AdvNLE Seminar 8

Pre-training Large Language Models

Dr Julie Weeds, Spring 2024

Warm-up

- what **applications** can you think of where it might be useful to have a measure of how similar two sentences or documents are?
  - what are the different ways in which two sentences might be similar?
  - how does this affect the applications?

Paraphrase identification and semantic matching

- text simplification
- automated marking
- question answering
- text summarization
- information retrieval
- recommendation systems
- document clustering

See: Yang et al. (2020) for a discussion of 4 different categories of semantic matching problems in NLP (https://arxiv.org/pdf/2004.12297.pdf)

Semantic matching in … Question Answering

- "Who became the head of the UK government in 1951?"
- "Who was elected as British prime minister in 1951?"

- If we know the answer to one of these questions …
- we probably know the answer to the other one too

Sir Winston Leonard Spencer Churchill, KG, OM, CH, TD, DL, FRS, RA was a British politician, statesman, army officer, and writer. He was Prime Minister of the United Kingdom from 1940 to 1945, during the Second World War, and again from 1951 to 1955.

Semantic matching in … Automated Marking

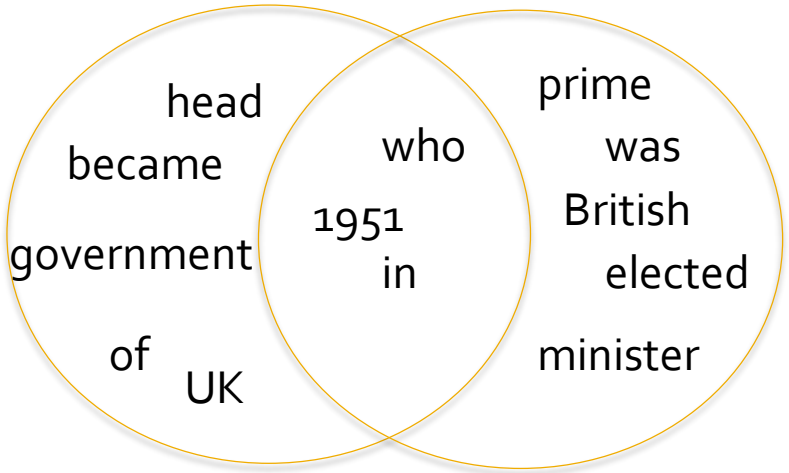
How are igneous rocks formed?

Igneous rocks form when magma (molten rock) cools and crystallizes, either at volcanoes on the surface of the Earth or while the melted rock is still inside the crust.

Igneous rocks (from the Latin word for fire) form when hot, molten rock crystallizes and solidifies.

Simple text matching

- word overlap
  - e.g., using Jaccard's coefficient



$$jacc_{sim(A,B)} = \frac{|A \cap B|}{|A \cup B|} = \frac{3}{13}$$

- Bag-of-words representations of sentences / documents
  - weight words using TF-IDF
  - cosine similarity between vectors
- TF-IDF = "term frequency, inverse document frequency"
- gives more weight to:
  - higher frequency terms
  - more discriminating terms

Beyond simple text matching

- Why do we need more powerful methods than simple text matching?

Previously

- Distributional models of word meaning
  - how similar are two words based on how they are used in text?
- Language models
  - how likely is a sequence of words in a language?
- Neural language models
- Tasks
  - Sequence labelling
  - Sequence classification
  - Sequence generation

This week

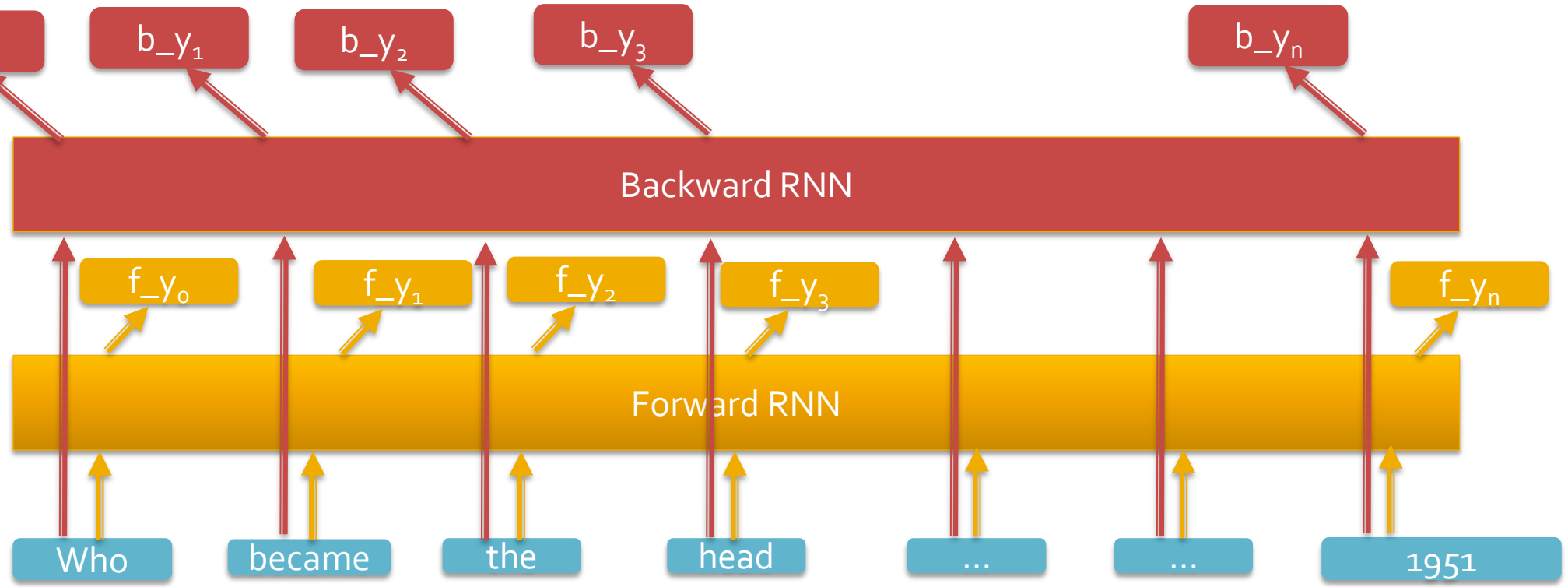
- Large language models
  - Contextualised word embeddings (ELMo)
  - Transformer architecture and attention
  - BERT (bidirectional encoder representation from transformers)
  - Pre-training:
    - Masked language modelling
    - Autoregressive language modelling
    - Next sentence prediction
  - **Sentence representations / paraphrasing (Seminar)**

Lecture questions

1. Imagine you have a 100M word corpus of news articles with a vocabulary of size 50K. Explain the difference between static word embeddings and contextualized word embeddings derived from this corpus.
2. Why are transformers now generally preferred to LSTMs in the NLP community?
3. In the sentence, "A few faint stars glimmered in the sky.", what words might need to pay attention to other words in the sentence, in order for a good contextualized word representation to be derived?
4. Explain how the output of an attention head is derived (for one of the words in the sentence above)
5. Will the encoder of a transformer produce the same representation of the sentences, "The dog bit the boy." and the "The boy bit the dog." Why/ why not?

Contextualised word embeddings

- Represent each word based on its context
- e.g., concatenation of [b_y_t,f_y_t]
- compose contextualised word embeddings as before



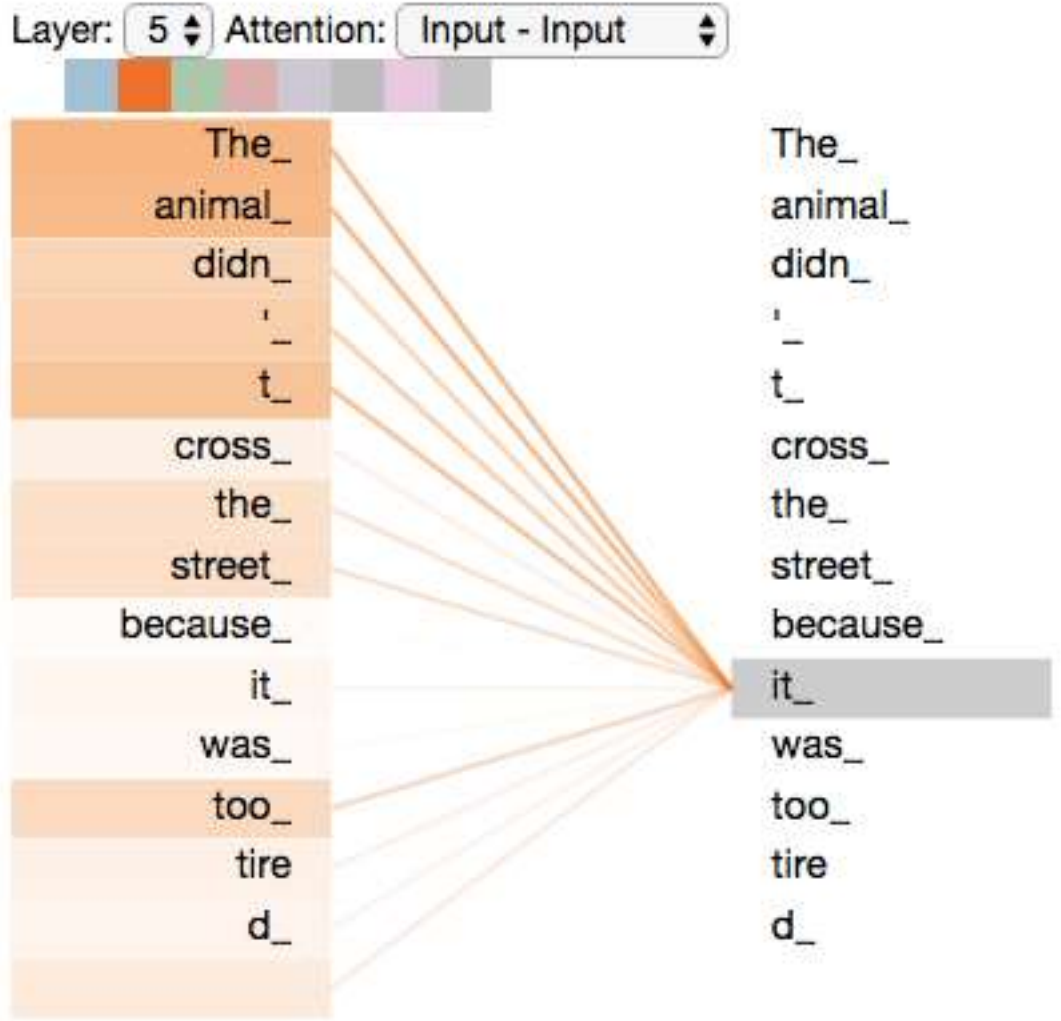
embedding layer

Transformers (Vashwani et al. 2017)

- Sequence transduction model using stacked self-attention
  - no convolutions or recurrence
  - easier to parallelize than RNNs
  - faster to train than RNNs
  - captures more long-range dependencies than CNNs with fewer parameters
- What's a sequence transduction model?
- What's stacked self-attention?

Self-Attention (High level)

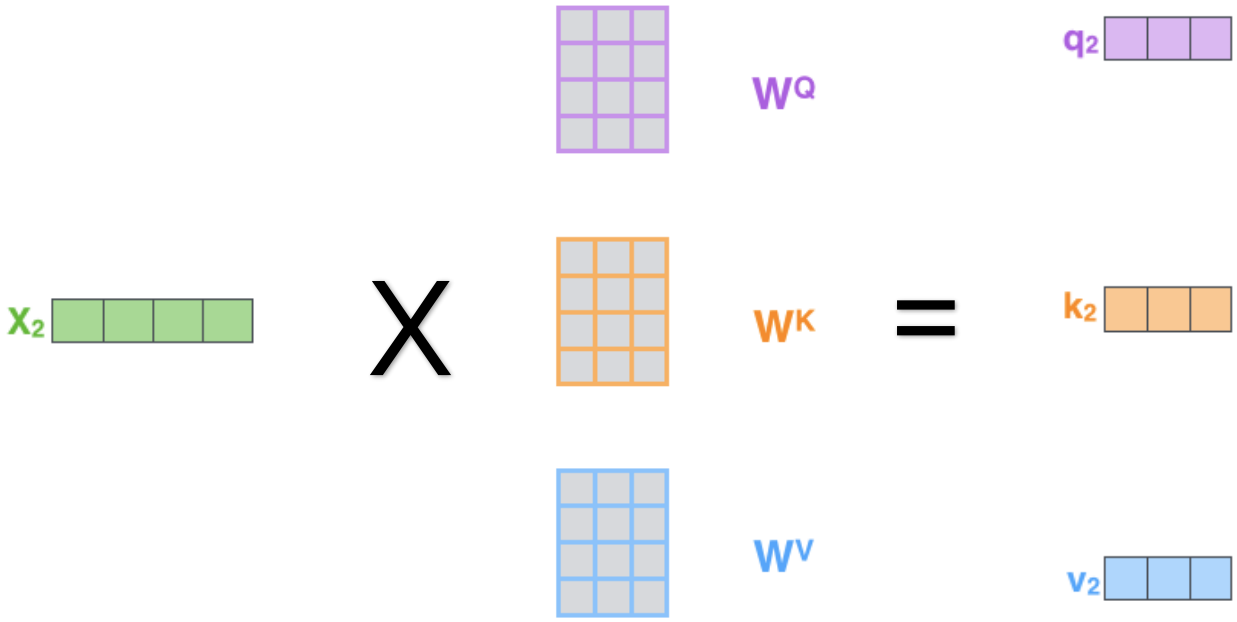
- What to pay attention to when encoding or decoding
  - **Attention**: in previous layer
  - **Self-attention**: in the same layer
- When encoding the word ***it*** in the sentence "*The animal didn't cross the street because* ***it*** *was too tired*", self-attention should allow the model to associate ***it*** with ***animal***



Example from http://jalammar.github.io/illustrated-transformer/

Self-attention - Step 1

- Step 1: from the encoder's input vector (e.g., word embedding), create 3 vectors: **Q**uery, **K**ey and **V**alue
- Q,K, and V embeddings usually smaller in number of dimensions e.g.,
  - input 512
  - output 64



input embedding

trainable weight matrices

output query, key and value embeddings

Self-attention score

- Step 2: To work out how much the word in position$_i$ (e.g., "*it*") should pay attention to a word in position$_j$ (e.g., "*animal*") words in the sentence, we take the **dot product** of the Q_i (the **query** vector for *it*) with the K_j (the **key** vector for *animal*)

$$SelfAtt(w_i, w_j) = Q_i . K_j$$

- Do this for every word in the sentence with every other word in the sentence

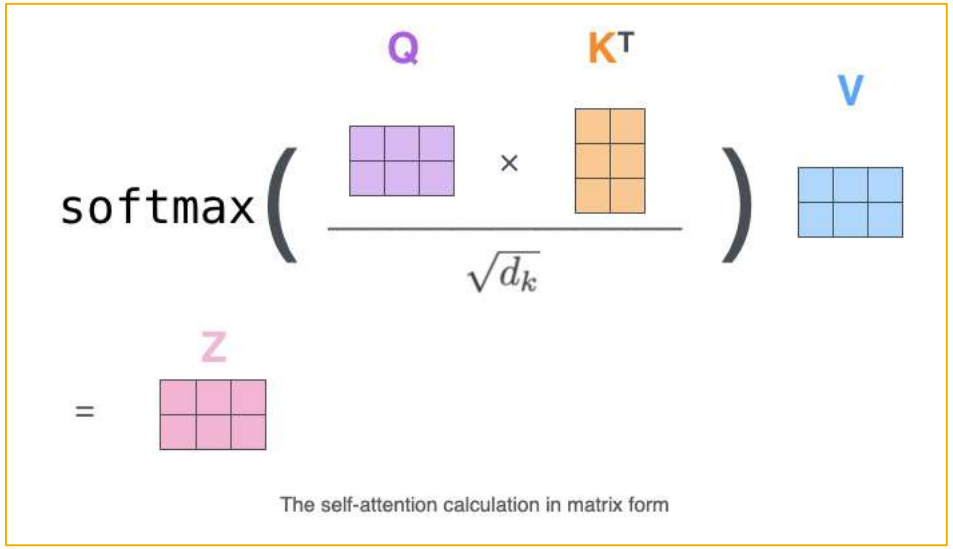
Normalise

- Step 3: Divide all of the scores by 8
  - the square root of the dimension of the key vectors which is 64 here
  - this is the default and seems to lead to more stable gradients
- Step 4: Apply a softmax to the scores
  - this results in them all being positive
  - and summing to 1
  - so can be thought of a as a probability / proportion – "what proportion of my attention should I pay to word j?"

Weight sum of the value vectors

- Step 5: For each word w_i, multiply the value vector (V_j) of each other word w_j by its softmax score with w_i
- Step 6: Sum to produce output embedding (at this layer) for w_i

$$Z_i = \sum_j softmax\_score(w_i, w_j) \times V_j$$



Self-attention summary

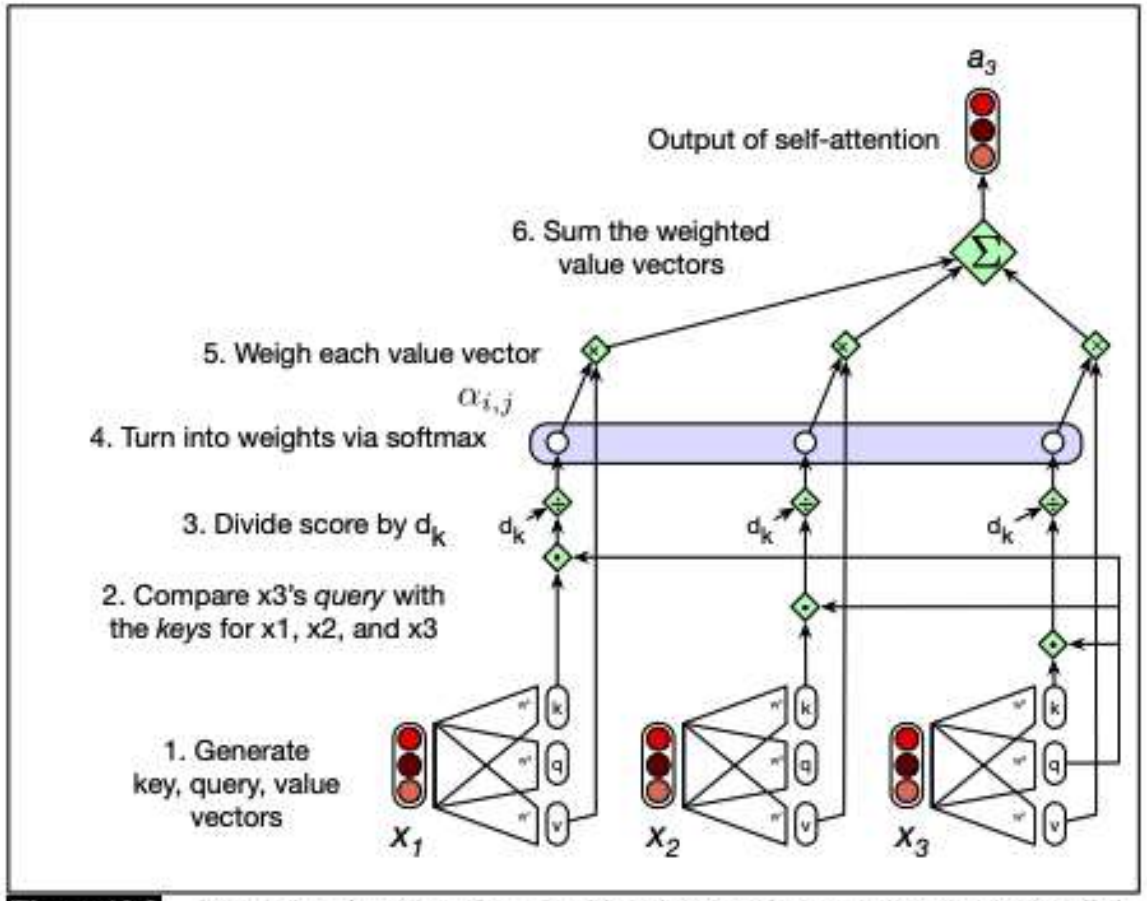
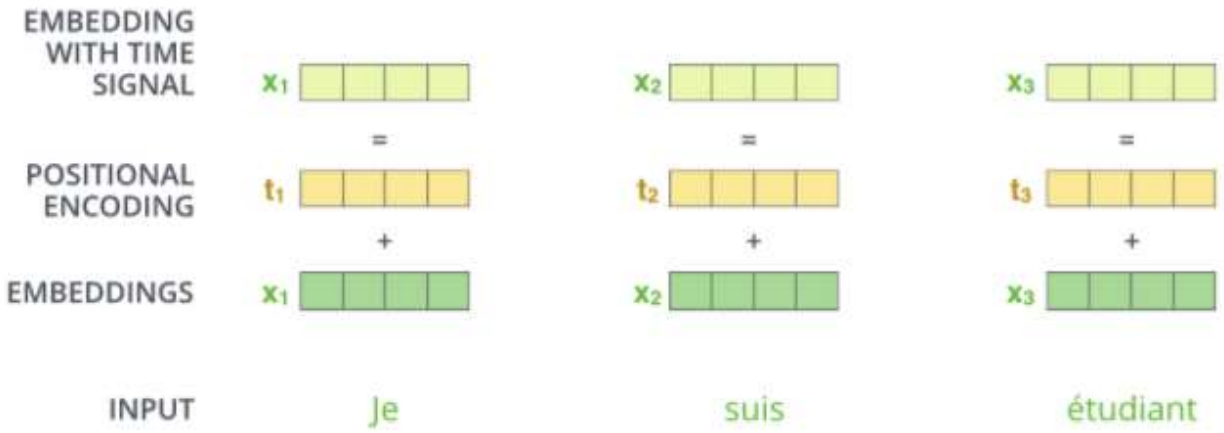


Figure 10.3 Calculating the value of $\mathbf{a}_3$, the third element of a sequence using causal (left-to-right) self-attention.

- Image taken from Chapter 10, Jurafsky and Martin
- This is actually for uni-directional self-attention
- However, it would be the same for the a_3 in the bidirectional case assuming an input sequence which is 3 tokens long!

Positional Encoding

- How does the model account for word order?
- Adds a positional encoding vector to each input embedding
- Positional encoding usually follows a fixed pattern
  - enables the model to determine the position of each word and distance between them

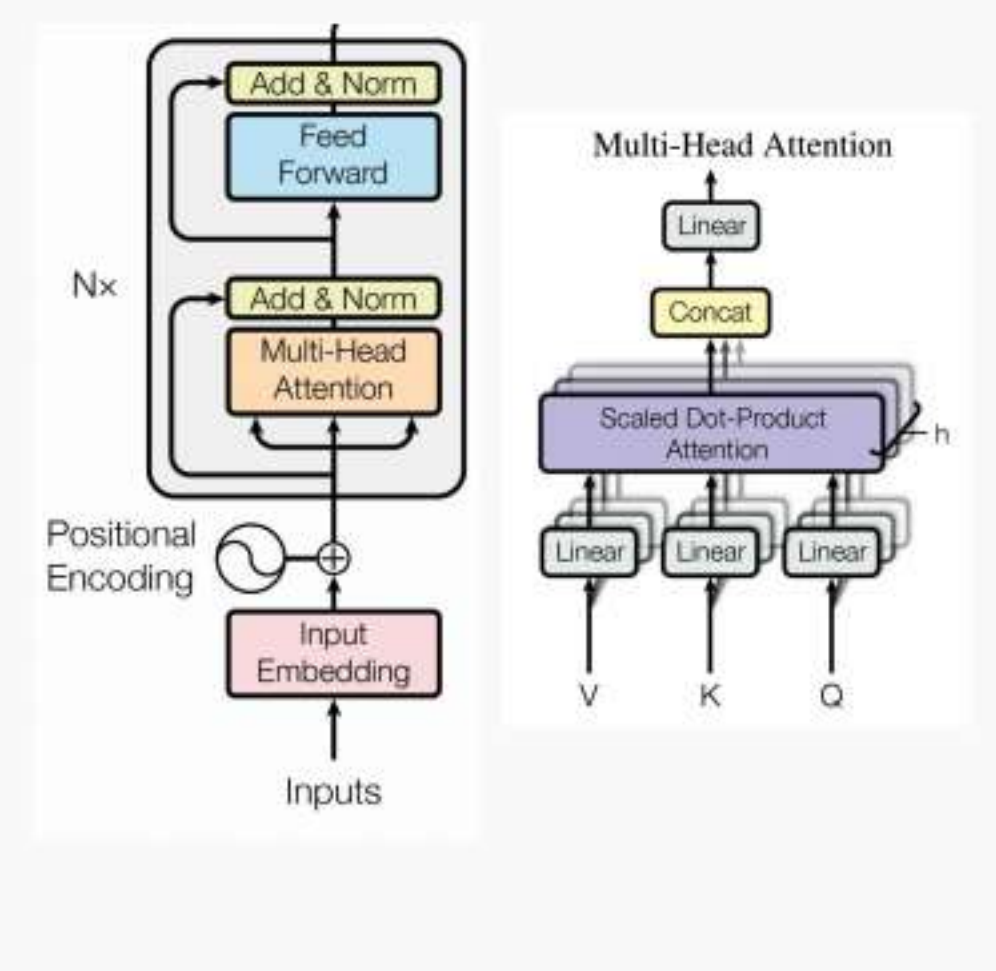


BERT (Devlin et al. 2019)

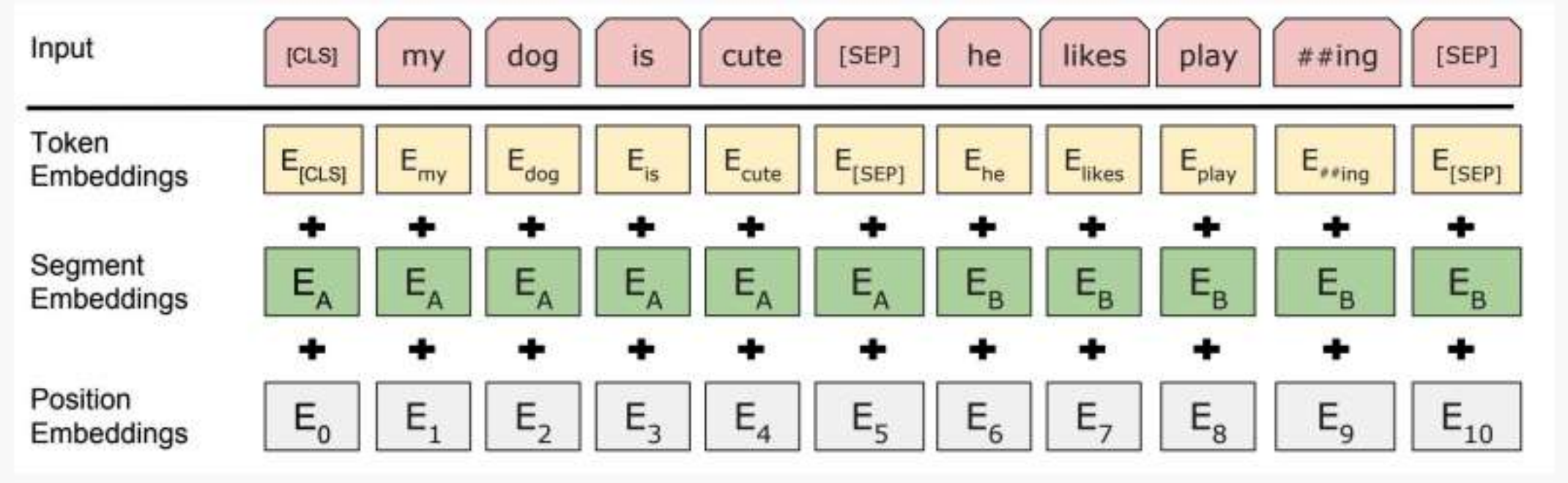
- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
  - just uses the encoder portion of the transformer architecture
  - uses left and right context of a target word to build representation
- Pre-trained on general language modelling tasks
  - masked language modelling task
  - next sentence prediction
- fine-tuned on task-specific training data
- Pre-trained models available from Google:
  - https://github.com/google-research/bert
- And from huggingface!

Encoder Representation

- Multi-headed self-attention
  - models context
- Feed-forward layers
  - non-linear hierarchical features
- Layer norm and residuals
  - makes training deep networks healthy
- Positional embeddings
  - learn relative positioning

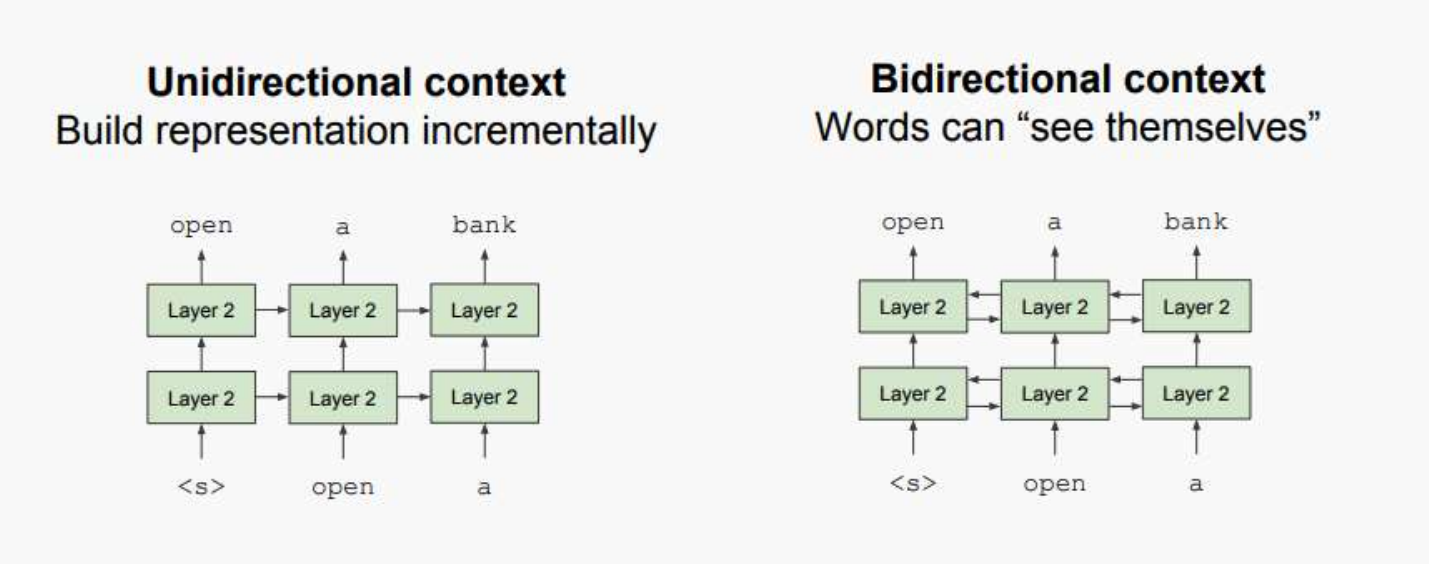


Input Representation



- Each token is sum of three embeddings
- 30,000 WordPiece vocabulary → morphology → better representations for rare words
- sentences are separated by a special "SEP" token
- a special "CLS" token is added at the beginning of the chunk
  - often used in classification

Unidirectional vs Bidirectional models

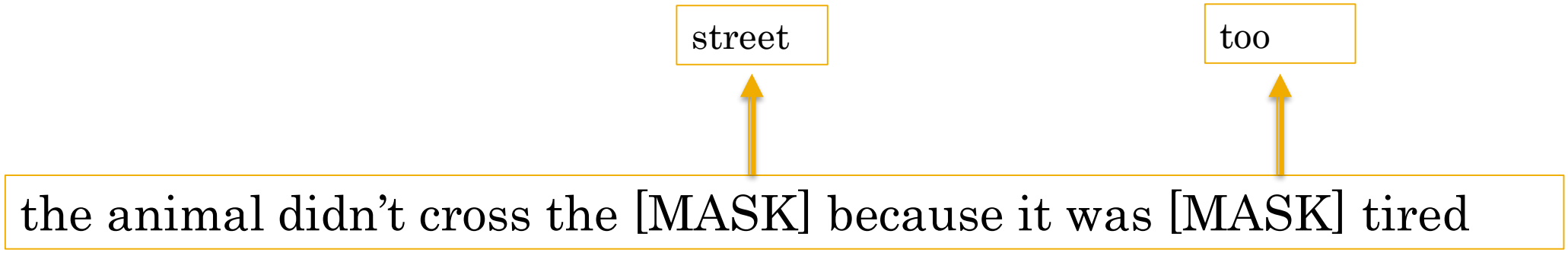


from https://nlp.stanford.edu/seminar/details/jdevlin.pdf

- Most language models trained to predict the next word in sequence
- Known as **auto-regressive** or **unidirectional**
- BERT uses the whole of the sentence to predict missing or masked word
- masked language model is **bidirectional**

Masked Language Model

- Mask out k% of the input words, and predict the masked words
  - if k is too small → expensive to train
  - if k is too large → not enough context
  - Google use k=15



Next Sentence Prediction

- To learn relationships between sentences, predict whether sentence B is actual sentence that follows sentence A or is a random sentence

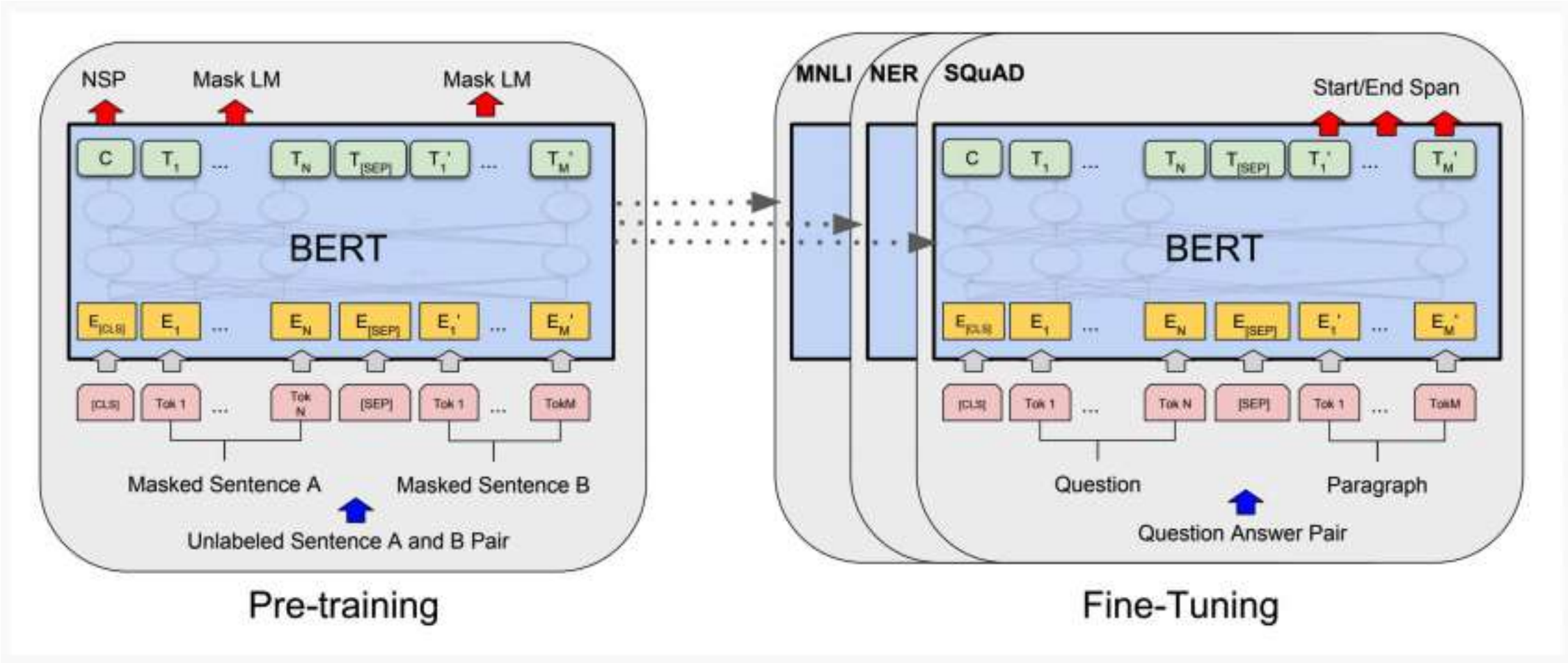
sentence A = The man went to the store
sentence B = He bought a gallon of milk
Label = IsNextSentence

sentence A = The man went to the store
sentence B = Penguins are flightless
Label = NotNextSentence

Pre-trained model details

- Data: Wikipedia (2.5B words) + BookCorpus (800M words)
- Batch Size: 131,072 words (1024 sequences * 128 length or 256 sequences * 512 length)
- Training Time: 1M steps (~40 epochs)
- Optimizer: AdamW, 1e-4 learning rate, linear decay
- `BERT-Base`: 12-layer, 768-hidden, 12-head
- `BERT-Large`: 24-layer, 1024-hidden, 16-head
- Trained on 4x4 or 8x8 TPU slice for 4 days

Fine-tuning Procedure



Representing sentences

- How can we use BERT to compare pairs of sentences to see if they mean the same or similar things?
- How do we get from a BERT encoding of a sequence of tokens to a representation of a sentence?
- What are the options?

Reimers and Gurevych (2019) present Sentence-BERT (SBERT), a modification of the BERT netowrk using siamese and triplet networks that they claim is able to derive semantically meaningful sentence embeddings. Once you have read the paper, consider the following questions.

1. For a pair regression task, the standard BERT set-up requires pairs of sentences to be presented as input to the encoder network. Why is this set-up unfeasible if you want to find the most similar sentences in a collection?

2. What have other researchers done to overcome this problem with using BERT?

3. What is the SBERT strategy?

4. What do you understand by the term Siamese network structure?

5. What are the different pooling strategies that the authors experiment with? What works best?

6. Outline the 4 evaluation tasks used for semantic textual similarity.

7. Why would Spearman's rank correlation coefficient be better than Pearson's product-moment correlation coefficient when comparing ratings of semantic textual similarity?

8. What are the different objective functions that the authors experiment with? When is each used?

9. How is the SentEval evaluation different to the previous evaluation experiments?

10. What are the main conclusions of the paper? Are you convinced?

Q1

- For a pair regression task, the standard BERT set-up requires pairs of sentences to be presented as input to the encoder network. Why is this set-up unfeasible if you want to find the most similar sentences in a collection?

Q2

- What have other researchers done to overcome this problem with using BERT?

Q3

- What is the SBERT strategy?

Q4

- What do you understand by the term Siamese network structure?

Q5

- What are the different pooling strategies that the authors experiment with? What works best?

Q6

- Outline the 4 evaluation tasks used for semantic textual similarity?

Q7

- Why would Spearman's rank correlation coefficient be better than Pearson's product-moment correlation coefficient when comparing ratings of semantic textual similarity?

Q8

- What are the different objective functions that the authors experiment with? When is each used?

Q9

- How is the SentEval evaluation different to the previous evaluation experiments?

Q10

- What are the main conclusions of the paper? Are you convinced?

Further reading

- Devlin et al. (2019): Pre-training of Deep Bidirectional Transformers for Language Understanding in NAACL 2019, https://www.aclweb.org/anthology/N19-1423/
- Peters et al. (2018): Deep contextualised word representations in Proceedings of NAACL 2018 https://arxiv.org/pdf/1802.05365.pdf
- Peters et al. (2018): Dissecting Contextual Word Embeddings: Architecture and Representation in Proceedings of EMNLP 2018 https://www.aclweb.org/anthology/D18-1179.pdf
- Reimers et al. (2019):
- Vashwani et al. (2017): Attention is all you need in Proceedings of NIPS 2017
- Yang et al. (2020) :

(https://arxiv.org/pdf/2004.12297.pdf)