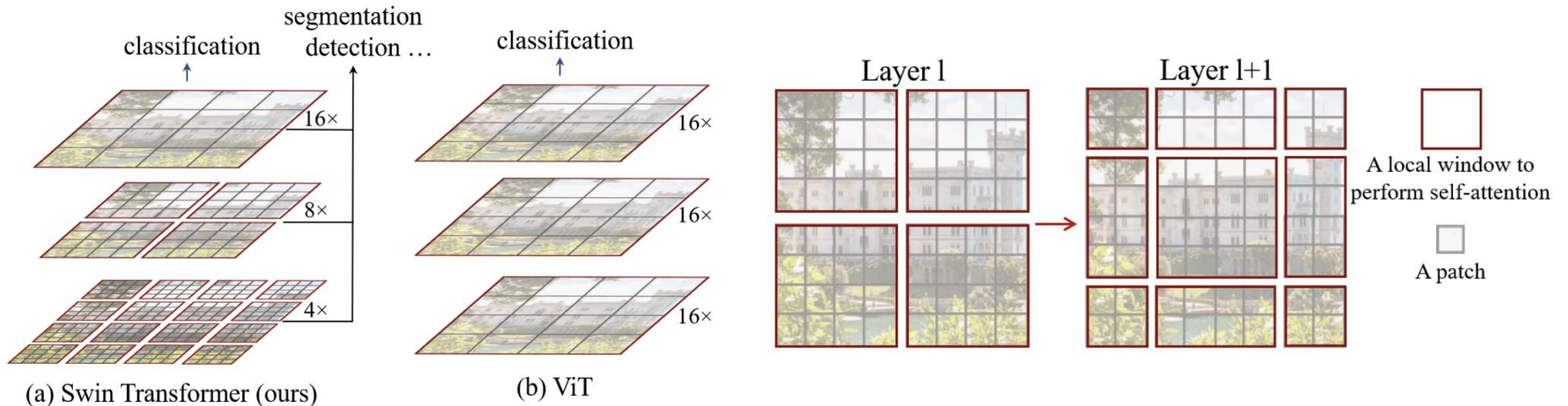
Z Liu, Y Lin, Y Cao, H Hu, Y Wei... - Proceedings of the ..., 2021 - openaccess.thecvf.com

... **Transformer**, called **Swin Transformer**, that capably serves as a general-purpose backbone for computer vision. Challenges in adapting **Transformer** ... a hierarchical **Transformer** whose ...

☆ Save 📄 Cite Cited by 11287 Related articles All 10 versions ⇨

■ Challenges in adapting Transformer from language to vision:

large variations in the scale of visual entities and the high resolution of pixels in images compared to words in text.



BERT

SOTA on 11 NLP tasks!



BERT

Bert: Pre-training of deep bidirectional transformers for language understanding

J Devlin, [MW Chang](#), [K Lee](#), [K Toutanova](#) - arXiv preprint arXiv ..., 2018 - arxiv.org

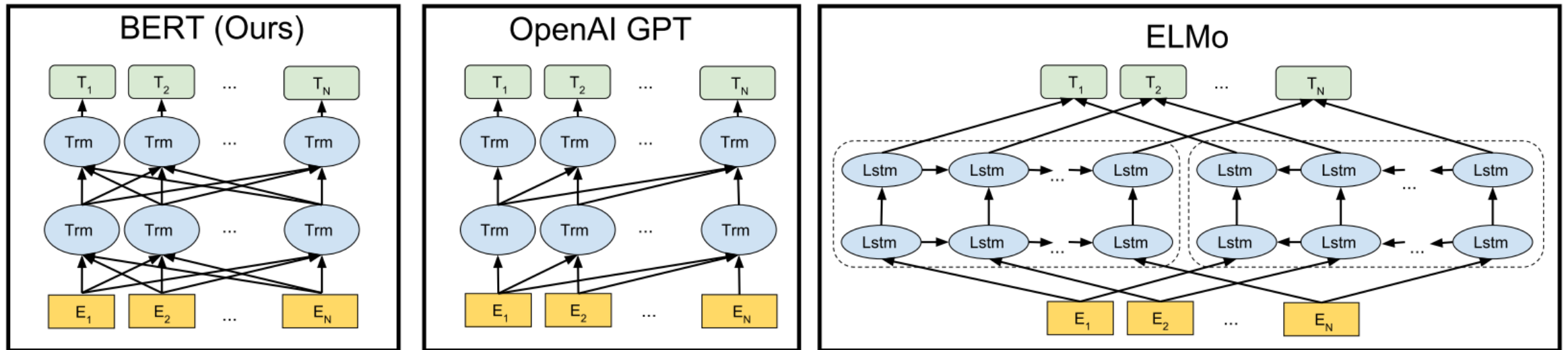
... We introduce **BERT** and its detailed implementation in this ... For finetuning, the **BERT** model is first initialized with the pre-... A distinctive feature of **BERT** is its unified architecture across ...

☆ Save 剪 Cite Cited by 81315 Related articles All 46 versions 》》

- BERT is a **pre-training framework** using deep bidirectional transformers for language understanding.
- It uses the idea of self-supervised learning, rather than training on any specific NLP task.
- After we obtain the BERT pre-trained model, we can fine-tune it for a specific NLP task.

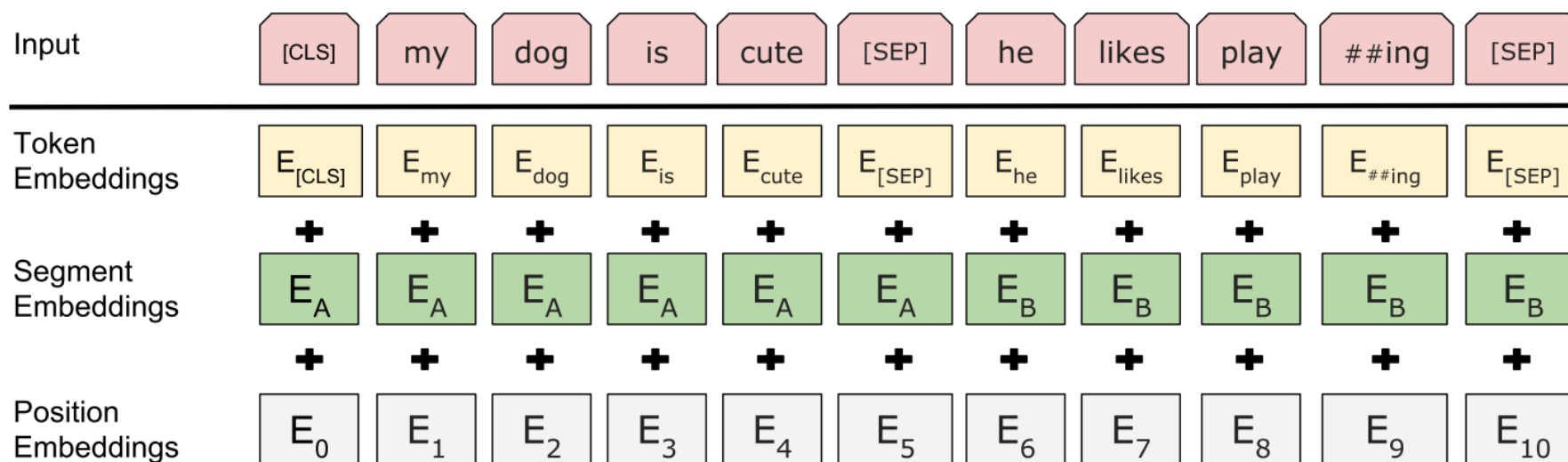
BERT Model Architecture

- BERT's model architecture is a multi-layer bidirectional Transformer encoder.
- The input is word embedding and output is context sensitive word representation.
 - Just like ELMo. But ELMo is a task-specific model, rather than a pre-trained model.



Input representation

- Positional embeddings are learnable, rather than fixed magic number as in the Transformer paper.
- Each input sequence is a pair of sentences, separated by the token `[SEP]`. It adopted two learnable embeddings to each sentence.
- `[CLS]` is the a special classification embedding for the first token of every sequence.



Pre-Training Task 1: Masked LM

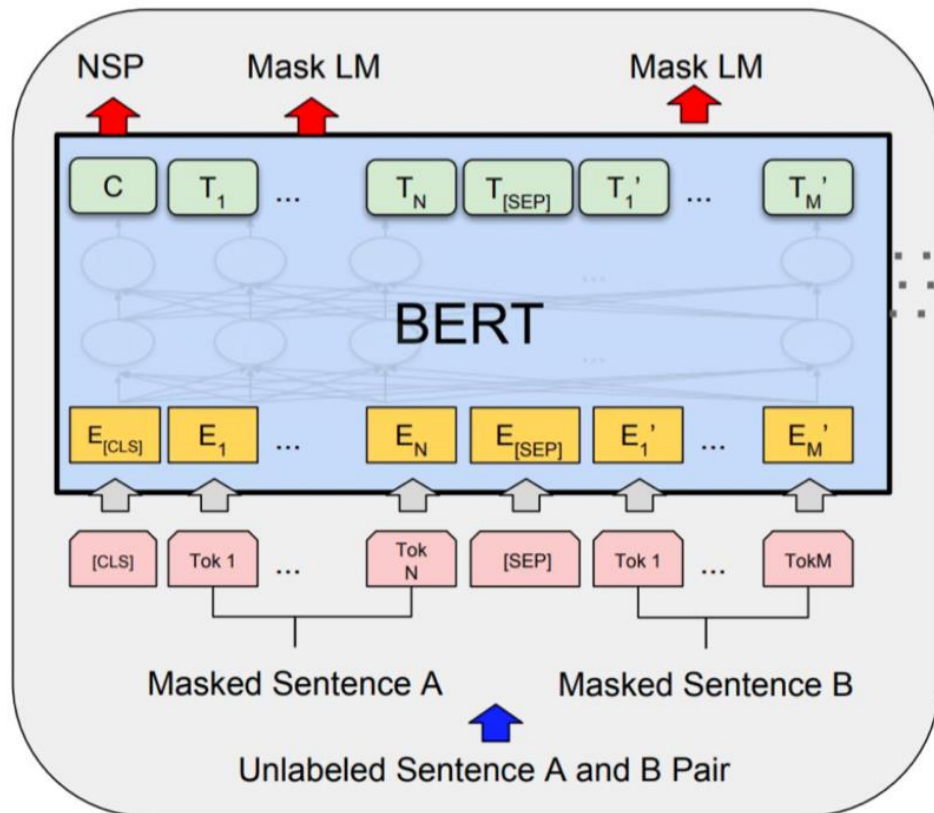
- Mask some percentage of the input tokens at random, and then predicting only those masked tokens.
- Use [MASK] token to replace 15% tokens randomly, and use the real token as the label to make it predict.
- However, the [MASK] token is never seen during fine-tuning. The authors proposed the following strategy:
 - 80% of the time: Replace the word with the [MASK] token.
 - e.g., my dog is hairy → my dog is [MASK].
 - 10% of the time: Replace the word with a random word.
 - e.g., my dog is hairy → my dog is apple.
 - 10% of the time: Keep the word unchanged. The purpose of this is to bias the representation towards the actual observed word.
 - e.g., my dog is hairy → my dog is hairy.

Pre-Training Task 2: Next Sentence Prediction

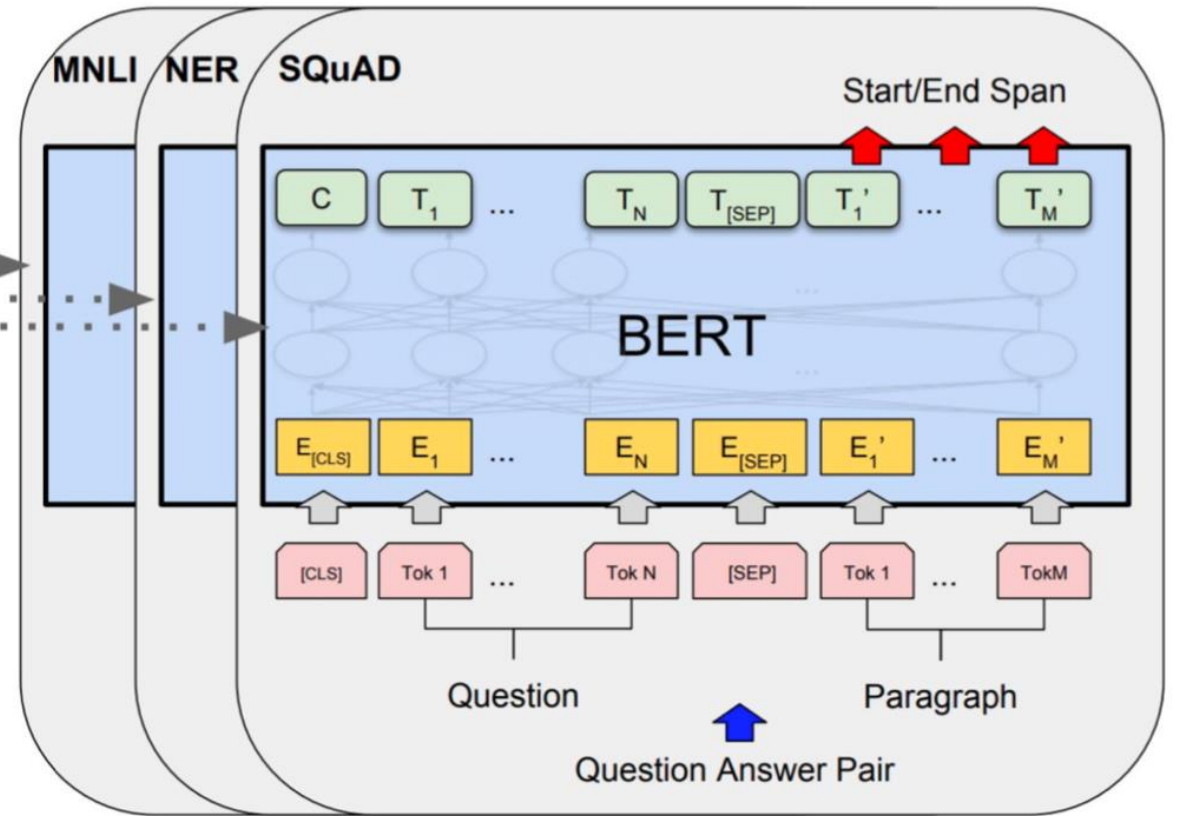
- Make the model understand the relationship between two text sentences.
- Choose the sentences A and B for each pre-training example.
 - 50% of the time B is the actual next sentence that follows A.
 - 50% of the time it is a random sentence from the corpus.
- Example:
 - **Input** = [CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]
Label = IsNext
 - **Input** = [CLS] the man [MASK] to the store [SEP] penguin [MASK] are flight ##less birds [SEP]
Label = NotNext



Pre-Train and Fine-Tune

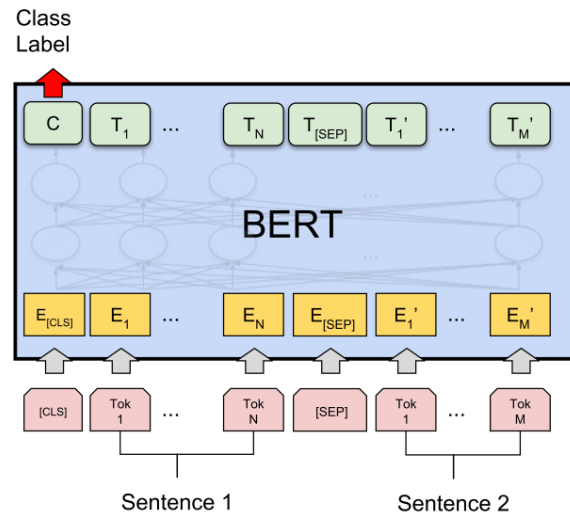


Pre-training

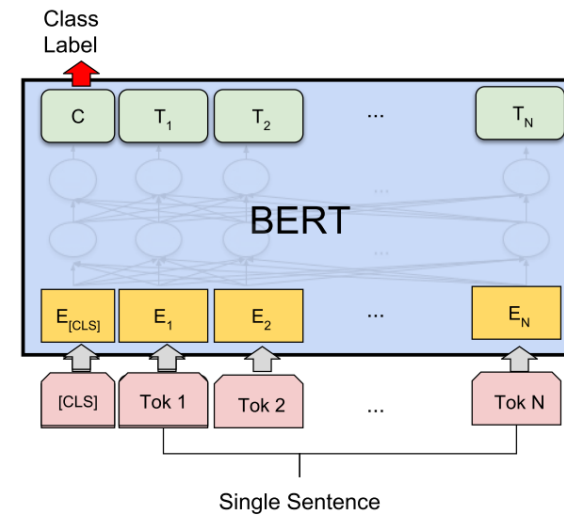


Fine-Tuning

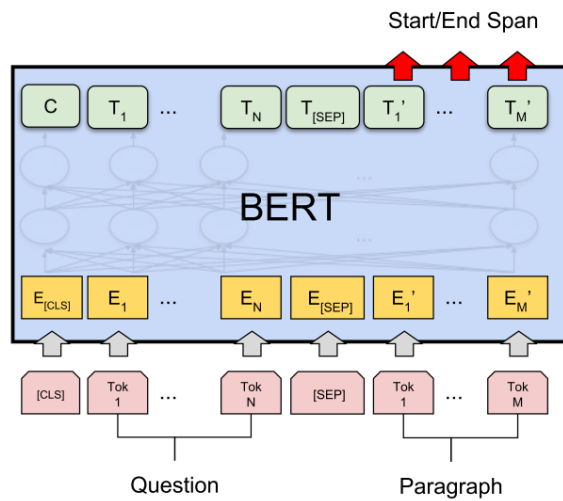




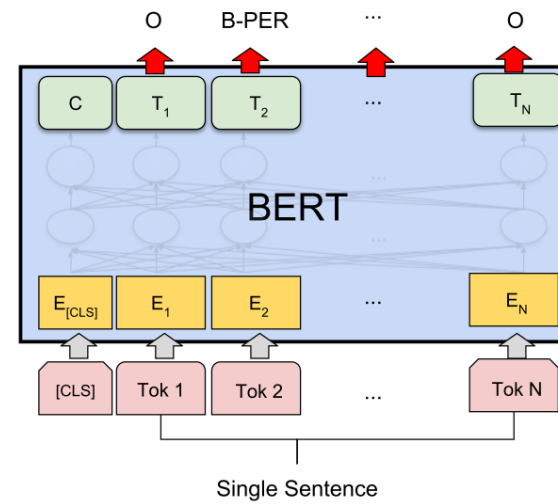
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(b) Single Sentence Classification Tasks:
SST-2, CoLA



(c) Question Answering Tasks:
SQuAD v1.1



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

GPT

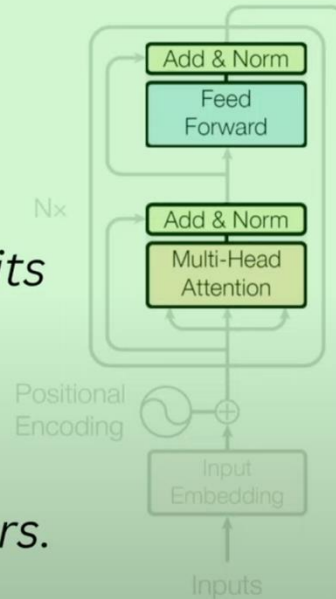


BERT v.s. GPT

BERT

Google

*use transfer learning to **continue learning** from its existing data when adding user-specific tasks and layers.*



GPT

OpenAI

*decodes from its massive pre-learned embeddings to present output that matches user prompts. It does **not learn** anything new.*

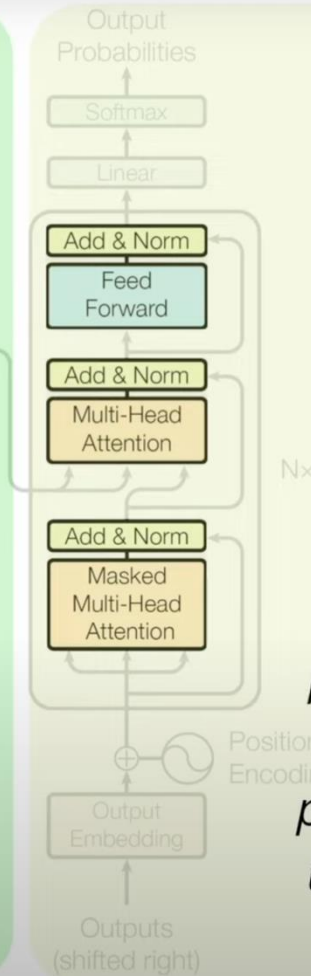
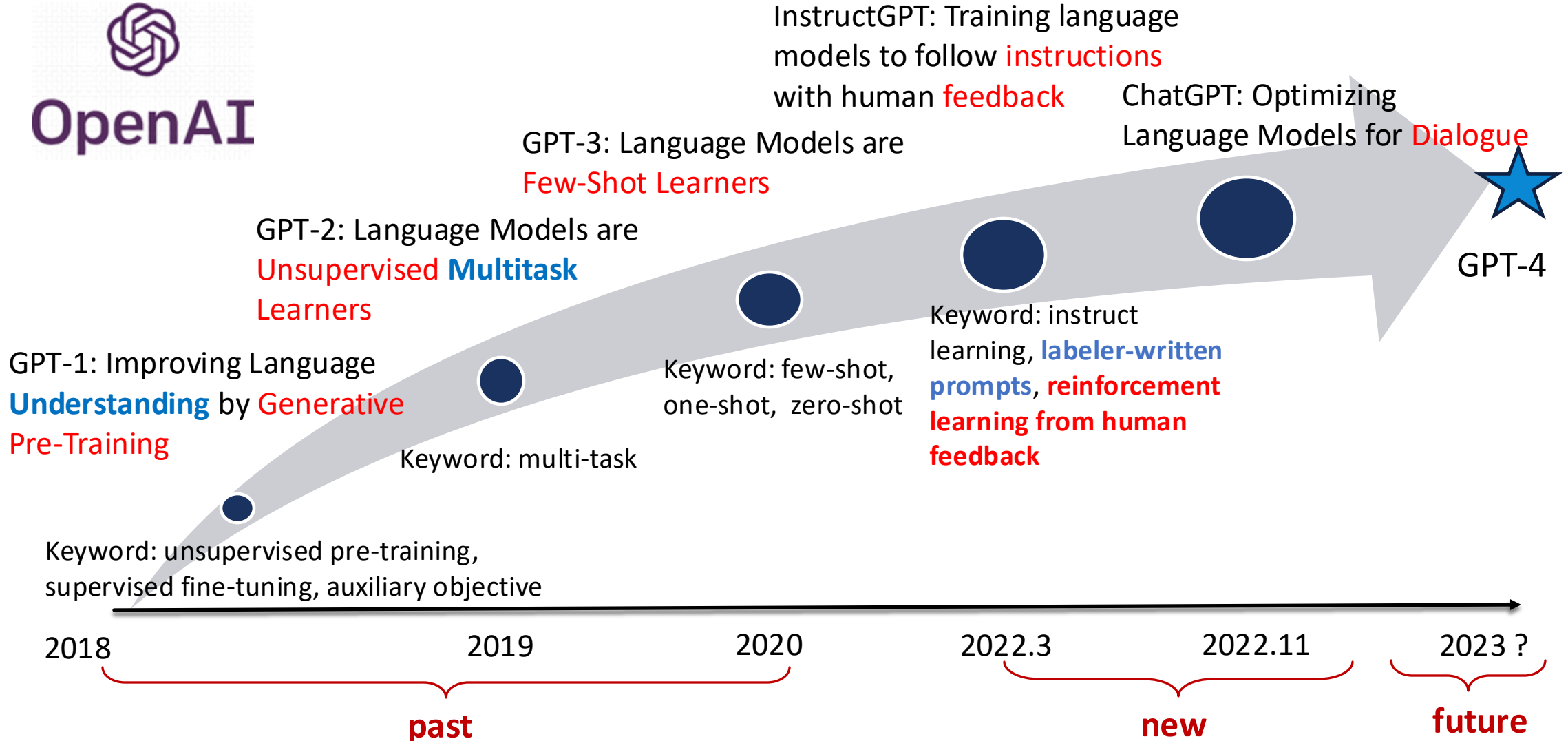


Figure 1: The Transformer - model architecture.



History of GPT



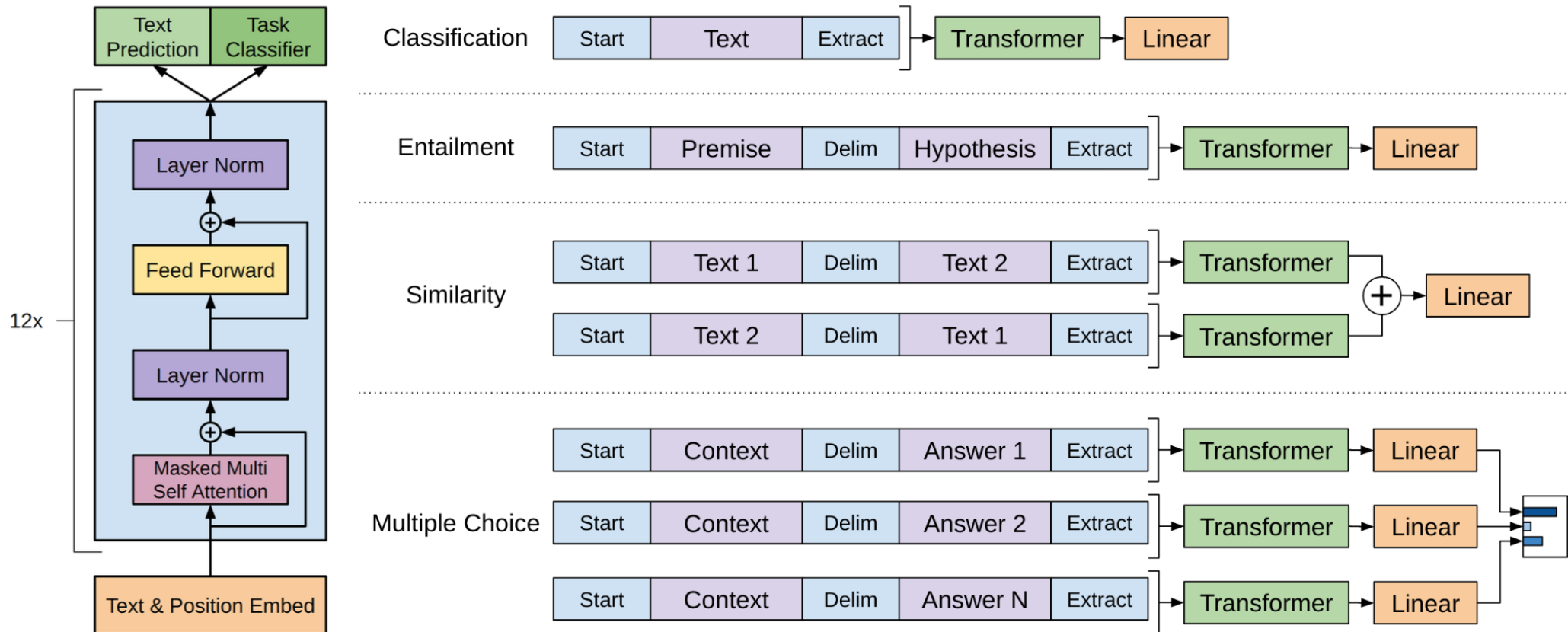
GPT-1

[PDF] Improving language understanding by generative pre-training

A Radford, K Narasimhan, T Salimans, I Sutskever - 2018 - mikecaptain.com

... on four types of **language understanding** tasks – natural **language** inference, question ... architectures specifically crafted for each task, significantly **improving** upon the state of the art in 9 ...

☆ Save 📄 Cite Cited by 6938 Related articles All 15 versions 🔗



GPT Keyword: unsupervised pre-training, supervised fine-tuning, auxiliary objective

- Previously, NLP tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with **supervised learning on task-specific datasets**.
- GPT-2 is trained on a new dataset of millions of webpages called WebText without any explicit supervision.

Ability: Zero-shot or one-shot:

- **Zero-shot**: use summarization as an example
 - Input: original text + “TL; DR”
 - Output: summary
- **One-shot**: use translation as an example
 - Input: “English sentence1 = French sentence1” + “English sentence2 = ”
 - Output: “French sentence2”

One-shot is not supervised information. It is not involved into the training process

Language models are few-shot learners

[T Brown, B Mann, N Ryder...](#) - Advances in neural ..., 2020 - proceedings.neurips.cc

... up **language models** greatly improves task-agnostic, **few-shot** ... GPT-3, an autoregressive **language model** with 175 billion ... **language model**, and test its performance in the **few-shot** ...

☆ Save 📄 Cite Cited by 16011 Related articles All 27 versions 🔗

- Pre-trained model with **In-context learning** (few-shot, one-shot, zero-shot) is becoming competitive with prior state-of-the-art fine-tuning approaches.

Zero-shot

The model predicts the answer given only a natural language description of the task. **No gradient updates are performed.**

```
1 Translate English to French: ← task description
2 cheese => ..... ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. **No gradient updates are performed.**

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← example
3 cheese => ..... ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. **No gradient updates are performed.**

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← examples
3 peppermint => menthe poivrée ←
4 plush girafe => girafe peluche ←
5 cheese => ..... ← prompt
```



InstructGPT

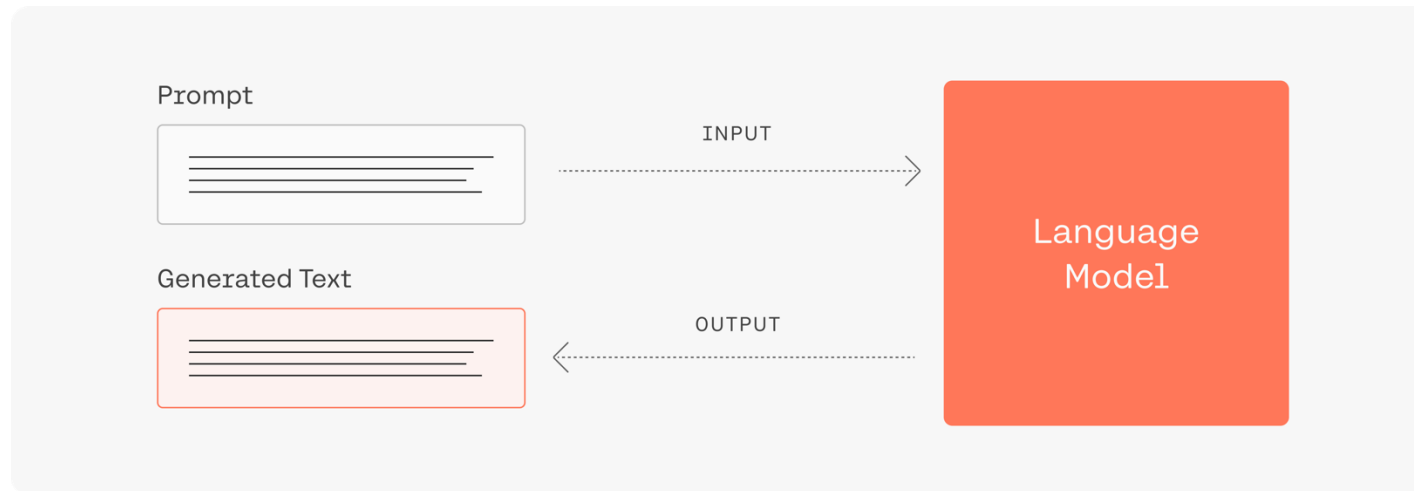
Training language models to follow instructions with human feedback

[L Ouyang, J Wu, X Jiang, D Almeida...](#) - Advances in ..., 2022 - proceedings.neurips.cc

... **InstructGPT**. In human evaluations on our prompt distribution, outputs from the 1.3B parameter **InstructGPT** model ... Moreover, **InstructGPT** models show improvements in truthfulness and ...

☆ Save 📄 Cite Cited by 2748 Related articles All 10 versions 🔗

- GPT-3 is good at in-context learning tasks, but these models are not aligned with their users.
 - Can only handle traditional NLP tasks, but not human interaction.
- Instruction Tuning
 - Unify tasks in the form of Prompts.
 - Fine-tune the language model.
 - The model can handle unseen tasks.

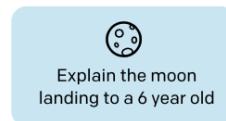


InstructGPT

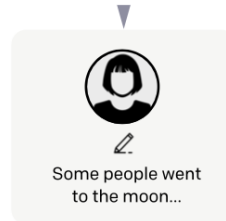
Step 1

Collect demonstration data, and train a supervised policy.

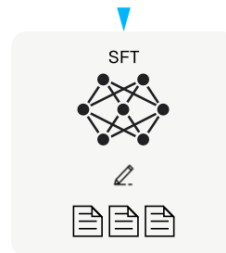
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



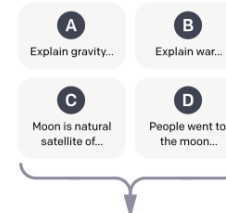
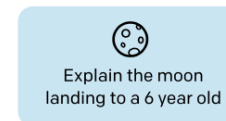
This data is used to fine-tune GPT-3 with supervised learning.



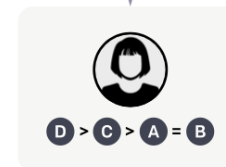
Step 2

Collect comparison data, and train a reward model.

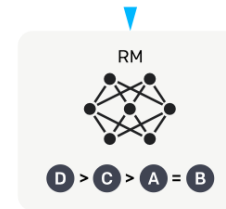
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



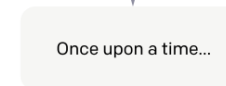
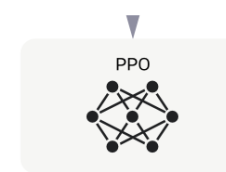
Step 3

Optimize a policy against the reward model using reinforcement learning.

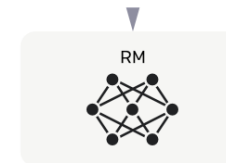
A new prompt is sampled from the dataset.



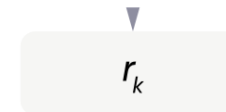
The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



ChatGPT v.s. InstructGPT

- ChatGPT can generate more detailed responses
 - This might stem from the annotators' preference for "more detailed responses" during the training reward model process => a preference for verbosity.
- ChatGPT excels more in multi-turn dialogue formats
 - This might be due to the multi-turn dialogue data annotated by the annotators during the instruction fine-tuning process.
- ChatGPT is better at capturing COT and long-term dependencies in multi-turn dialogues
 - This could be attributed to ChatGPT's initialization model.



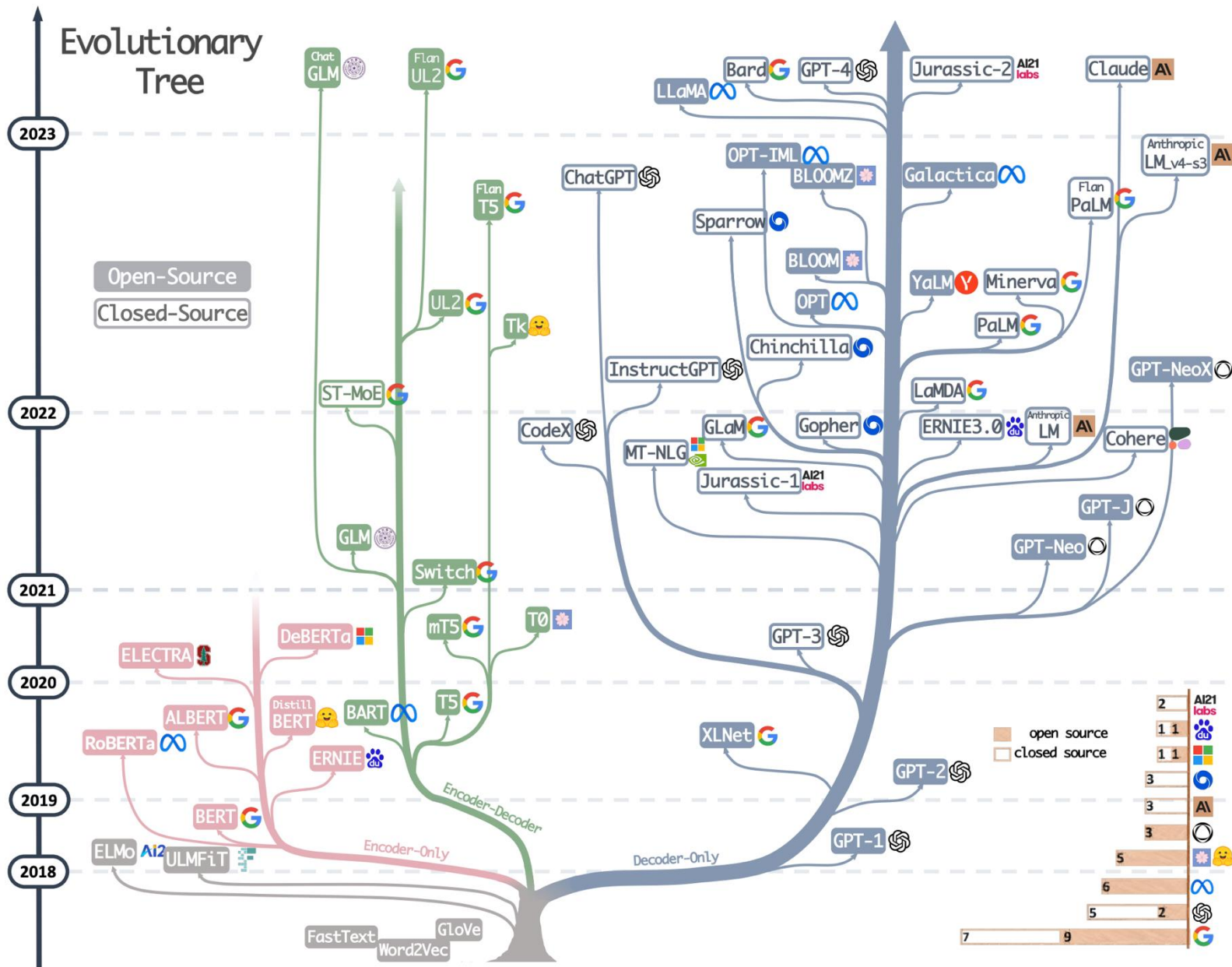


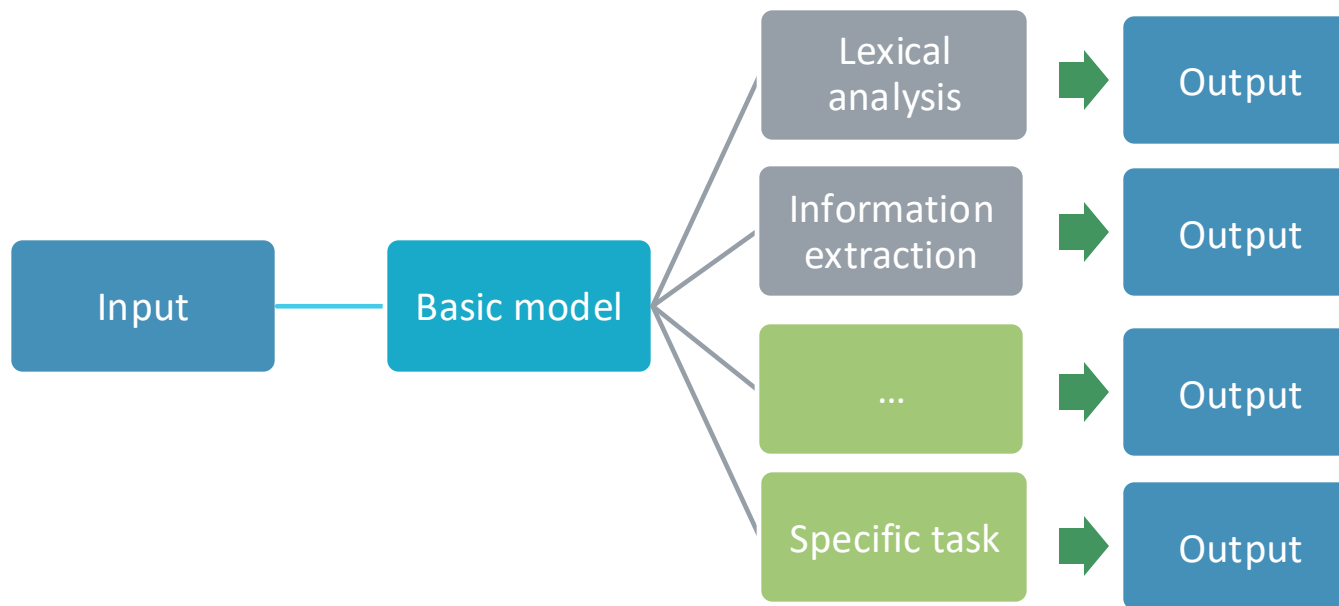
Image source: Yang, Jingfeng, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. "Harnessing the power of llms in practice: A survey on chatgpt and beyond." arXiv preprint arXiv:2304.13712 (2023).

Times of NLP Have Changed...

Before 2015



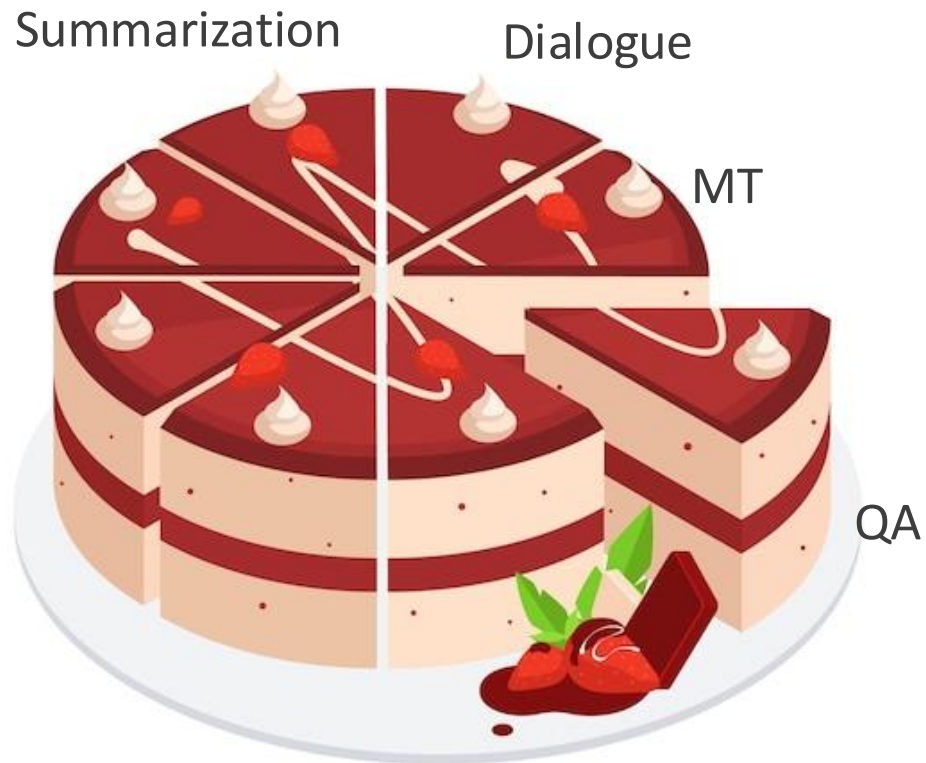
2015-2022



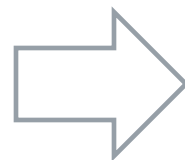
2022-



Times of NLP Have Changed...



Divided by task



Divided by process

Conclusion

After this lecture, you should know:

- Why do we need word embedding?
- How to generate $xxx \rightarrow vec$?
- Why context information is important can how to incorporate it into word embedding?
- What is multi-head self-attention?
- What is a pre-trained language model and how to use it?

Suggested Reading

- Word2vec paper: Distributed representations of words and phrases and their compositionality
- ELMo paper: Deep contextualized word representations
- Transformer paper: Attention is all you need
- 李沐: Transformer论文逐段精读
- BERT paper: Bert: Pre-training of deep bidirectional transformers for language understanding
- Excellent Transformer tutorial with notebook
- Illustrated Transformer



Assignment 3

- Assignment 3 will be released soon. The deadline is **18:00, 11th November.**

Thank you!

- Any question?
- Don't hesitate to send email to me for asking questions and discussion. 😊