PyPharma NLP Workshop 2019: Introduction to Biomedical NLP

Diego Saldana, Data Scientist at Roche

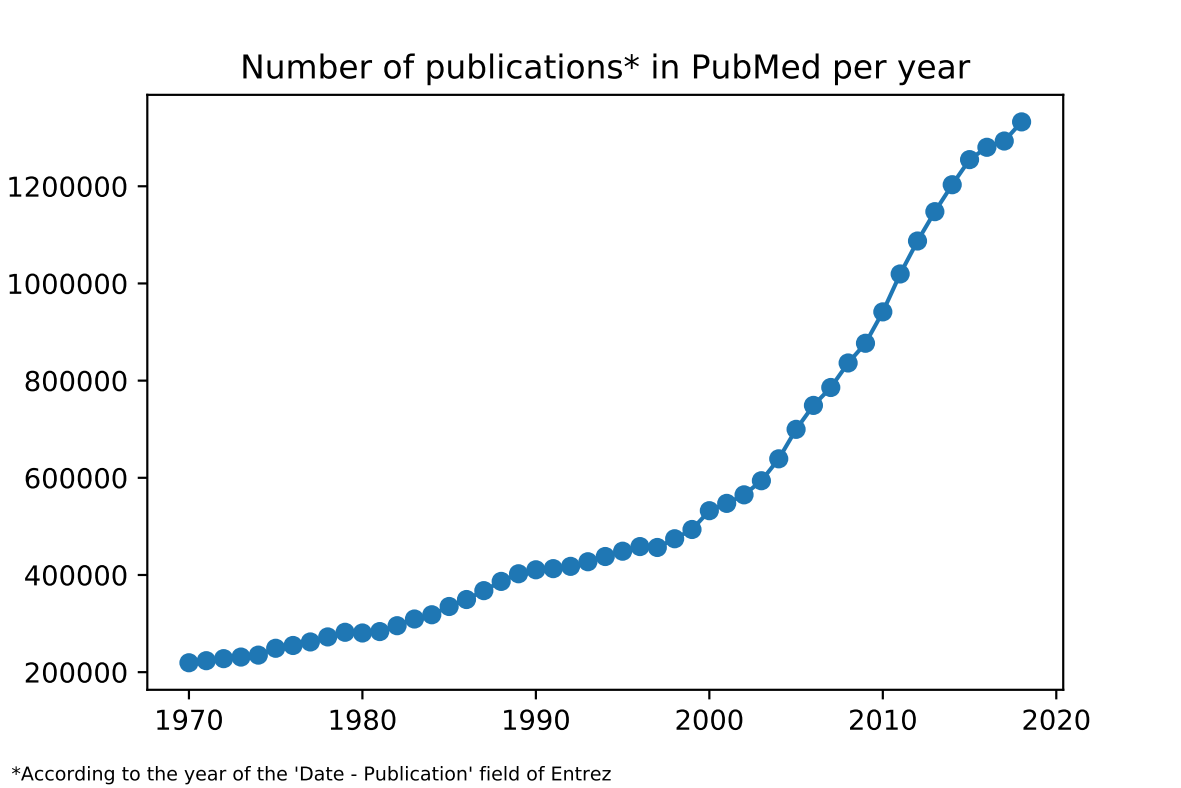
November 2019

PyPharma NLP Workshop

- The objective is to introduce the audience to recent progress in biomedical natural language processing.
- In particular, we will focus on self-attention based models: BioBERT.
- Familiarize the audience with BioBERT and the tasks that can be performed: Classification, Named Entity Recognition, Relation Extraction, Question Answering.
- We will emphasize practical use of available open source tools as a gateway towards deeper understanding through further reading on the topic by attendees.

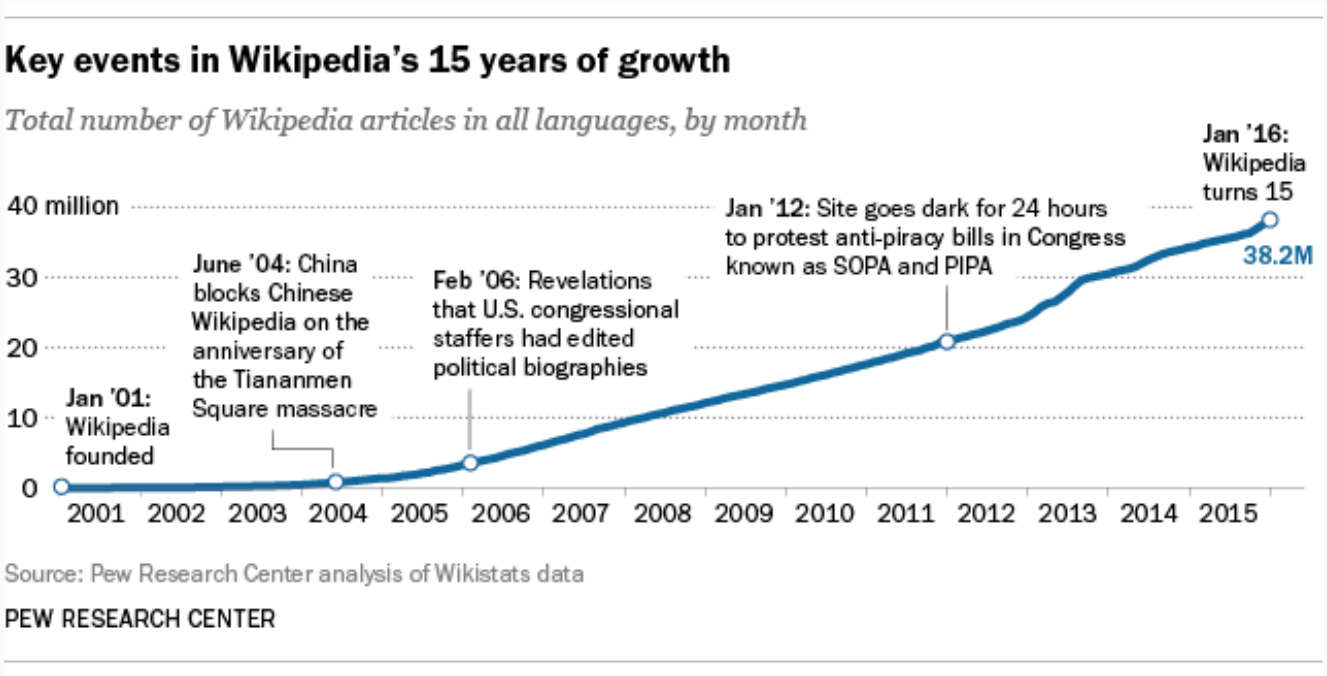
The Information Flood (1/3)

- Most of the information out there is in the form of natural language: scientific papers, clinical notes, social media, textbooks, lectures, websites.



The Information Flood (2/3)

- Most of the information out there is in the form of natural language: scientific papers, clinical notes, social media, textbooks, lectures, websites.



The Information Flood (3/3)

- Most of the information out there is in the form of natural language: scientific papers, clinical notes, social media, textbooks, lectures, websites.
- This information is potentially very useful but cannot readily be used programmatically and stored in databases, searched, or analyzed.
- As a result this valuable information is "locked into a vault" until a human reads it, structures it and puts it into some database.
- And even when that happens, the scope in which the data can be used is usually limited and chosen by the extractors.
- How can machines help?

Humans vs. Machines (1/2)

What are some examples of tasks can computers perform well in 2019?

- Categorizing documents (e.g. automatically assigning MeSH headings to PubMed abstracts)
- Extracting entities from text (e.g. extracting Drugs, Diseases from PubMed abstracts)
- Extracting relations from text (e.g. extracting Adverse Events from PubMed abstracts)
- Answering simple questions based on a small amount of context (e.g. "Which drug should be used as an antidote in benzodiazepine overdose?")

Humans vs. Machines (2/2)

- Machines and humans have different strengths and weaknesses when processing text.

Table 5. IAA scores between the annotators over the *ADE-seed-set1* corpus containing 50 documents. Enumerations related to dosages are zeroes since no dosage information was annotated during this round.

Annotators	**Entity (*exact match*)**			**Entity (*partial match*)**		
	Drug	**Adverse effect**	**Dosage**	**Drug**	**Adverse effect**	**Dosage**
1 and 2	0.76	0.66	0.00	0.82	0.86	0.00
1 and 3	0.28	0.43	0.00	0.38	0.55	0.00
2 and 3	0.29	0.40	0.00	0.38	0.51	0.00

Annotators	Relation (*exact entity match with exact relation*)		Relation (*partial entity match with exact relation*)	
	Drug-adverse effect	Drug-dosage	Drug-adverse effect	Drug-dosage
1 and 2	0.64	0.00	0.79	0.00
1 and 3	0.14	0.00	0.37	0.00
2 and 3	0.10	0.00	0.37	0.00

Humans vs. Machines (3/2)

- Machines and humans have different strengths and weaknesses when processing text.
- Machines in particular are capable of processing vast amounts of text in a very short period of time in a very consistent way and performing simple tasks.
- Humans are take much more time to process text and are less consistent, however they are capable of much more complex reasoning and understanding.

Some Natural Language Processing Tasks

Language Modelling

A language model assigns probabilities to sequences of tokens, where tokens t can be words, characters, sub-words, etc:

$$P\Big(t_1, t_2, t_3, ..., t_N\Big).$$

Take four sentences:

- "The dog ran after the cat."
- "The dog ran after the tiger."
- "The stone ran after the tiger."
- "Tiger stone the after ran."

Clearly, each subsequent sentence is less probable than the next. A good language model should assign probabilities to these sentences accordingly.

Document Classification (1/2)

A document classifier assigns one or more class labels to a document.

Examples of document classification include:

- Predicting MeSH headings for PubMed abstracts.
- Annotating PubMed abstracts according to the Hallmarks of Cancer (HoC).
- Classifying sentences as having mentions of Adverse Drug Reactions (ADRs) or not.

Document Classification (2/2)

A document classifier assigns one or more class labels to a document.

Luteolin (10 mg/kg/d) significantly reduced the volume and the weight of solid tumors in prostate xenograft mouse model, indicating that luteolin inhibited tumorigenesis by targeting angiogenesis.	Inducing Angiogenesis
Arsenic exposure by 10 weeks and after also induced marked and sustained increases in colony formation, indicative of the loss of contact inhibition, and increased invasiveness, both cancer cell characteristics.	Evading growth suppressors Activating invasion and metastasis
Epidermal growth factor was present in 12.7% of normal ovaries, with a range 0.030-0.533 ng/mg DNA, and in 31.8% of benign ovarian tumours, with a range 0.1335-2.080 ng/ml DNA.	Sustaining proliferative signaling
Together, these results show that an antisense gene for SV40-T antigen can efficiently block the cell proliferation and the cell immortalization of VA-13 cells.	Sustaining proliferative signaling Enabling replicative immortality
These results suggest that aberrant CDK6 expression or activation that is frequently observed in human tumors can contribute through NF-κB to chronic inflammation and neoplasia.	Tumor promoting inflammation
G1896A in the precore region and C1653T mutation in the X region of genotype C2 HBV are important risk factors for HCC development.	Genomic instability and mutation

Named Entity Recognition

A Named Entity Recognizer extracts entities from a document.

- Examples of potential named entities include: drugs, diseases, genes, mutations, proteins, etc.
- One can extract the entities themselves as well as the boundaries. That is, the start and the end of the entity mention in the text.
- One can also subsequently perform Named Entity Resolution: Mapping the extracted entity to a concept in a standardized vocabulary.

Relation Extraction

A Relation Extractor extracts two or more entities and a relationship between them. Examples of potential relations to extract include:

- A drug inducing an adverse reaction.
- A gene mutation inducing resistance to a drug.
- A gene regulating a biological pathway.
- A drug targetting a protein.
- A protein interacting with another protein.
- etc

Question Answering

A Question Answering system provides an answer to a question given some context. That is, a set of documents. An example question would be:

Context: Orteronel is an investigational, partially selective inhibitor of CYP 17,20-lyase in the androgen signalling pathway, a validated therapeutic target for metastatic castration-resistant prostate cancer.

...

Question: Orteronel was developed for treatment of which cancer?
Answer: castration-resistant prostate cancer

Some Highlights in the History of NLP

Bag-Of-Words models

Bag-Of-Words (BOW) models ignore context and ordering of the words in a sentence and model them as an unordered collection of words (Harris 1981).

- Often the words are pre-processed: lowercasing, stemming, removing stop words, tf-idf, etc.
- Early uses of bags of words were notably in spam filtering.
- We can build a word-document-matrix with documents as rows and words as columns.
- Note that such a matrix is very sparse.

Sentence: The dog was barking at the other dog.
BOW representation: dog: 2, bark: 1, other: 1, all other words: 0

Latent Semantic Analysis

Applying Singular Value Decomposition to a Word-Document-Matrix is referred to as Latent Semantic Analysis (Dumais 2004).

- We obtain latent variables representing a space in which words having similar meanings are closer to each other than words having very distant meanings.
- The model can thereby deal with synonyms, antonyms, singular-plural forms of words, etc.
- Similar documents are also close to each other in latent space.
- An early method for distributional semantics.
- It's also a form of dimensionality reduction.

Latent Dirichlet Allocation (1/2)

Latent Dirichlet Allocation (Blei, Ng, and Jordan 2003) is a bayesian approach that models the document generating process as a probabilistic graphical model. We have:

- A distribution of words over topics.
- A distribution of topics over documents.
- Each document is a collection of topic-word pairs drawn from these distributions.

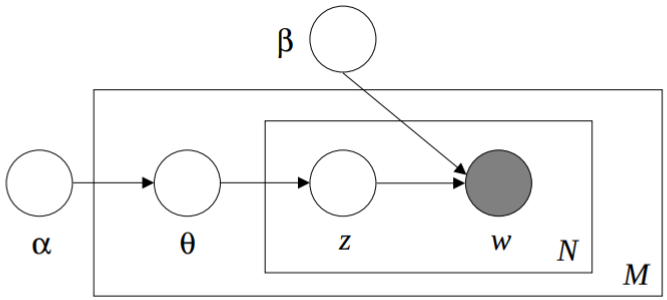


Figure 1: Graphical model representation of LDA. The boxes are "plates" representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

Latent Dirichlet Allocation (2/2)

Latent Dirichlet Allocation (Blei, Ng, and Jordan 2003) is a bayesian approach that models the document generating process as a probabilistic graphical model. We have:

- Commonly used for topic modelling.
- Note that the number of topics must be pre-specified prior to inference.
- Topics have no automatically assigned names.

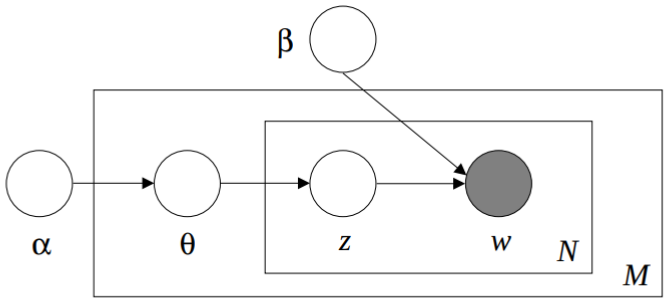


Figure 1: Graphical model representation of LDA. The boxes are "plates" representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

Word2Vec (1/4)

Word2vec (Mikolov et al. 2013) is a method to produce word embeddings. Word embeddings allow us to project words into a space that has some interesting properties.

- Based on the Skip-gram model proposed by Mikolov in the original paper, which models the probability of a word given the surrounding words (ordering is not important) using a single layer neural network.
- Words having similar meanings are close to each other, and distant from words having very different meanings.
- Word arithmetic is possible. For example one may do the operation

$$vec(\text{"}Madrid\text{"}) - vec(\text{"}Spain\text{"}) + vec(\text{"}France\text{"}) \sim vec(\text{"}Paris\text{"})$$

Word2Vec (2/4)

Word2vec (Mikolov et al. 2013) is a method to produce word embeddings. Word embeddings allow us to project words into a space that has some interesting properties.

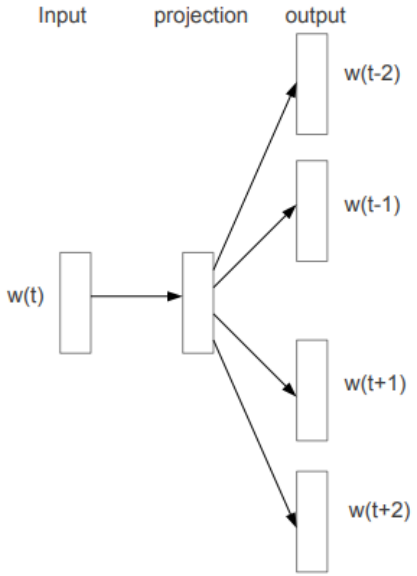


Figure 1: The Skip-gram model architecture. The training objective is to learn word vector representations that are good at predicting the nearby words.

Word2Vec (3/4)

Word2vec (Mikolov et al. 2013) is a method to produce word embeddings. Word embeddings allow us to project words into a space that has some interesting properties.

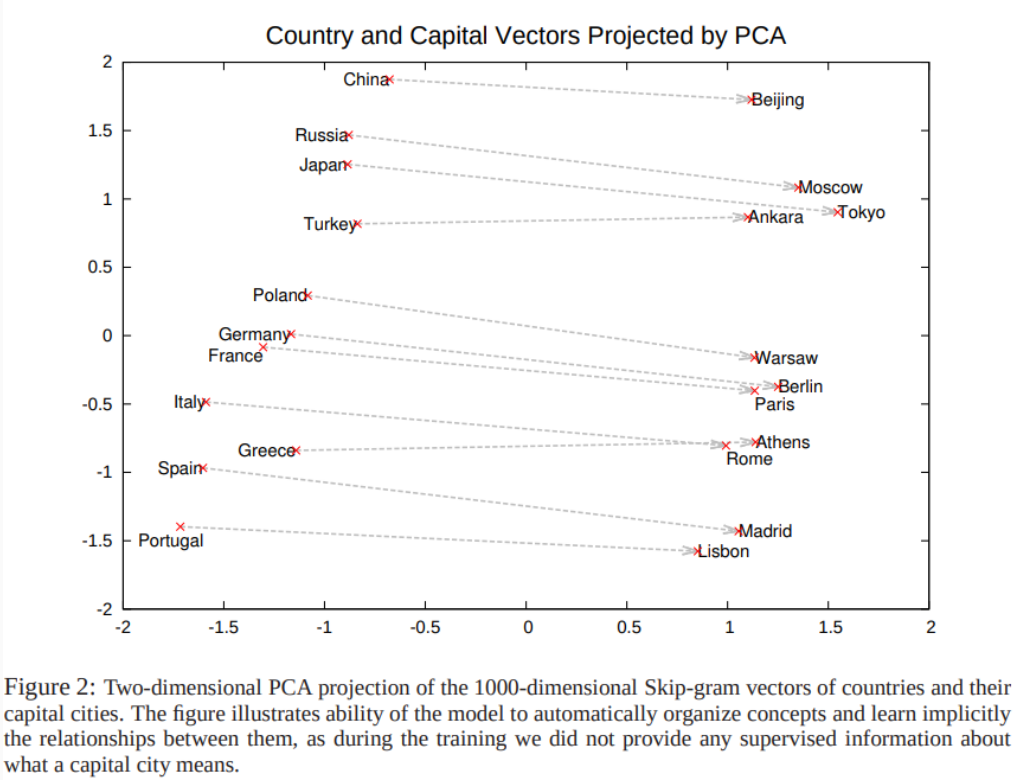


Figure 2: Two-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training we did not provide any supervised information about what a capital city means.

Word2Vec (4/4)

Word2vec (Mikolov et al. 2013) is a method to produce word embeddings. Word embeddings allow us to project words into a space that has some interesting properties.

- Pyysalo et al. (2013) fit a word2vec model on PubMed, PubMec Central, and biomedical articles in wikipedia and pubblished the resulting biomedical word-embeddings.
- Biomedical embeddings work better on biomedical tasks due to the domain specific content being more more similar to the content in which the algorithms are applied.
- For example, they have led to better performance when classifying sentences as containing ADRs or not (Saldana 2018).

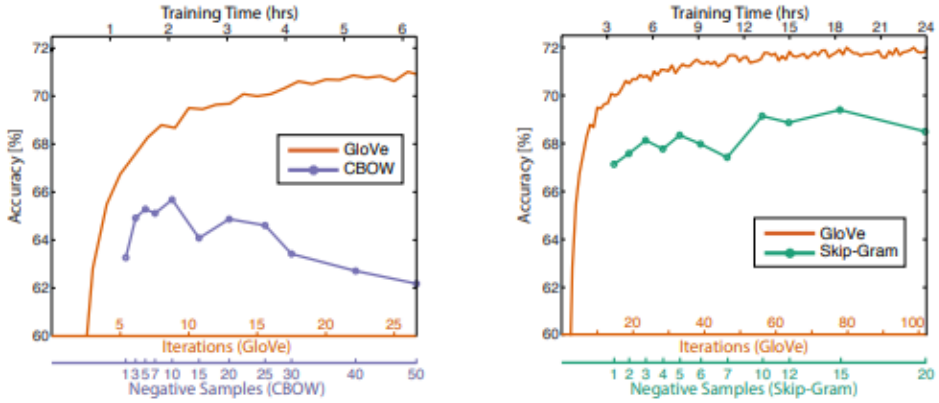
GloVe (1/2)

Global Vectors for word representation (GloVe) described by Pennington, Socher, and Manning (2014) that use a log-bilinear regression model to model word co-occurrences within a context window.

- It was designed to have the attractive properties that enable word arithmetic operations seen in word2vec.
- The authors showed that GloVe can outperform word2vec in the word analogy task.
- Is also easier to parallelize by virtue of its implementation, allowing it to be trained in much larger datasets more easily.
- Like in word2vec, the word vectors obtained with GloVe are fixed and do not change with context.

GloVe (2/2)

Global Vectors for word representation (GloVe) described by Pennington, Socher, and Manning (2014) that use a log-bilinear regression model to model word co-occurrences within a context window.



(a) GloVe vs CBOW (b) GloVe vs Skip-Gram

Figure 4: Overall accuracy on the word analogy task as a function of training time, which is governed by the number of iterations for GloVe and by the number of negative samples for CBOW (a) and skip-gram (b). In all cases, we train 300-dimensional vectors on the same 6B token corpus (Wikipedia 2014 + Gigaword 5) with the same 400,000 word vocabulary, and use a symmetric context window of size 10.

ELMO

ELMO is a model based on pre-training of bi-directional language models (LSTMs) to produce context dependent word vectors (Peters et al. 2018).

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

Table 4: Nearest neighbors to "play" using GloVe and the context embeddings from a biLM.

BERT

BERT (Peters et al. 2018) is a purely attentional model based on bi-directional transformers to produce context dependent word vectors similar to ELMO.

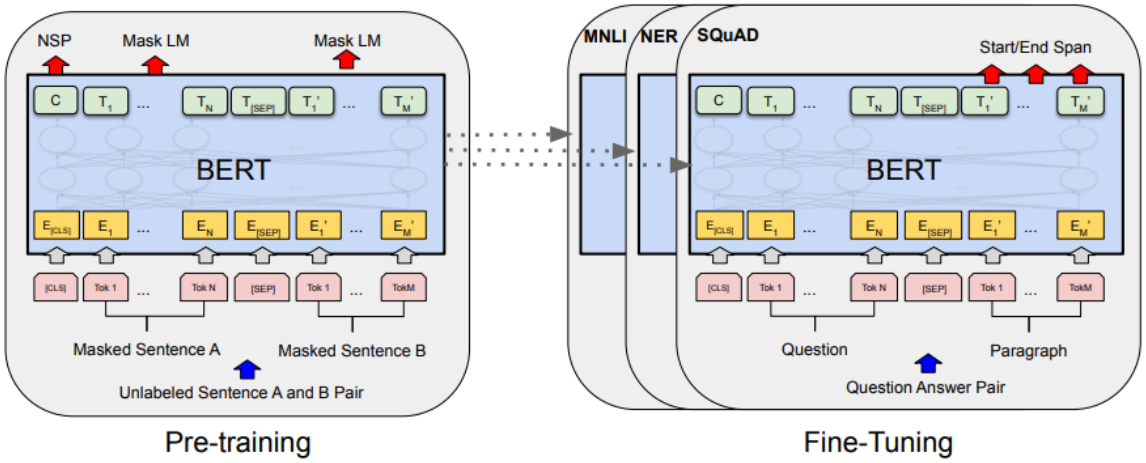
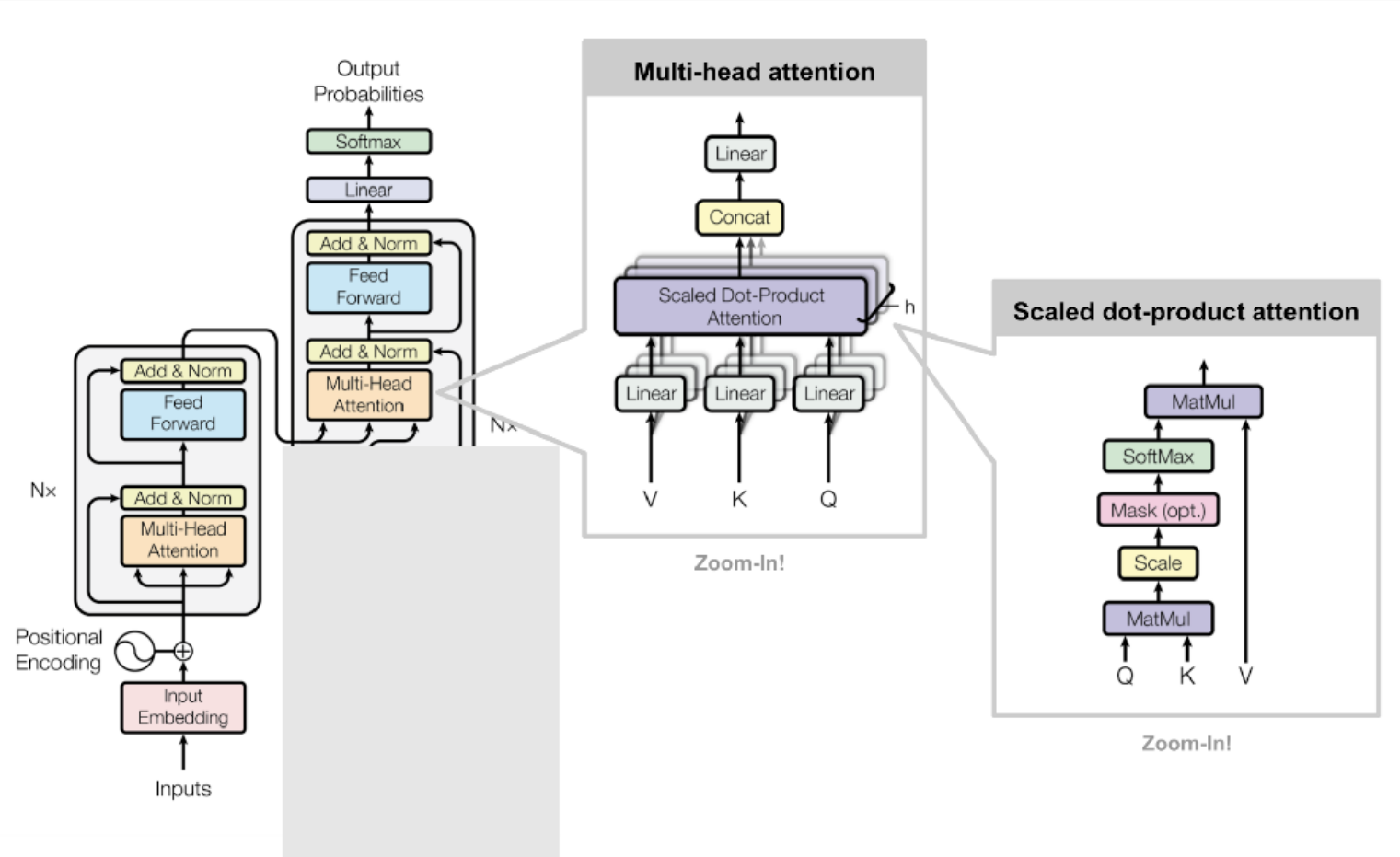


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. `[CLS]` is a special symbol added in front of every input example, and `[SEP]` is a special separator token (e.g. separating questions/answers).

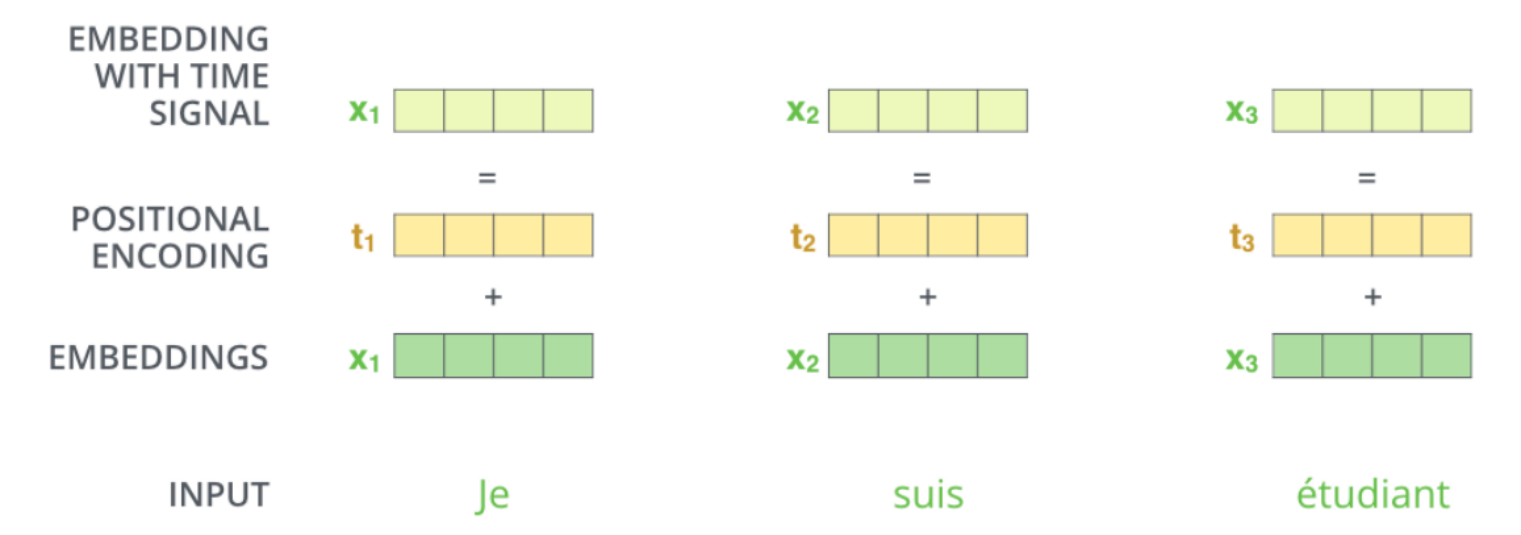
The Transformer Architecture

What follows is from:
http://jalammar.github.io/illustrated-transformer/

The Transformer Architecture



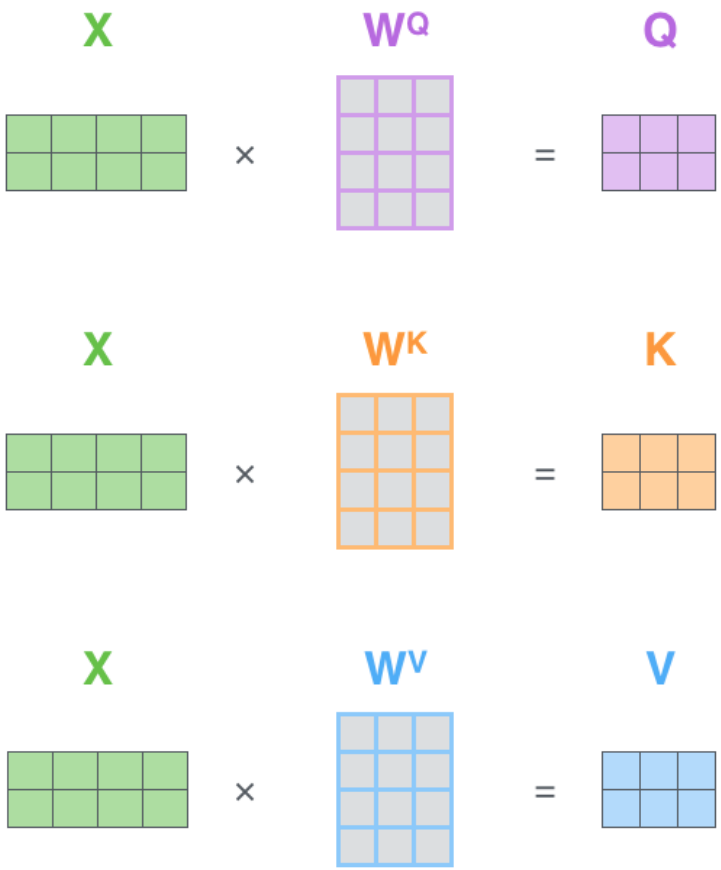
Embedding + Positional Embedding (1/2)



Embedding + Positional Embedding (2/2)

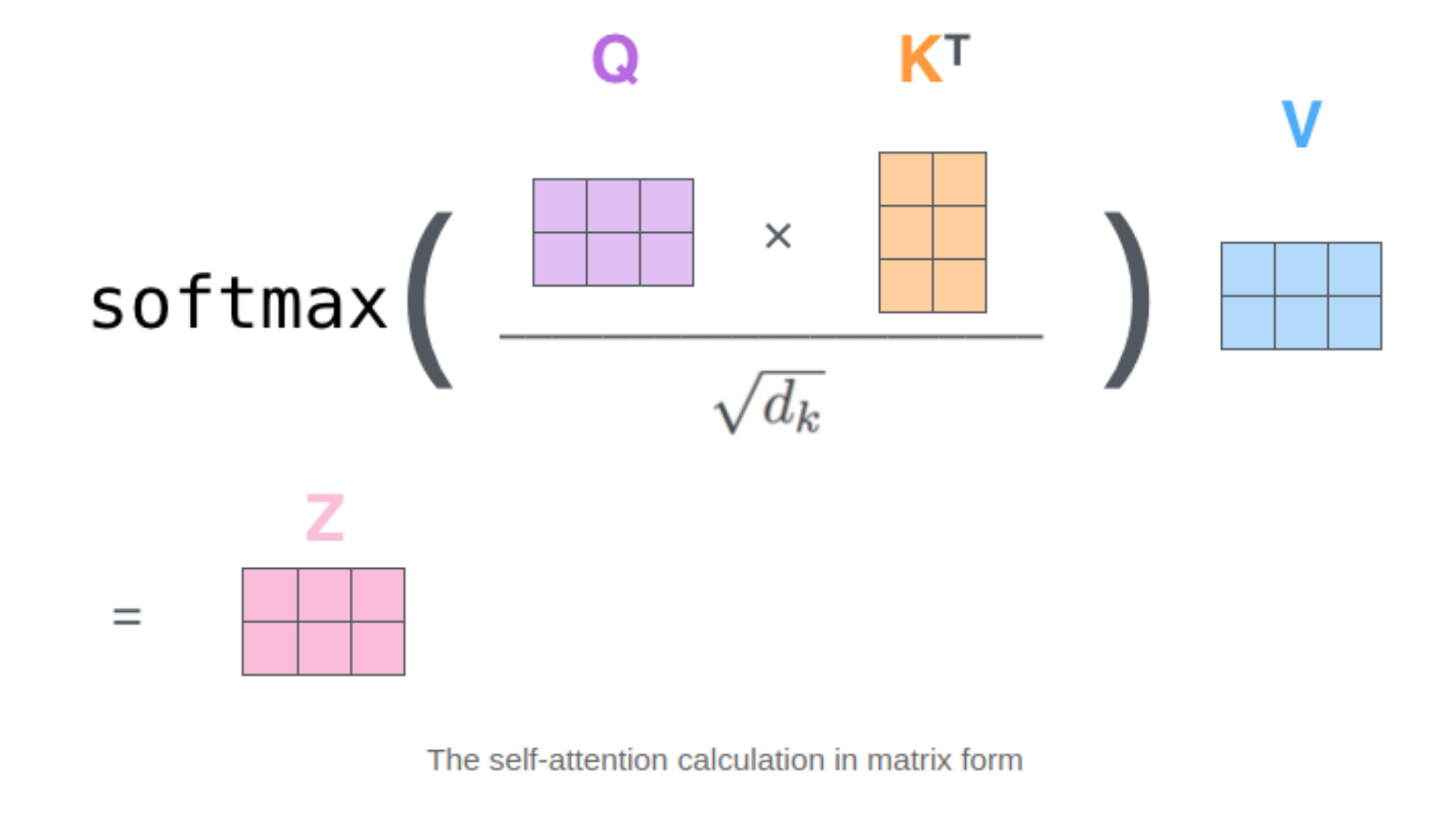
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}})$$

Queries, Keys, and Values



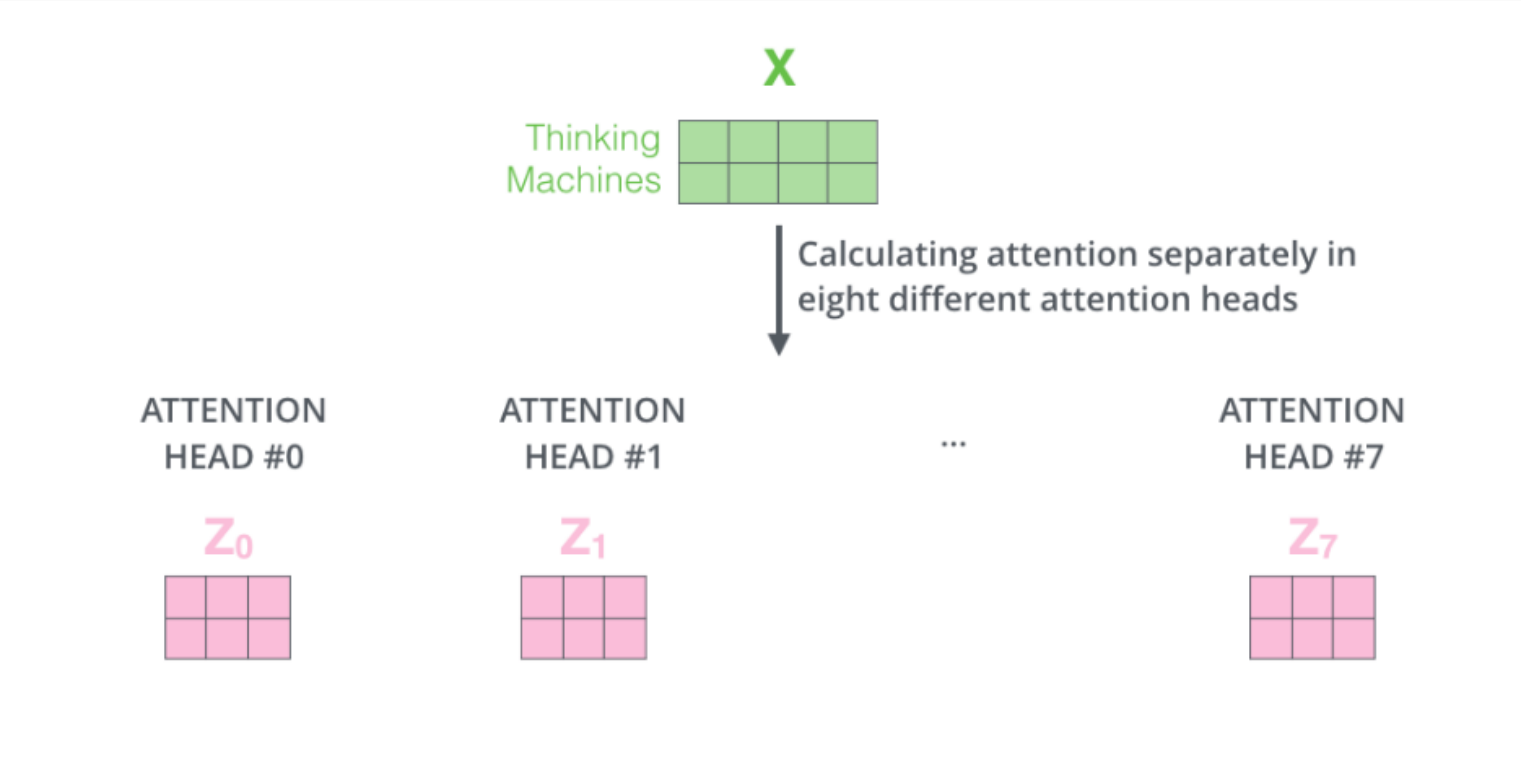
Every row in the X matrix corresponds to a word in the input sentence. We again see the difference in size of the embedding vector (512, or 4 boxes in the figure), and the q/k/v vectors (64, or 3 boxes in the figure)

Self-attention



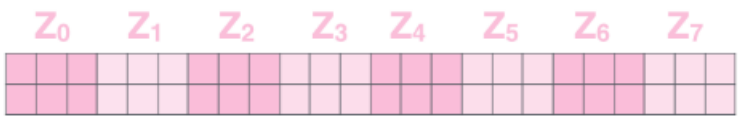
The self-attention calculation in matrix form

Multi-head self-attention



Concatenation

1) Concatenate all the attention heads

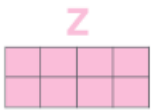


2) Multiply with a weight matrix W^O that was trained jointly with the model

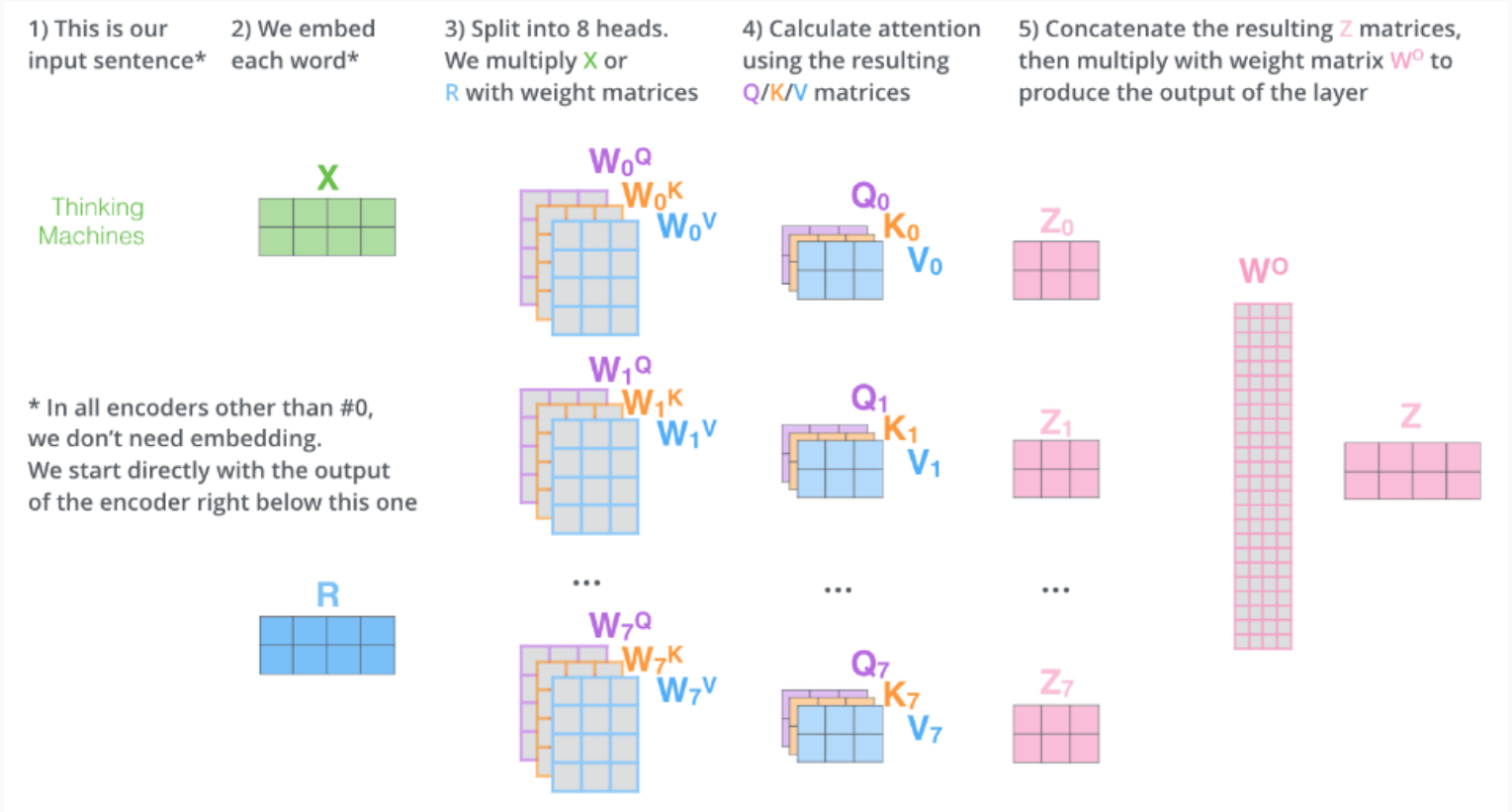
X

3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

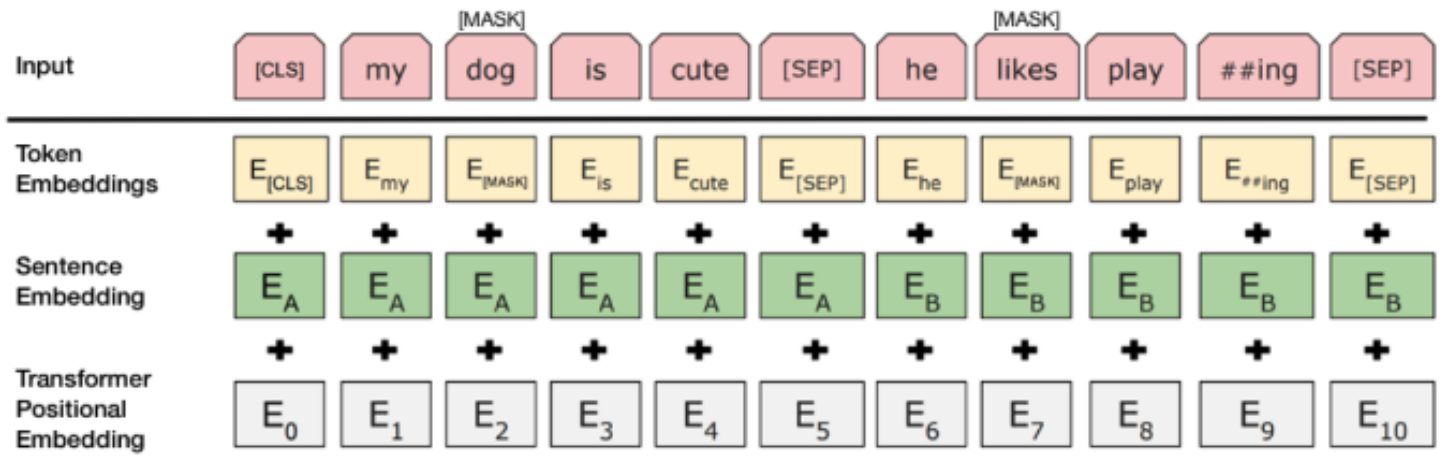
=



Overview


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
X
R
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one
W0Q
W0K
W0V
W1Q
W1K
W1V
W7Q
W7K
W7V
Q0
K0
V0
Q1
K1
V1
Q7
K7
V7
Z0
Z1
Z7
WO
Z

BERT Training



Source: BERT [Devlin et al., 2018], with modifications

References

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. "Latent Dirichlet Allocation." *J. Mach. Learn. Res.* 3 (March). JMLR.org: 993–1022. http://dl.acm.org/citation.cfm?id=944919.944937.

Dumais, Susan T. 2004. "Latent Semantic Analysis." *Annual Review of Information Science and Technology* 38 (1): 188–230. doi:10.1002/aris.1440380105.

Harris, Zellig S. 1981. *Distributional structure*. Springer.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. "Distributed Representations of Words and Phrases and Their Compositionality." In *Advances in Neural Information Processing Systems 26*, edited by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, 3111–9. Curran Associates, Inc. http://papers.nips.cc/paper/