IN4325

Neural IR

Arthur Câmara (WIS, TU Delft)
Slides by Claudia and Arthur

56 Full Papers.

- 4 contain “neural” or “deep" in title
- 5 contain “embedding” in title
- 4 contain “Transformers" or **BERT** in title
- 19 others papers with other keywords

~57% of papers with some type of neural IR on the title.

(Probably *****waaaay*** more, but less explicit)**

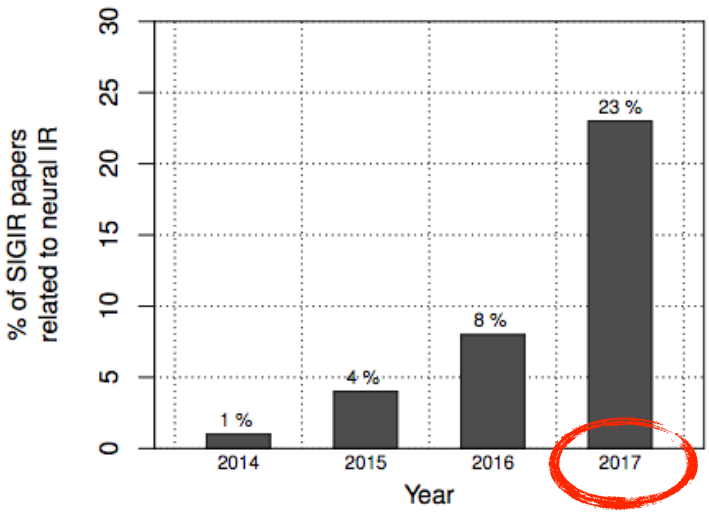
https://www.microsoft.com/en-us/research/wp-content/uploads/2017/06/fntir-neuralir-mitra.pdf

A great starting point for Neural IR - 123 pages of insights!
We follow some of it here.

Neural IR

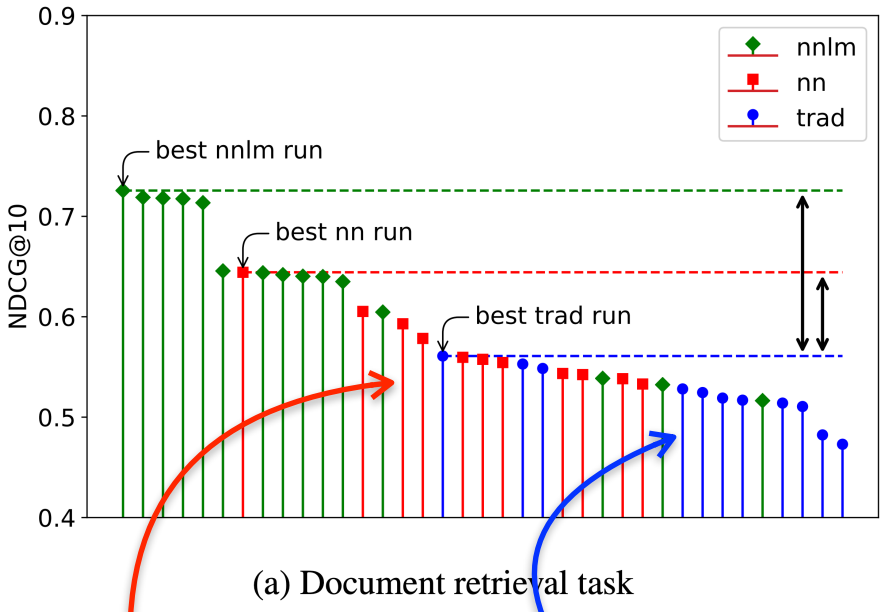
… is [redacted] taking over the IR research field

(yet)

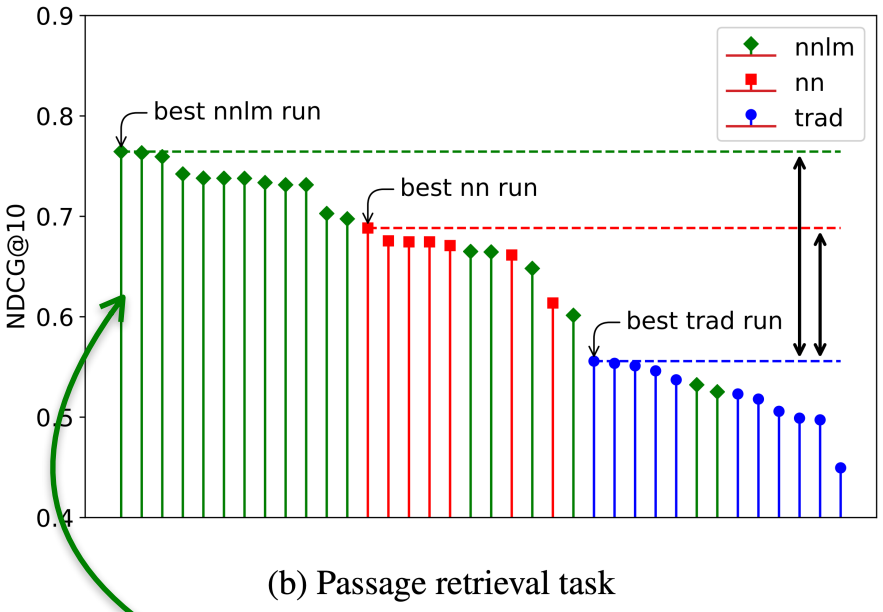


Covered on a high level: this topic by itself could take up all whole quarter of lectures. Unfortunately, we do not have the time.

What about 2020?



(a) Document retrieval task



(b) Passage retrieval task

Neural Nets (usually) outperform traditional methods

NN Language Models outperform everything else!

https://arxiv.org/abs/2003.07820v2

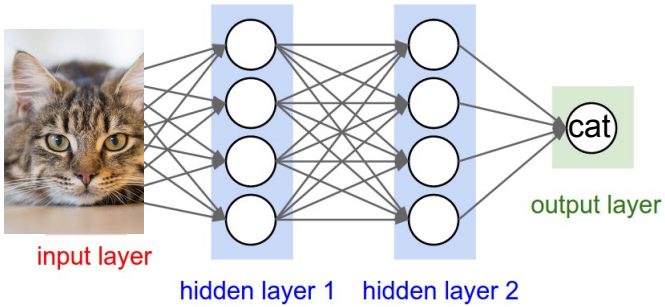
http://cs231n.github.io/neural-networks-1/

https://arxiv.org/pdf/1409.4842.pdf

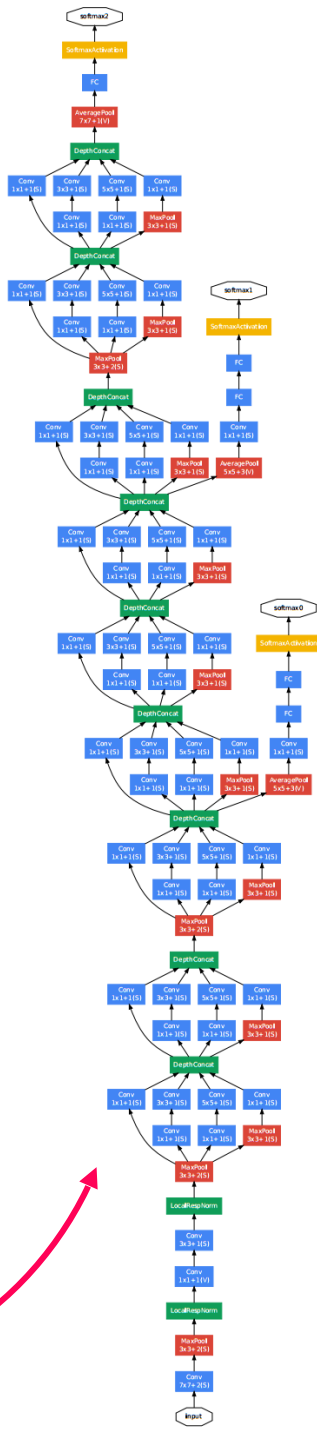
Overview

`NeuralIR` is the application of shallow or deep neural networks to IR tasks.

`NeuralIR` can make use of neural NLP techniques to enhance the retrieval Pipeline.



GoogLeNet (2014)



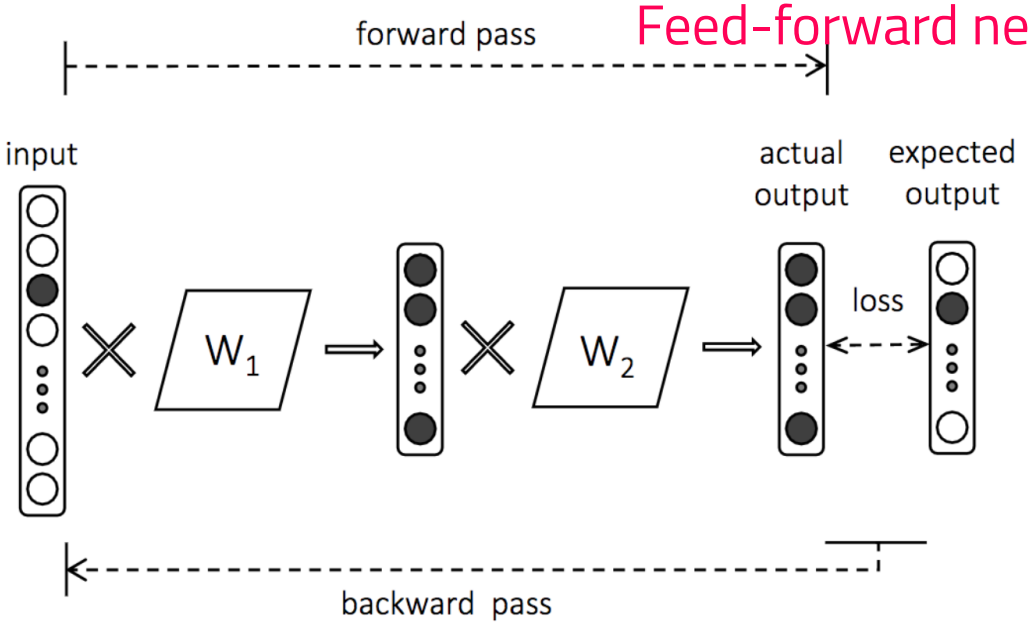
`NeuralIR` models contain thousands/millions of parameters. They require a large amount of training data.

(We will see how to help with this)

Instead of hand-crafting **features** (classic ML), we now handcraft NN **architectures**

https://www.microsoft.com/en-us/research/wp-content/uploads/2017/06/fntir-neuralir-mitra.pdf

Neural net basics



Feed-forward neural net with one hidden layer.

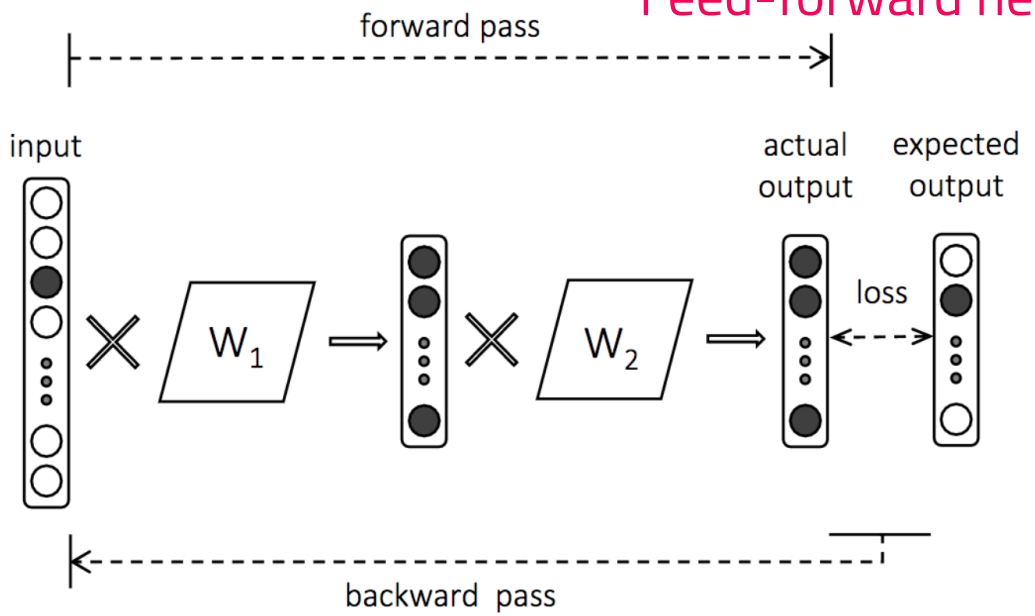
https://www.microsoft.com/en-us/research/wp-content/uploads/2017/06/fntir-neuralir-mitra.pdf

Initially: deep nets save us from expensive and elaborate feature engineering.

Now: lets engineer architectures and hardware/software that allows us to efficiently train deep nets.

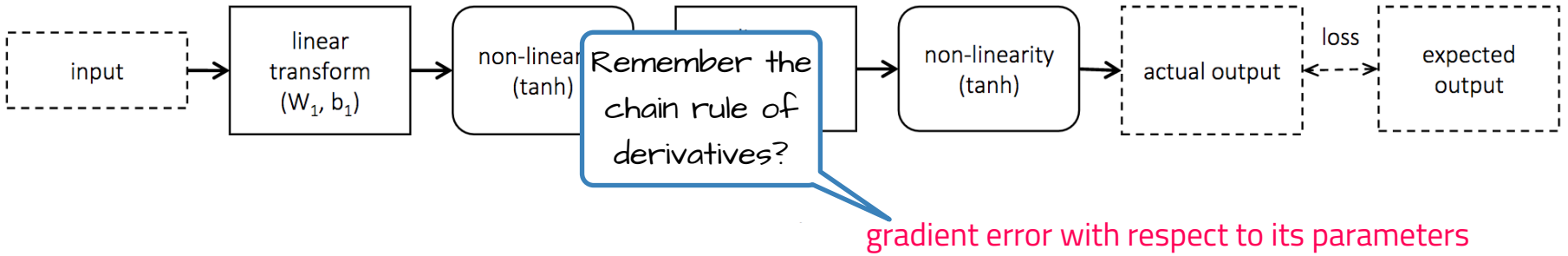
Neural net basics

Feed-forward neural net with one hidden layer.



$$\vec{y} = tanh(W_2 \cdot tanh(W_1 \cdot \vec{x} + \vec{b}_1) + \vec{b}_2)$$

Chain of computational steps.



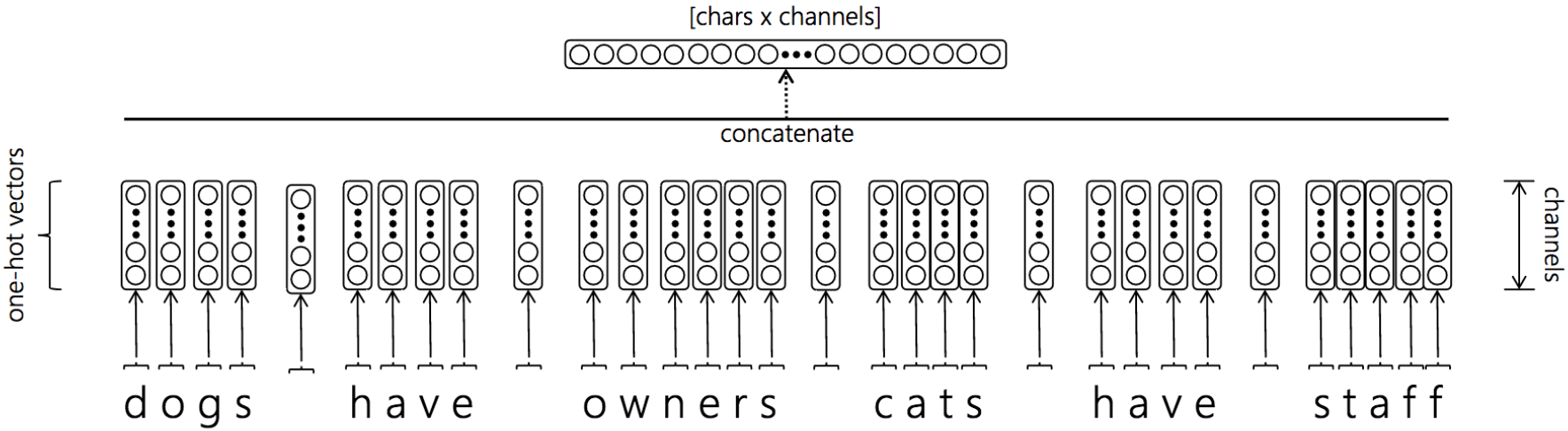
Text representations

Neural IR

https://www.microsoft.com/en-us/research/wp-content/uploads/2017/06/fntir-neuralir-mitra.pdf

Text input to deep neural nets

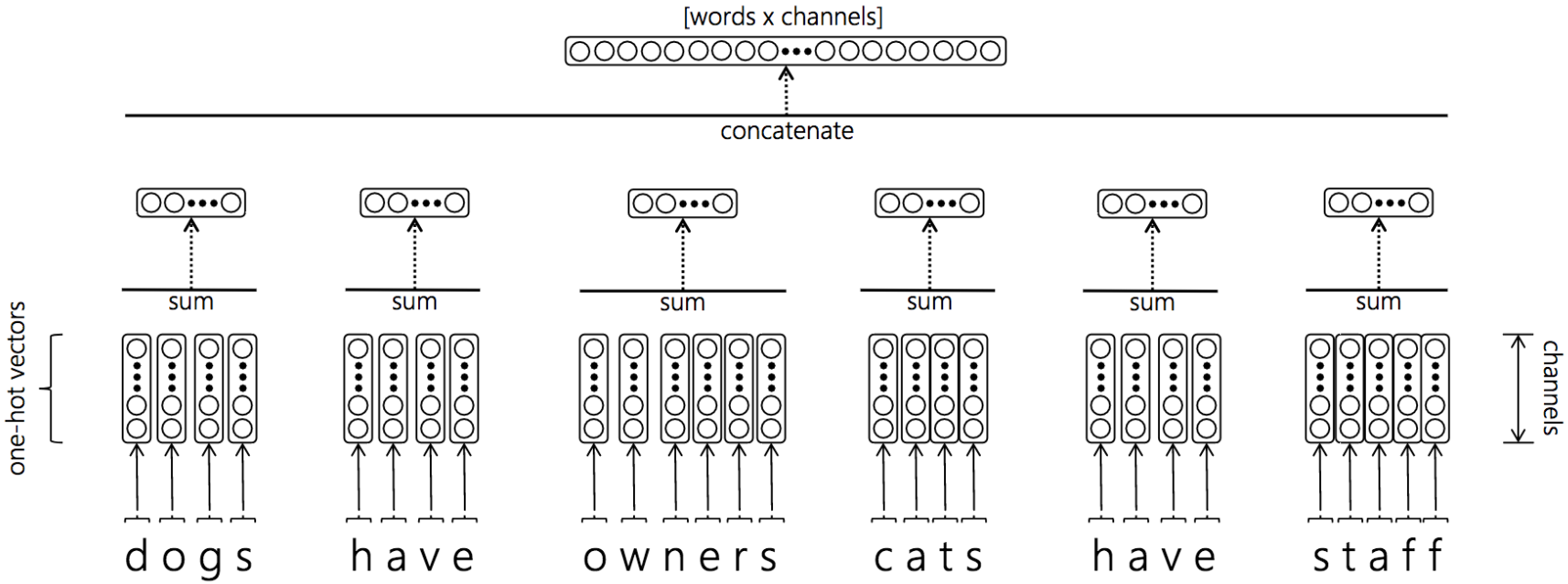
character-level input



https://www.microsoft.com/en-us/research/wp-content/uploads/2017/06/fntir-neuralir-mitra.pdf

Text input to deep neural nets

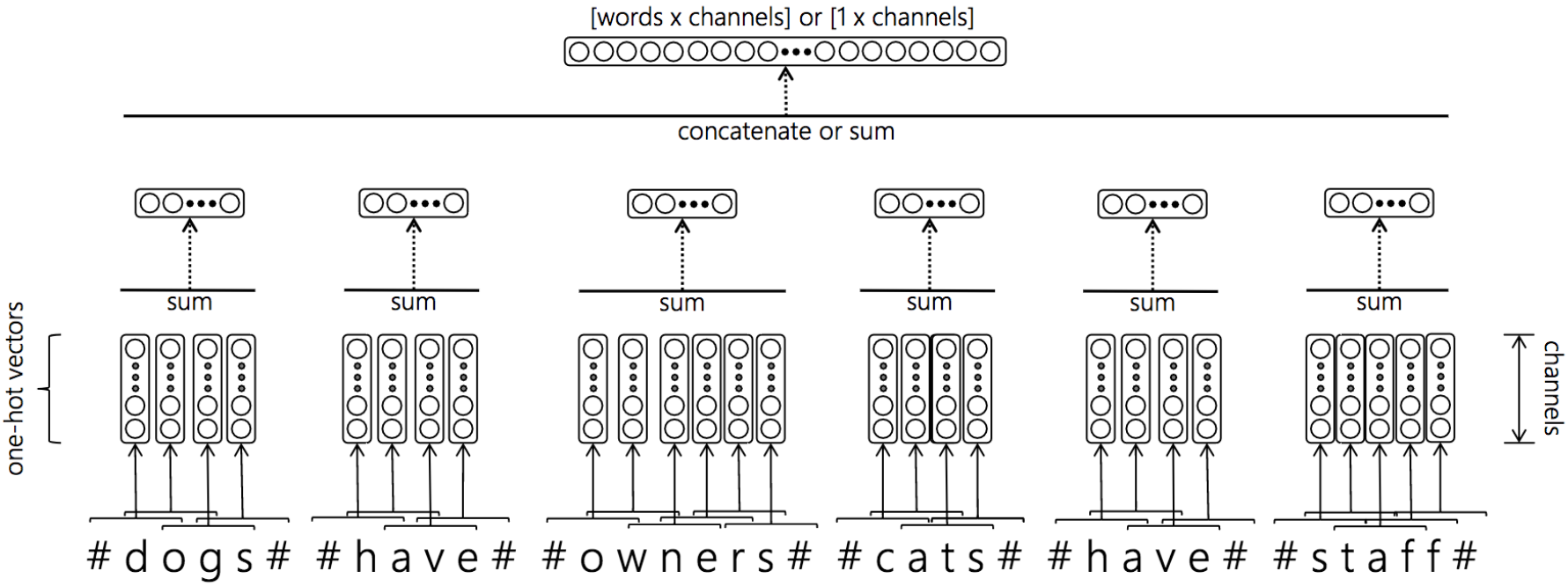
term-level input with bag-of-chars per term



https://www.microsoft.com/en-us/research/wp-content/uploads/2017/06/fntir-neuralir-mitra.pdf

Text input to deep neural nets

term-level input with bag-of-trigrams per term



Embeddings

... representation in a new space that should preserve the relationships between items of the original representation.

Dense vector representation.

Low-dimensional.

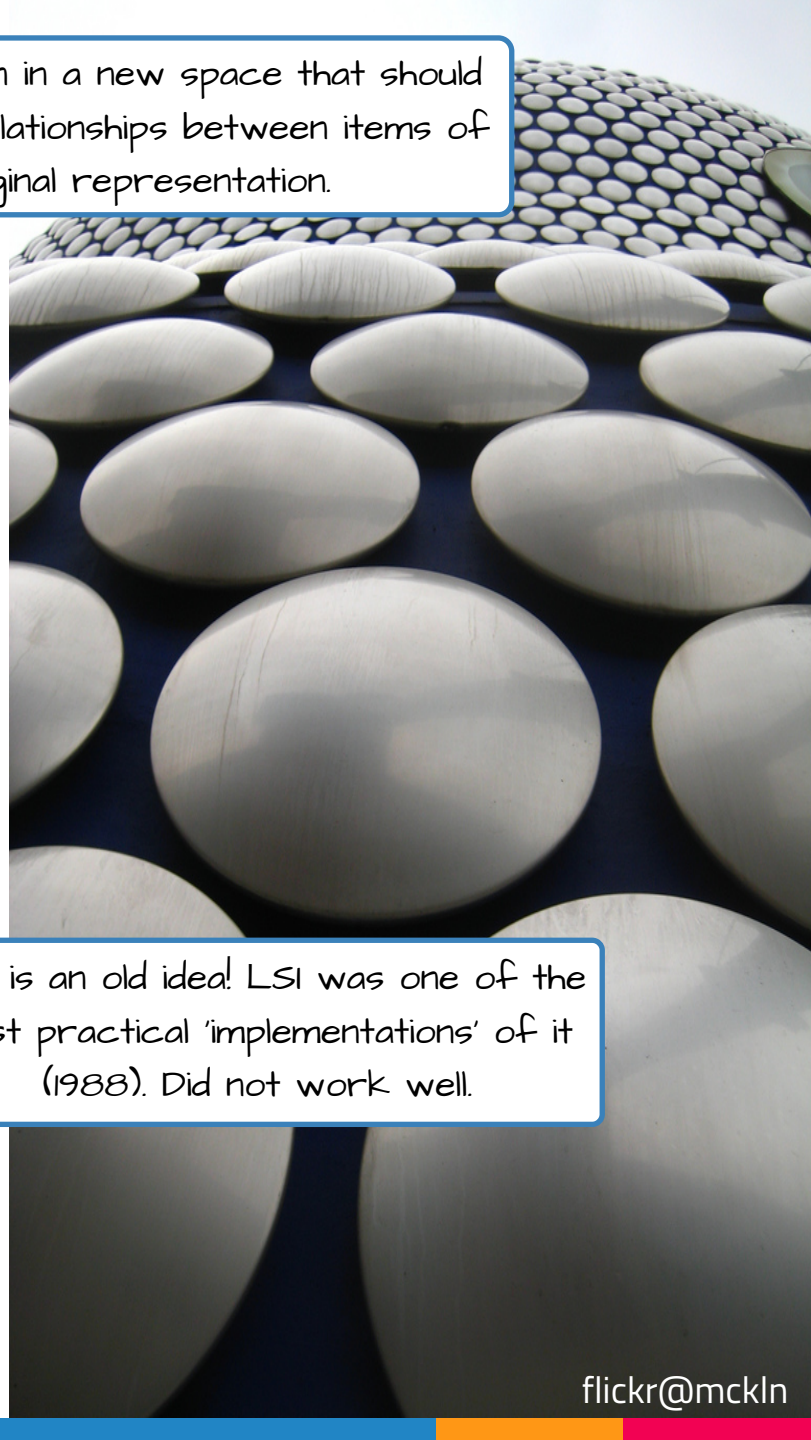
Learnt from data.

Distributed representation most often refers to learnt embeddings.

In today's NLP literature, the default encoding.

Motivated by the **distributional hypothesis** (Harris, 1954): "*a word is characterized by the company it keeps*" (Firth, 1957)

This is an old idea! LSI was one of the first practical 'implementations' of it (1988). Did not work well.

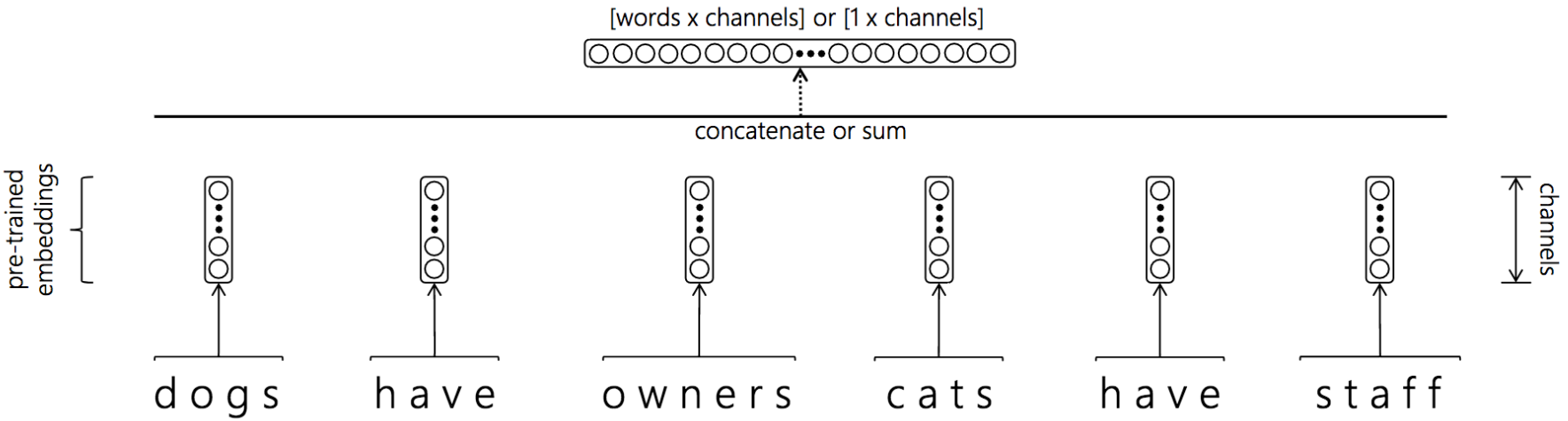


flickr@mckln

https://www.microsoft.com/en-us/research/wp-content/uploads/2017/06/fntir-neuralir-mitra.pdf

Text input to deep neural nets

term-level input with pre-trained word embeddings



GloVe:

- Pre-trained word vectors. This data is made available under the Public Domain Dedication and License v1.0 whose full text can be found at: http://www.opendatacommons.org/licenses/pddl/1.0/.
  - Wikipedia 2014 + Gigaword 5 (6B tokens, 400K vocab, uncased, 50d, 100d, 200d, & 300d vectors, 822 MB download): glove.6B.zip
  - Common Crawl (42B tokens, 1.9M vocab, uncased, 300d vectors, 1.75 GB download): glove.42B.300d.zip
  - Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB download): glove.840B.300d.zip
  - Twitter (2B tweets, 27B tokens, 1.2M vocab, uncased, 25d, 50d, 100d, & 200d vectors, 1.42 GB download): glove.twitter.27B.zip

Break! In the next part, we will cover some modern IR neural Architectures

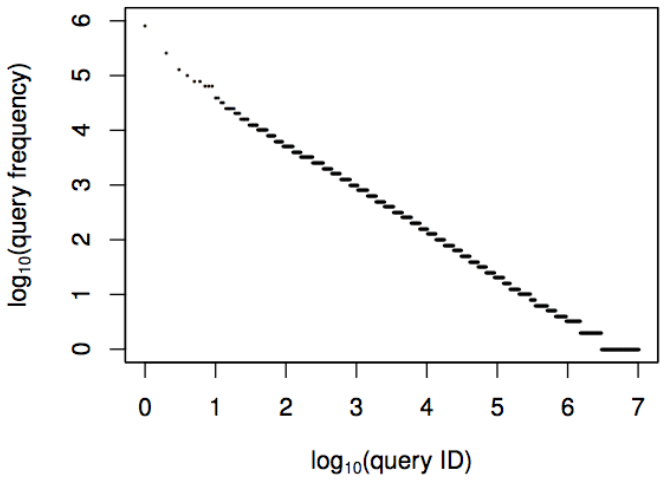
https://www.microsoft.com/en-us/research/wp-content/uploads/2017/06/fntir-neuralir-mitra.pdf

Neural IR architectures

https://www.microsoft.com/en-us/research/wp-content/uploads/2017/06/fntir-neuralir-mitra.pdf

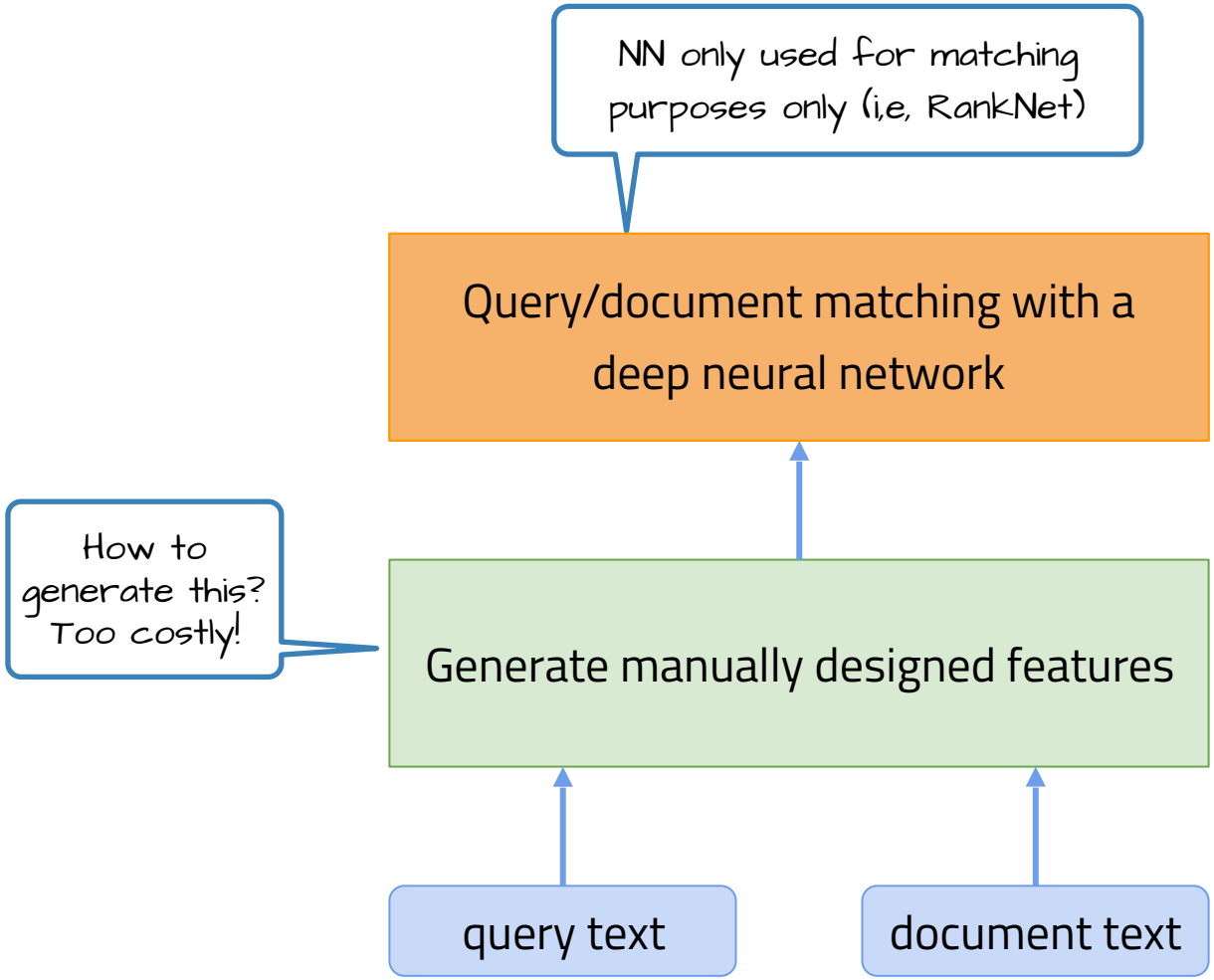
A good retrieval system should …

1. Have **semantic understanding** and enable **exact term matching** (*cat* and *feline* are similar query terms, but *hot dog* is not similar to *warm puppy*)
2. Be **robust** to rare **inputs**: remember the long tail!
3. Be **robust** to **corpus variance**:
   BM25 out-of-the-box is a good ranking model for all domains, neural models usually require a lot of training (training on domain X data and testing on domain Y data can go wrong ….)

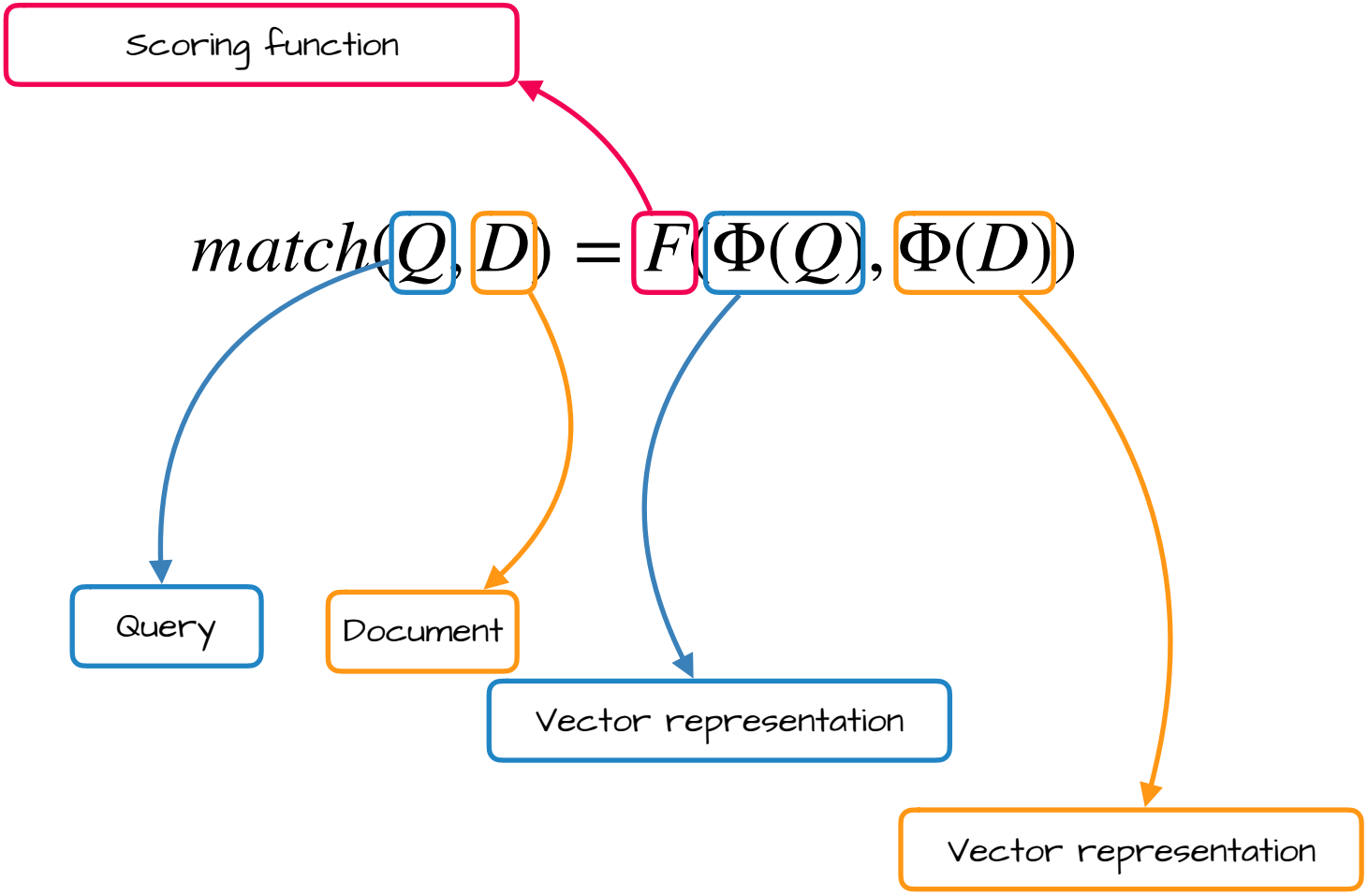


4. Be **robust** to **variable length input**
   (classic IR has document length normalization)
5. Be **sensitive** to **context** (e.g. location)

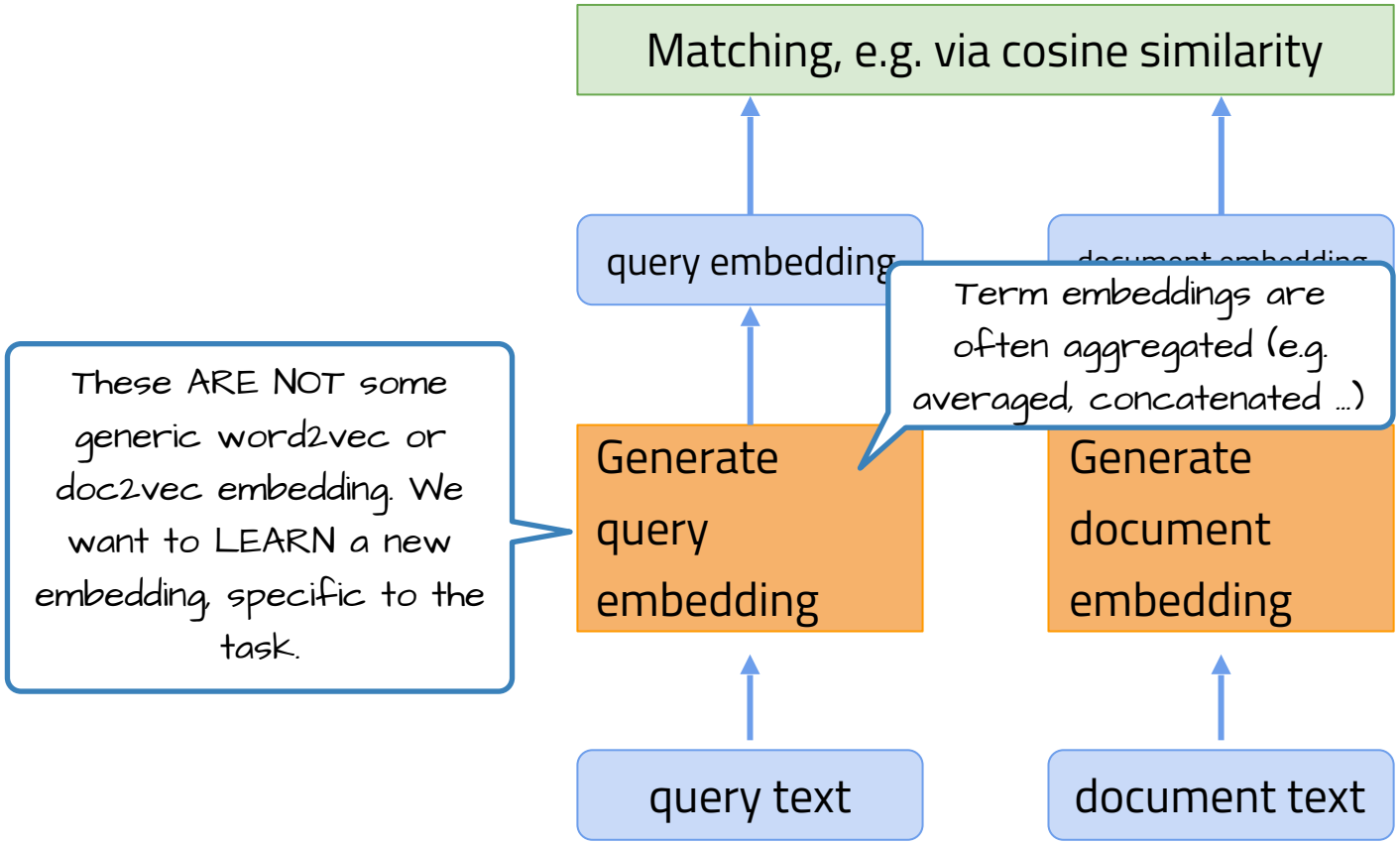
Learning to rank



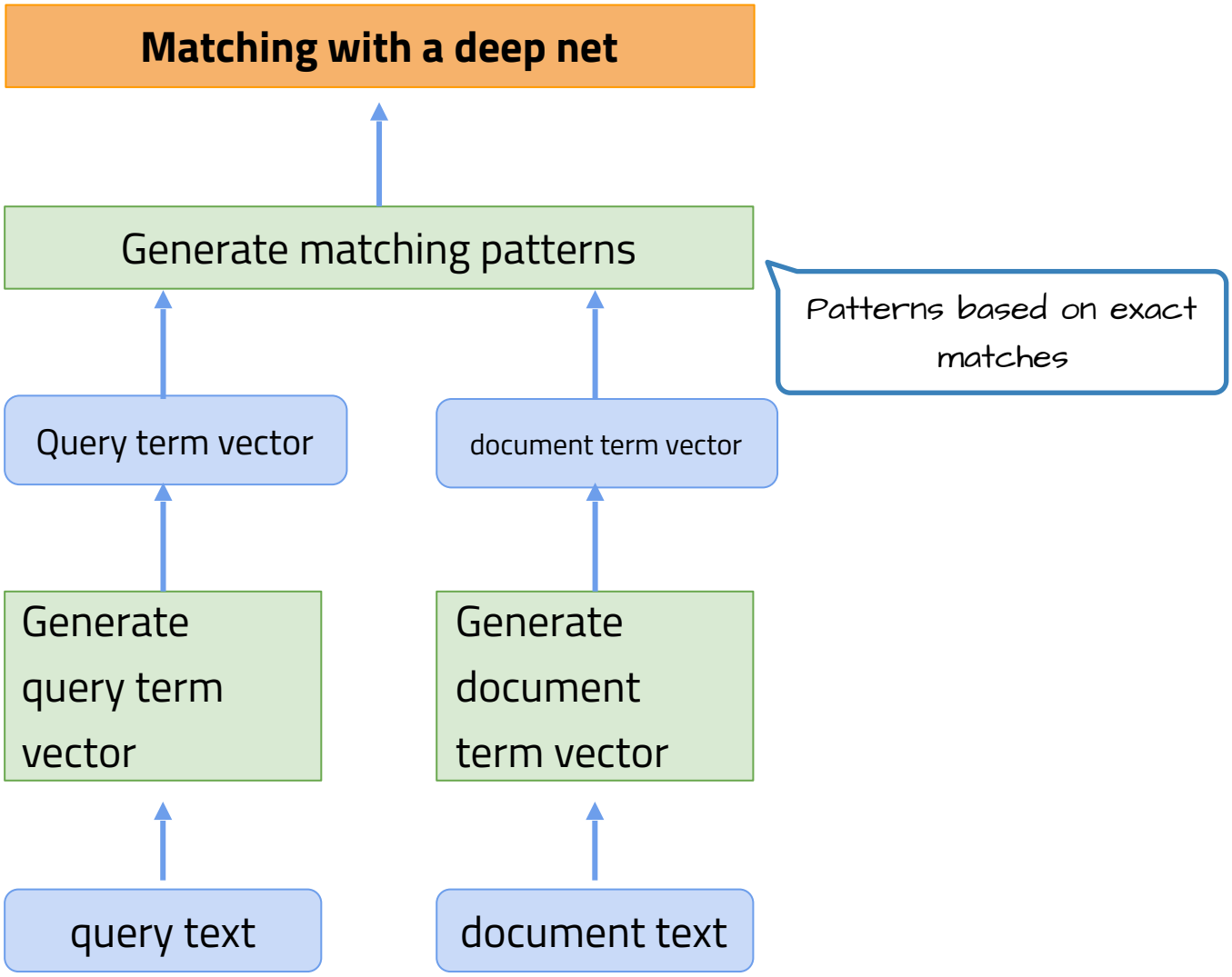
Generally speaking...



Representation focused models



Interaction focused models

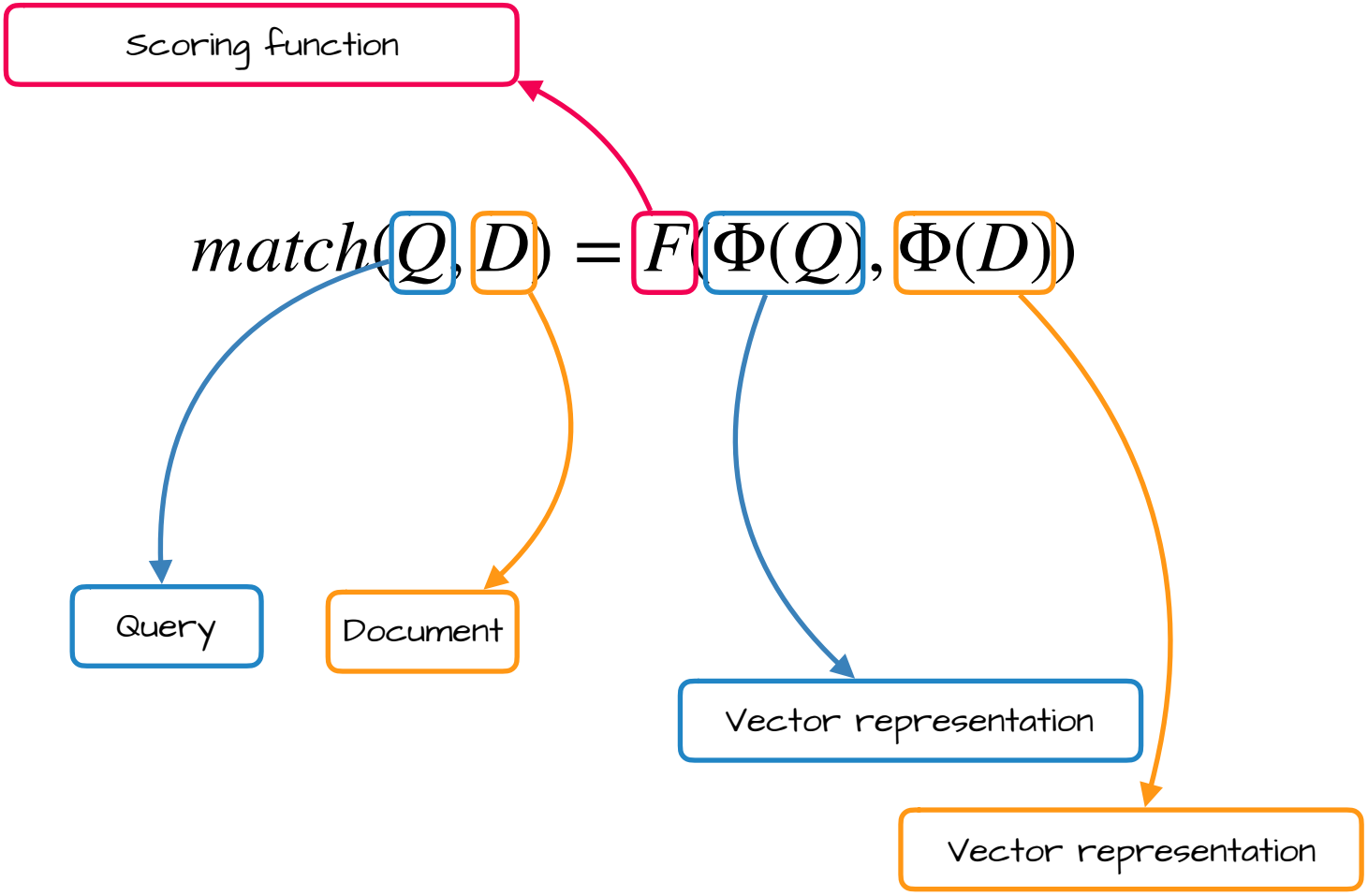


Representation focused models

(Or Siamese Networks)

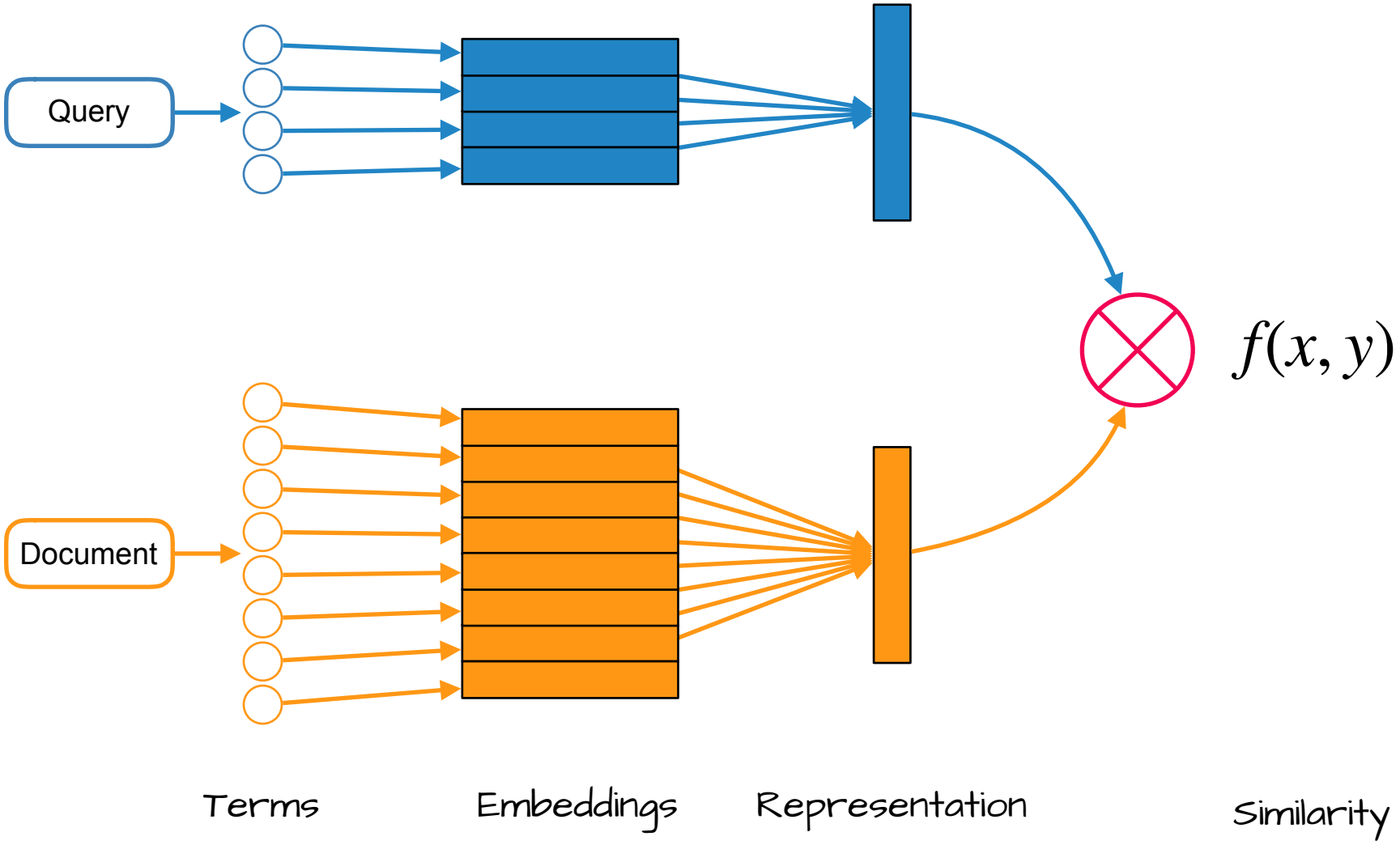
https://arxiv.org/pdf/1707.07700.pdf

Representation focused models



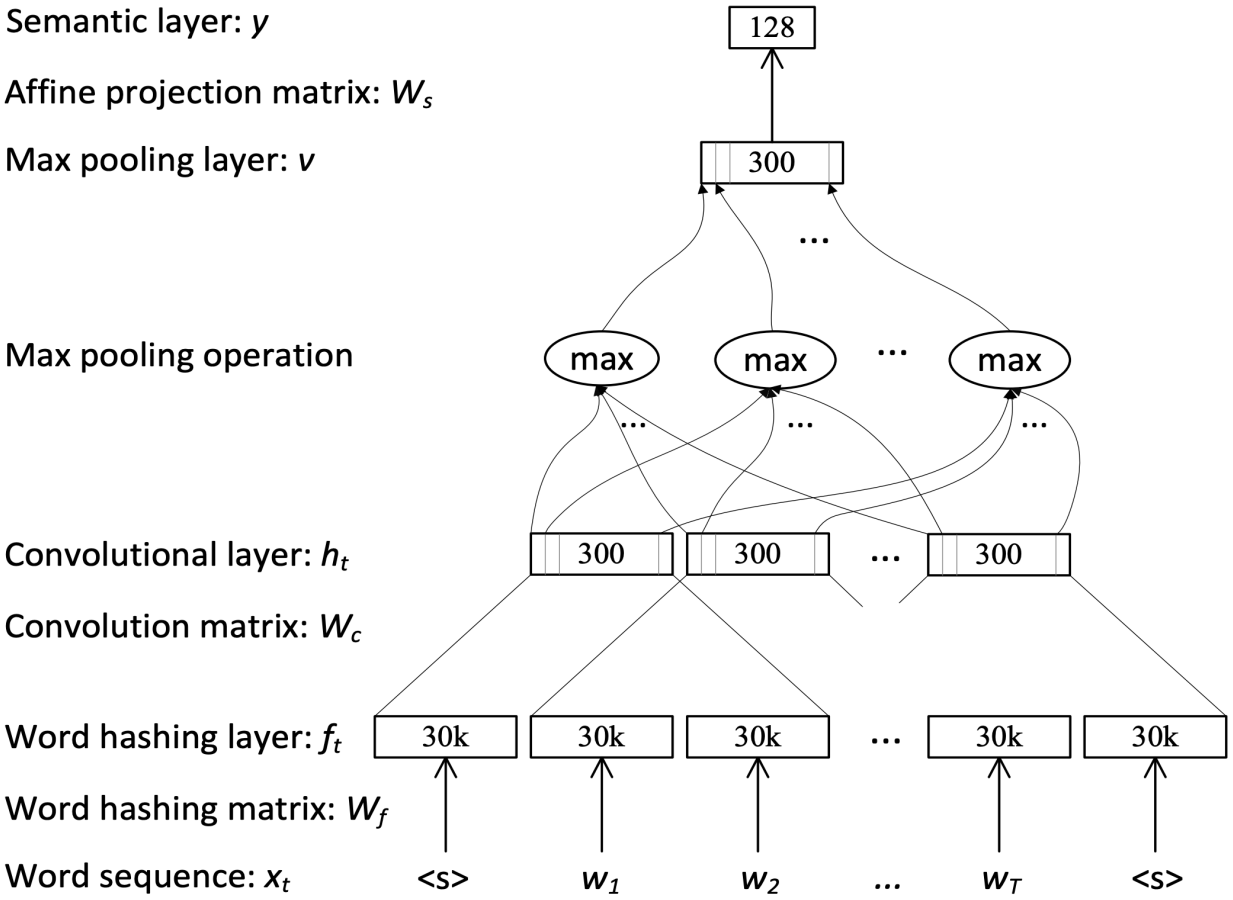
https://arxiv.org/pdf/1707.07700.pdf

Representation focused models



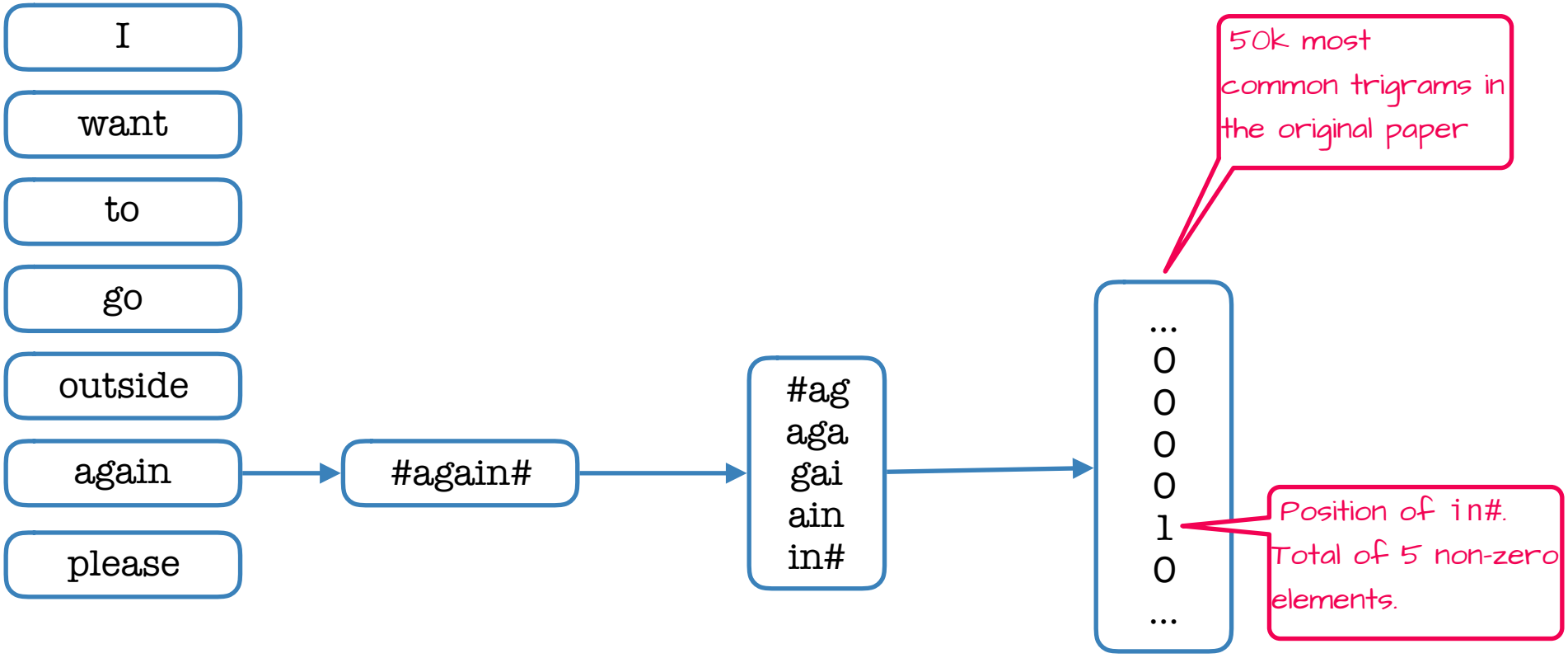
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/www2014_cdssm_p07.pdf

CDSSM - Convolutional Deep Structure Semantic Models



https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/www2014_cdssm_p07.pdf

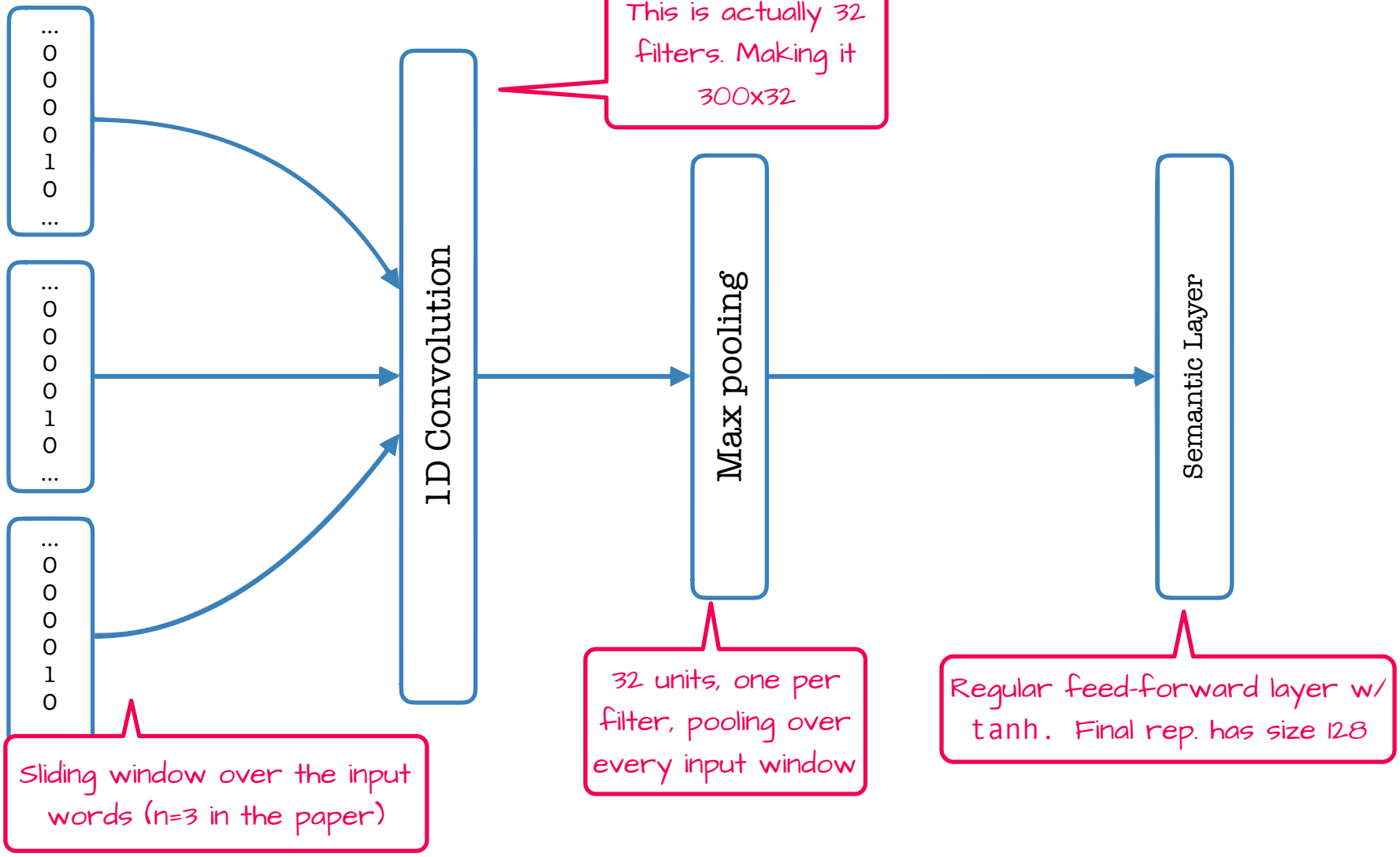
CDSSM - Word Hashing Layer



Input query | End-of-word markers | Trigrams | trigram feature vector

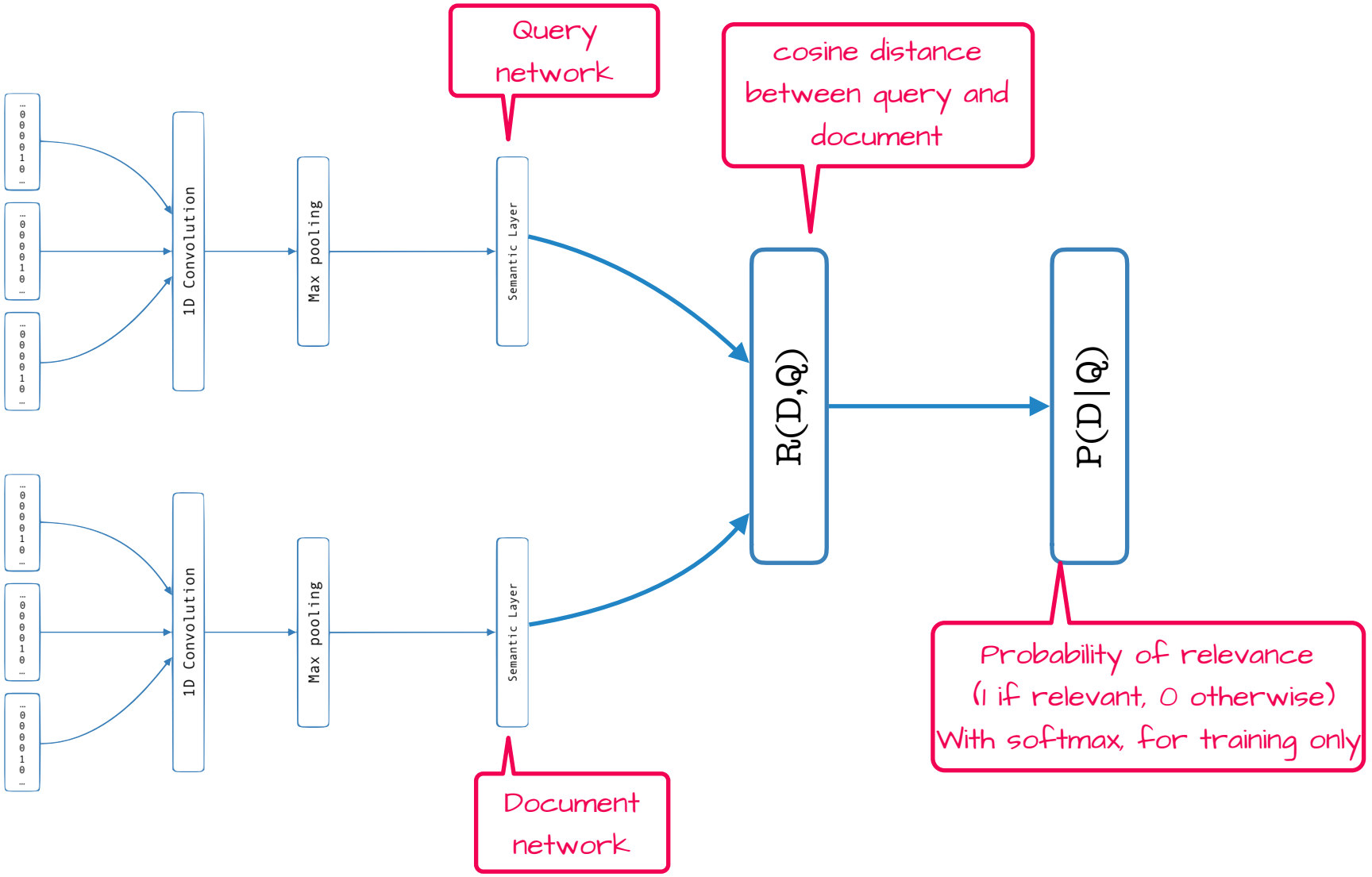
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/www2014_cdssm_p07.pdf

CDSSM - Convolutional Layer



https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/www2014_cdssm_p07.pdf

CDSSM - Full Network



https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/www2014_cdssm_p07.pdf

CDSSM - Results

Using bing queries

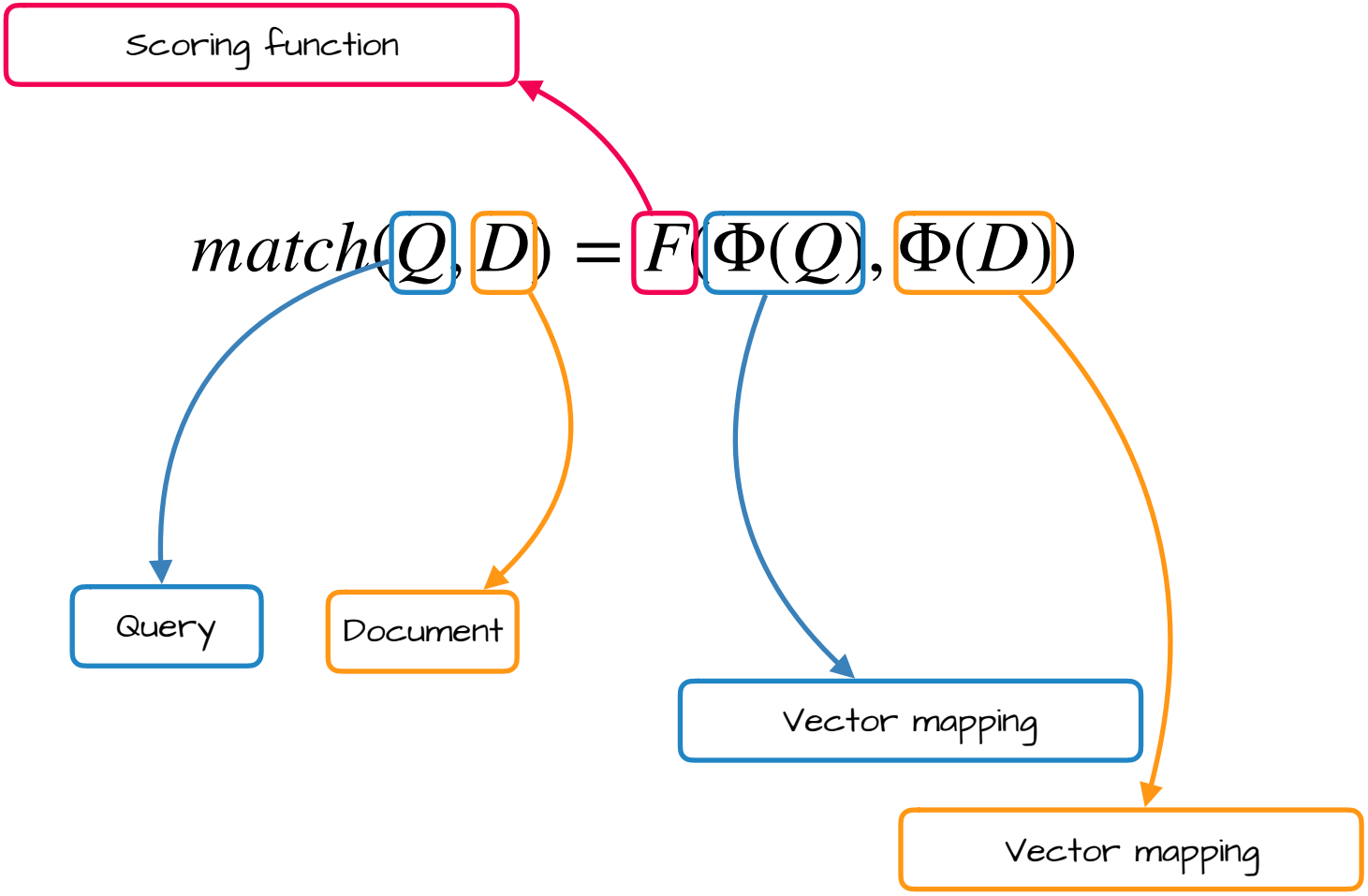
#	Models	NDCG@1	NDCG@3	NDCG@5
1	BM25	0.305	0.328	0.388
2	ULM	0.304	0.327	0.385
3	WTM	0.315	0.342	0.411
4	PTM (len <=3)	0.319	0.347	0.413
5	DSSM	0.320	0.355	0.431
6	**C-DSSM (win=3)**	**0.342**	**0.374**	**0.447**

https://dl.acm.org/citation.cfm?id=2767830

Interaction focused models

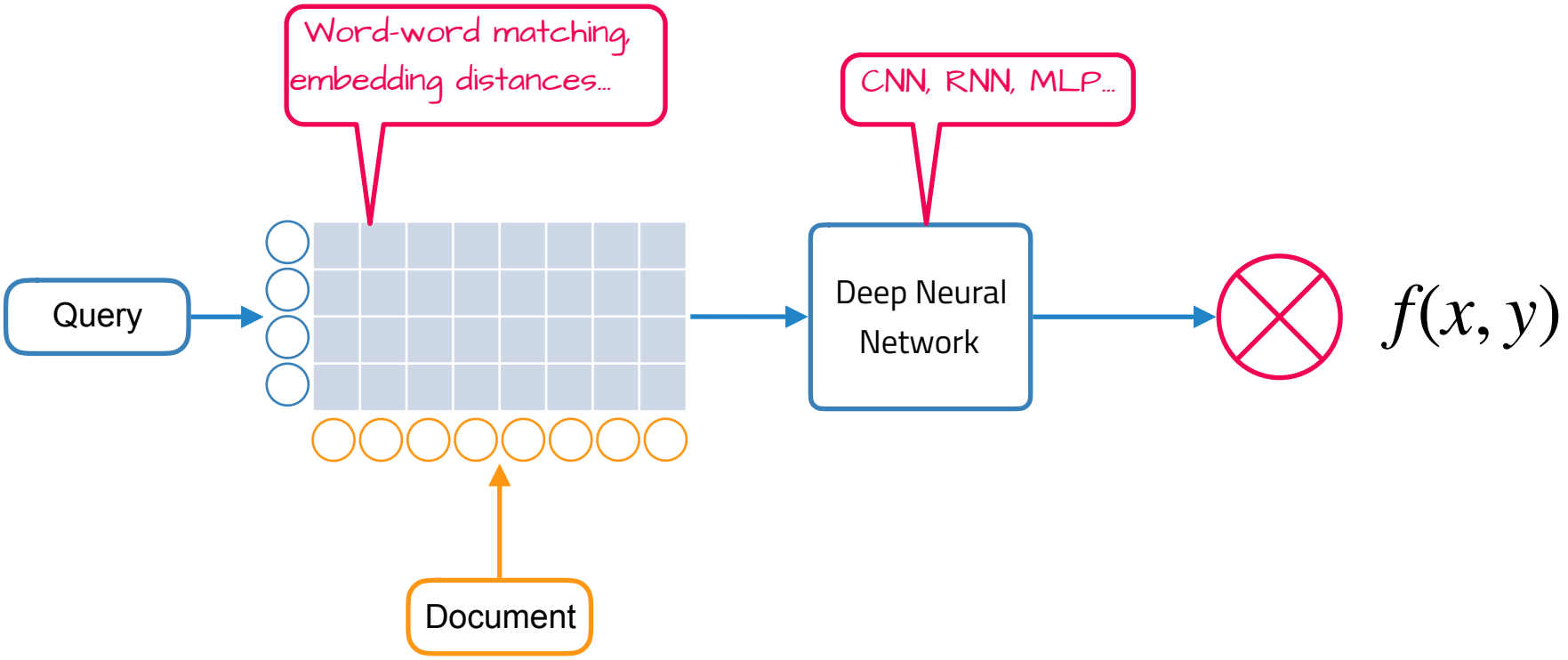
https://arxiv.org/pdf/1707.07700.pdf

Interaction-focused models



https://arxiv.org/pdf/1707.07700.pdf

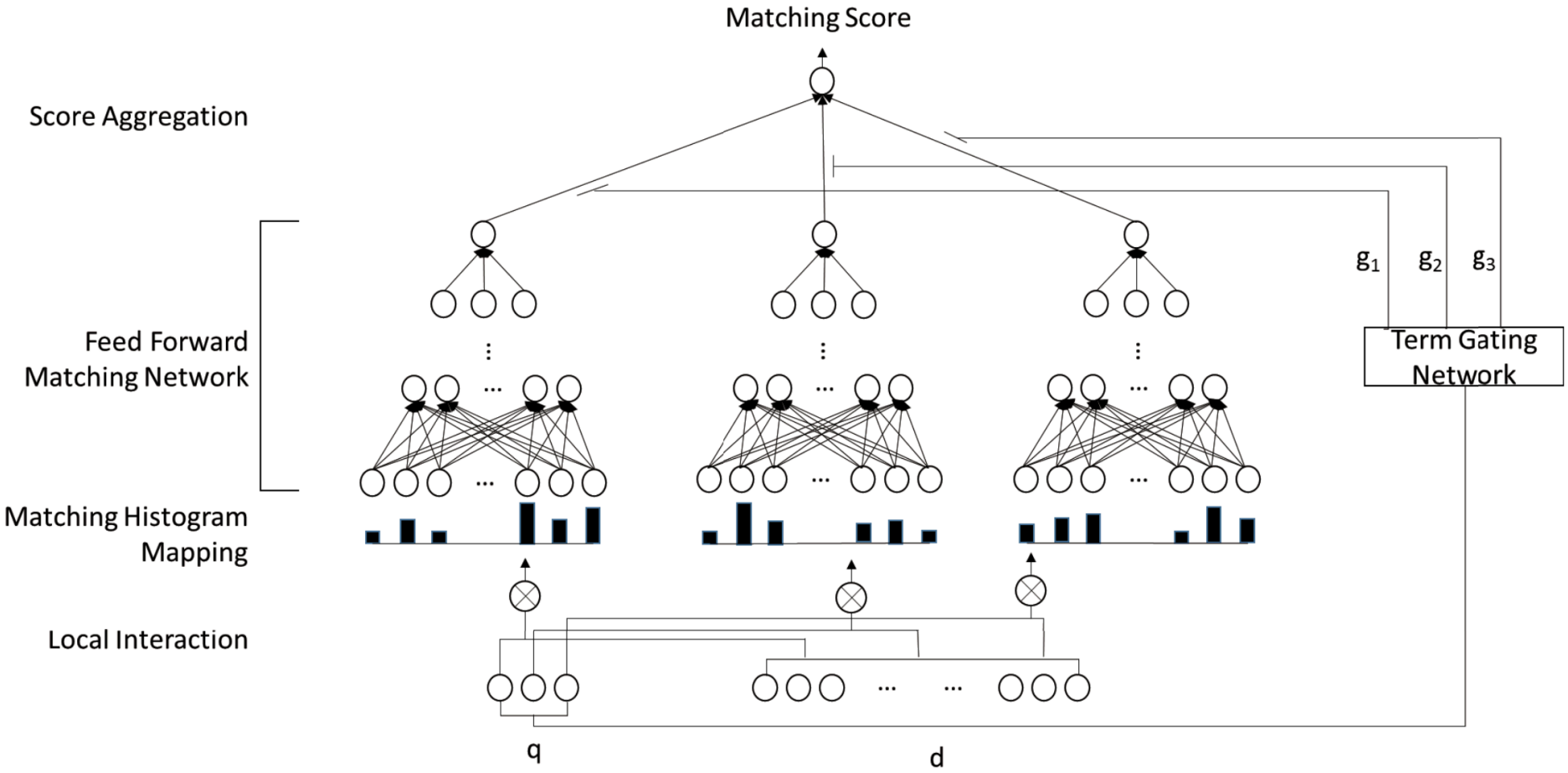
Interaction-focused models



Interaction matrix (M) Deep Neural Net (G) Relevance Score (H)

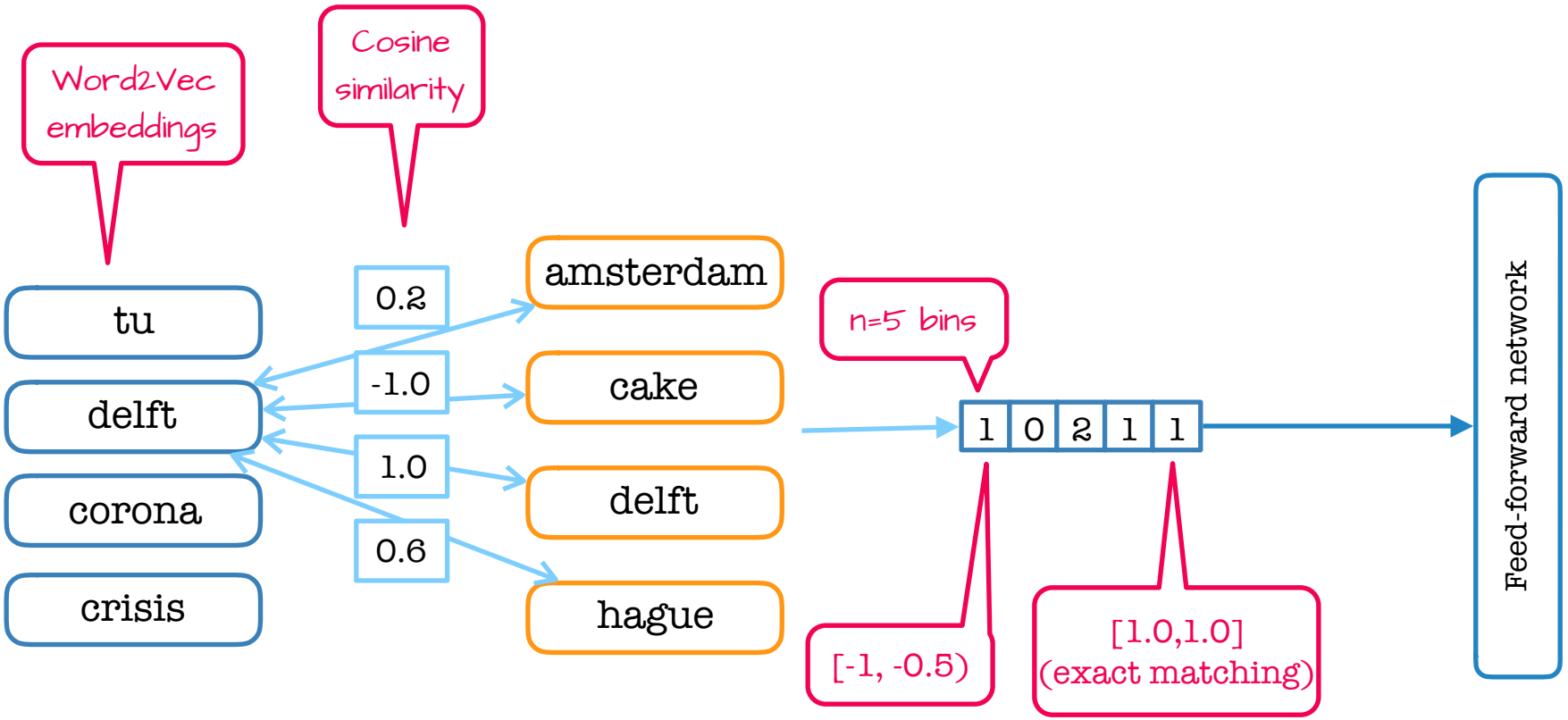
https://arxiv.org/pdf/1711.08611.pdf

DRMM - Deep Relevance Matching Model



https://arxiv.org/pdf/1711.08611.pdf

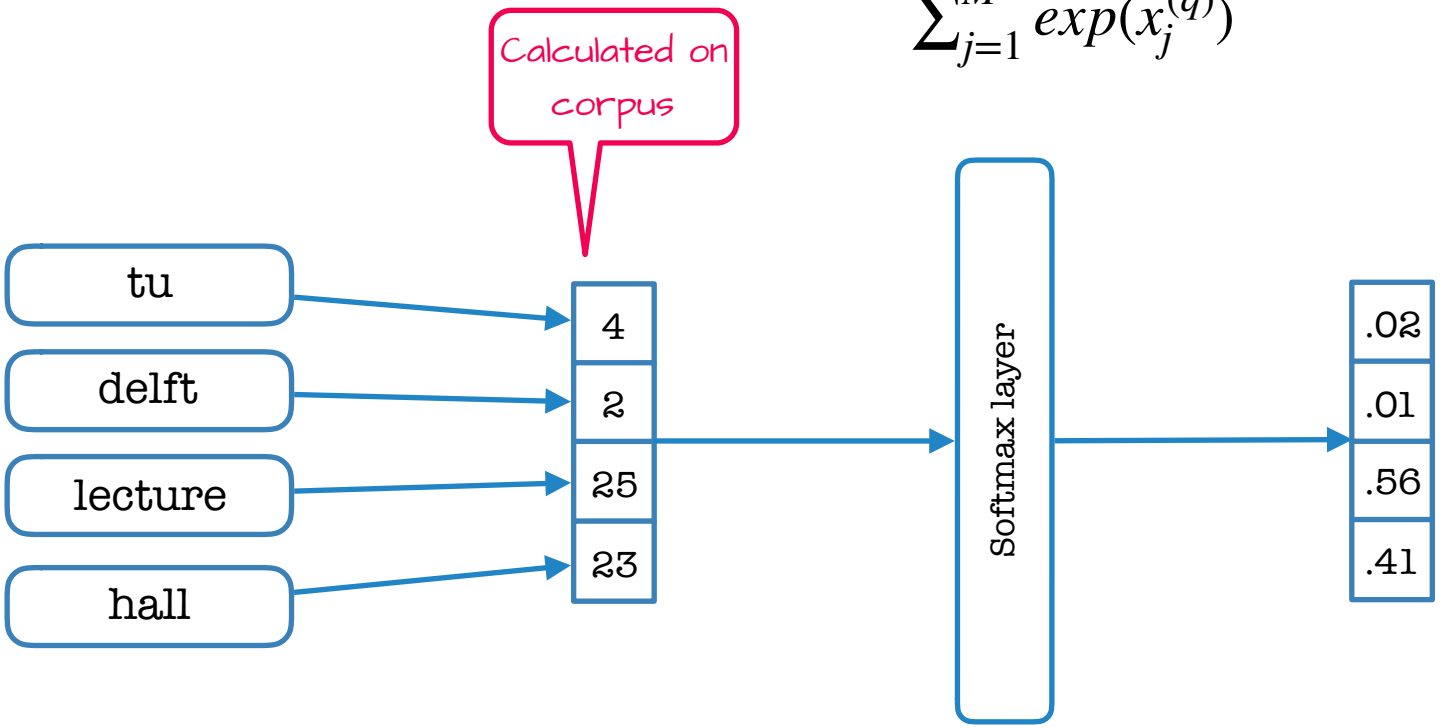
DRMM - Local Interactions and Histogram Mapping



https://arxiv.org/pdf/1711.08611.pdf

DRMM - Term Gating Network

$$g_i = \frac{exp(x_i^{(q)})}{\sum_{j=1}^{M} exp(x_j^{(q)})}$$

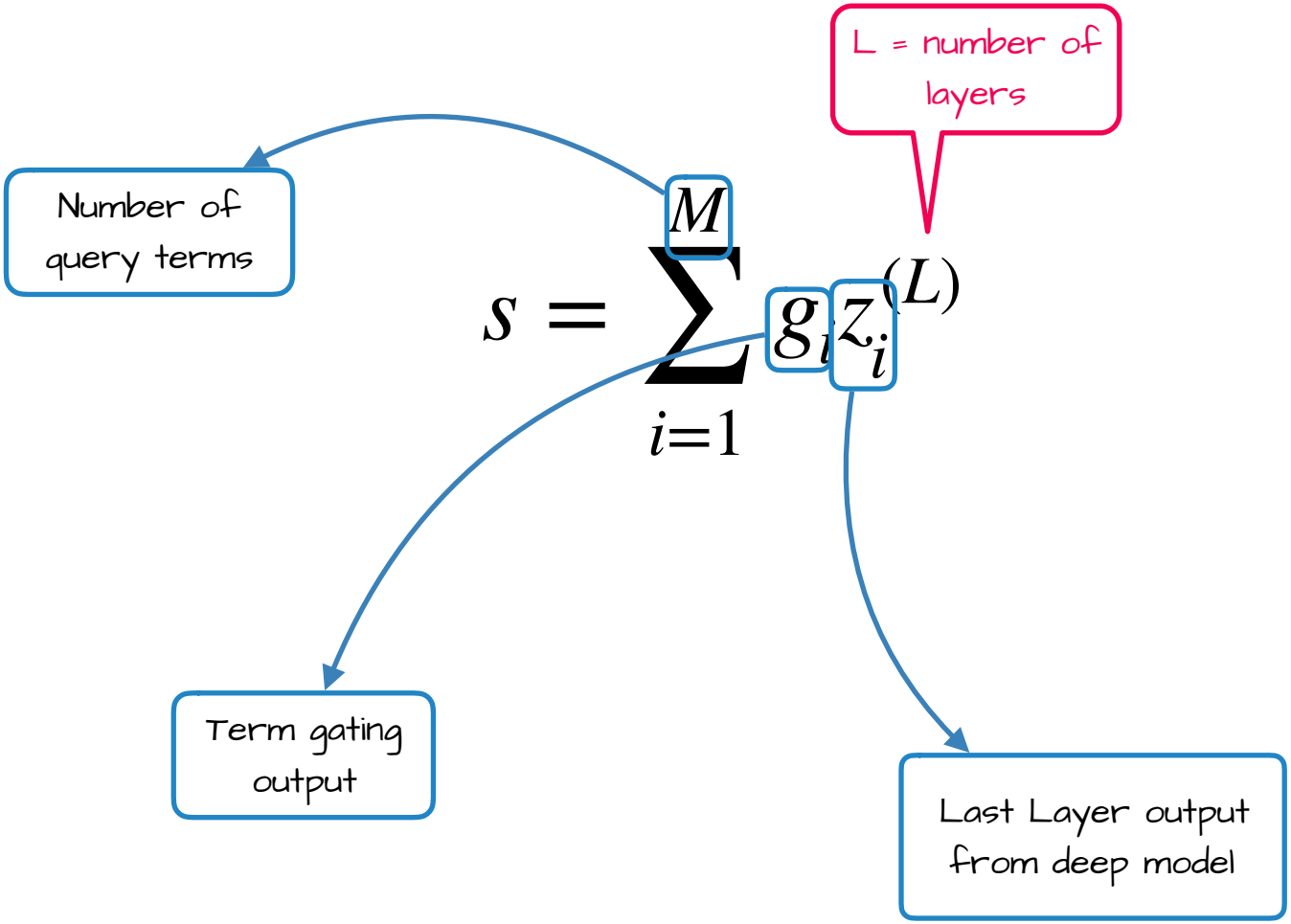


Input query

IDF Vector

https://arxiv.org/pdf/1711.08611.pdf

DRMM - Final aggregation



https://arxiv.org/pdf/1711.08611.pdf

DRMM - Results

Robust04, titles only

#	Models	MAP	NDCG@20	P@20
1	QL	0.253	0.415	0.369
2	BM25	0.255	0.418	0.370
3	CDSSM	0.067	0.146	0.125
4	ARC-II	0.067	0.147	0.128
5	**DRRM**	**0.279**	**0.431**	**0.382**

Caveats?

https://cs.uwaterloo.ca/~jimmylin/publications/Lin_SIGIRForum2018.pdf

A properly tuned BM25+RM3…

Condition	AP	P20
BM25	0.238	0.354
BM25 + Features	0.250	0.367
Neural_x	0.258	0.372
Neural_y	0.256	0.370
$\text{Neural}_x + \text{Neural}_y$		0.373
$\text{A} + \text{Neural}_y$		0.380
$\text{A} + \text{Neural}_y + \text{M}$		0.380
$\text{B} + \text{Neural}_y$	0.270	0.383
$\text{B} + \text{Neural}_y + \text{M}$	0.272	0.386
Anserini: QL	0.2481	0.3517
Anserini: BM25	0.2528	0.3598
Anserini: BM25 + RM3 (independent)	0.2991	0.3901
Anserini: BM25 + RM3 (joint)	0.2956	0.3931

This is basically DRMM…

http://rali.iro.umontreal.ca/rali/sites/default/files/publis/p59-nie%5B1%5D.pdf

Insights

- Generally, interaction-focused techniques work best (Nie, et al., ICTR ’18).
- For short texts, vocabulary mismatch is more serious than for long texts
- MatchZoo has lots of models implemented in Python/TensorFlow (**BEWARE! Implementations may be different from original paper**!) [https://github.com/NTMC-Community/MatchZoo/].
- Properly turned “classical" IR models are still good (if not better than neural) at most IR tasks.

What now?

[…] It is unclear to me, at least for "classic" ad hoc retrieval problems without vast quantities of training data from behavior logs, whether neural techniques are actually more effective in absolute terms.

Jimmy Lin, SIGIR Forum, December 2018

Things have changed FAST in one year…

> *I believe there is now clarity: deep **transformer** models, heavily **pretrained** via **language modeling** tasks, have significantly and substantially improved the effectiveness of document retrieval, even in the **absence of vast amounts of training data**.*

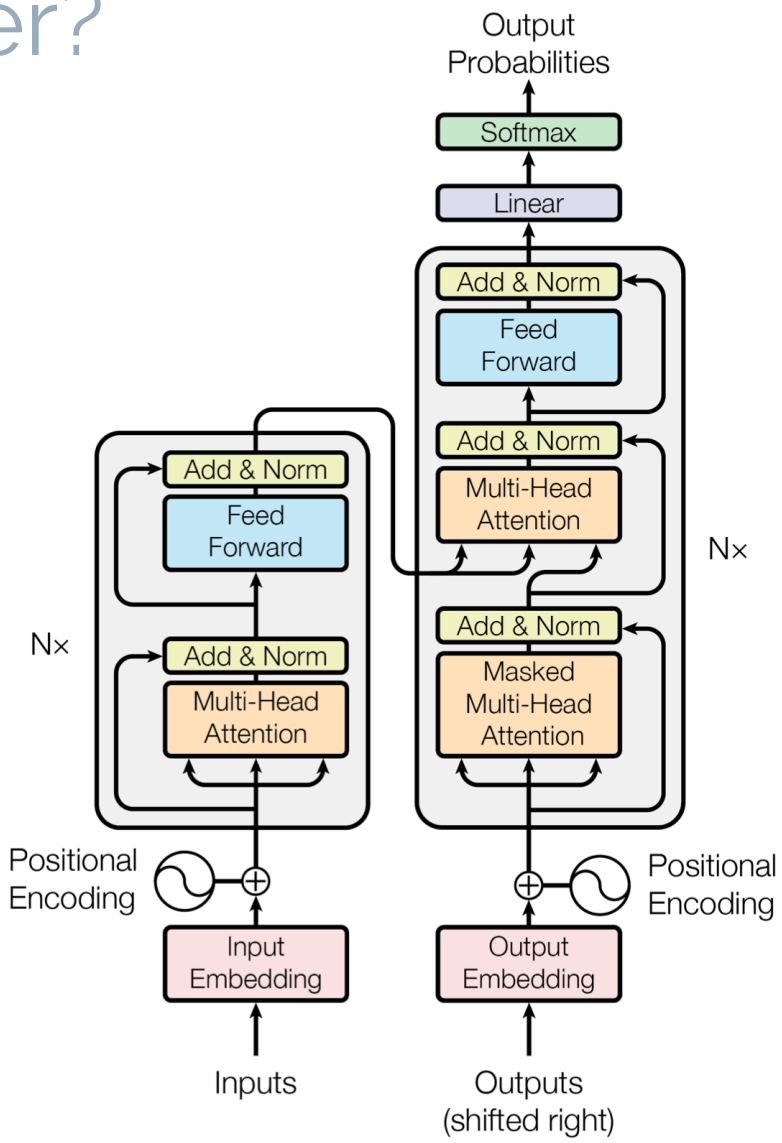
Jimmy Lin, SIGIR Forum, December 2019

What is a Transformer?

https://arxiv.org/abs/1706.03762

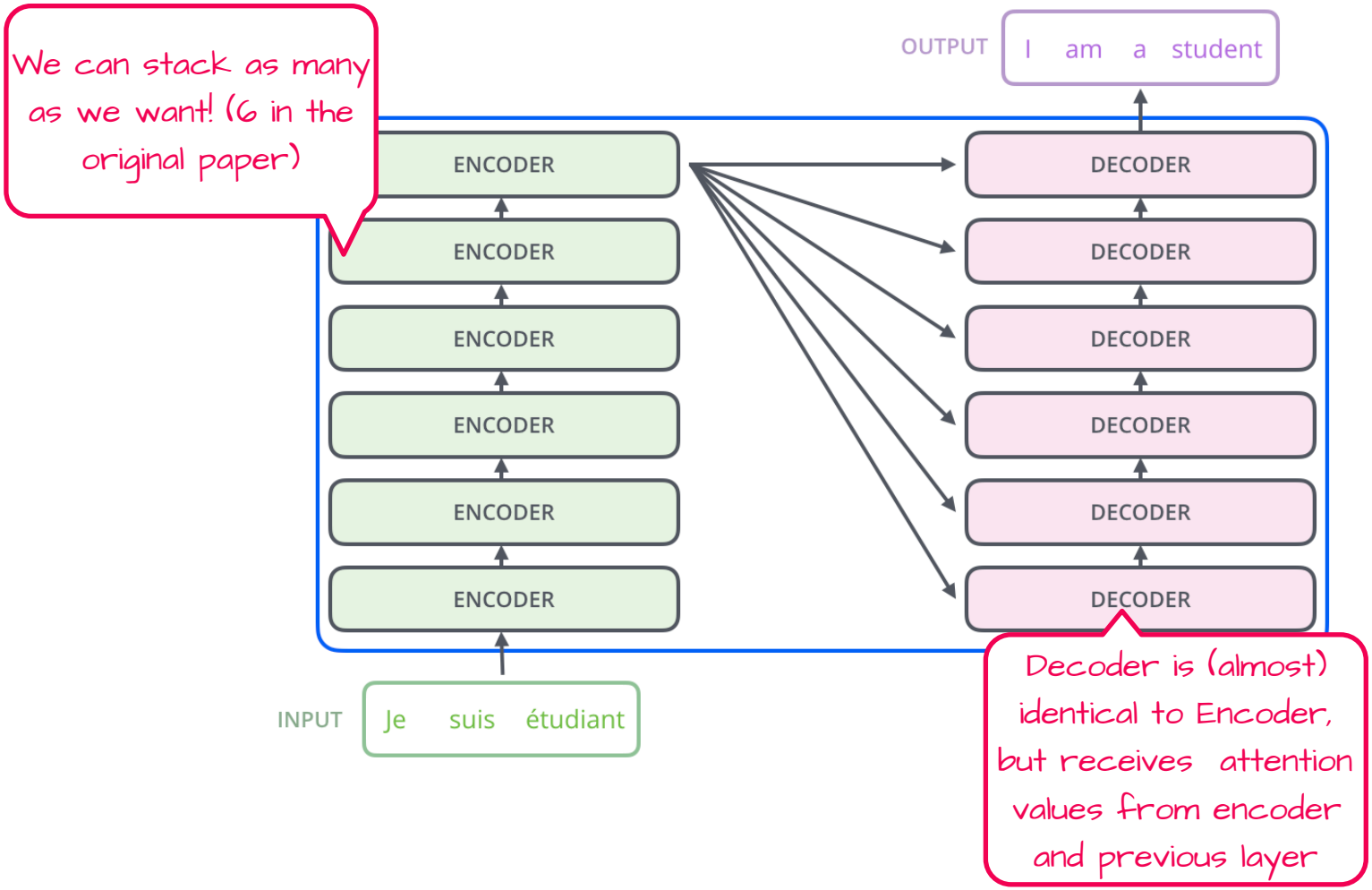
What is a Transformer?

- A novel neural architecture, built from scratch for text.
- No use of recurrence or convolutions.
- Based entirely on self-attention.
- Highly "parallelizable".
- Easier and faster to train.
- We will cover the main ideas here. For the commented paper with code, check here: http://nlp.seas.harvard.edu/2018/04/03/attention.html
- Images are from here: http://jalammar.github.io/illustrated-transformer/ (check it out! It's really good!)



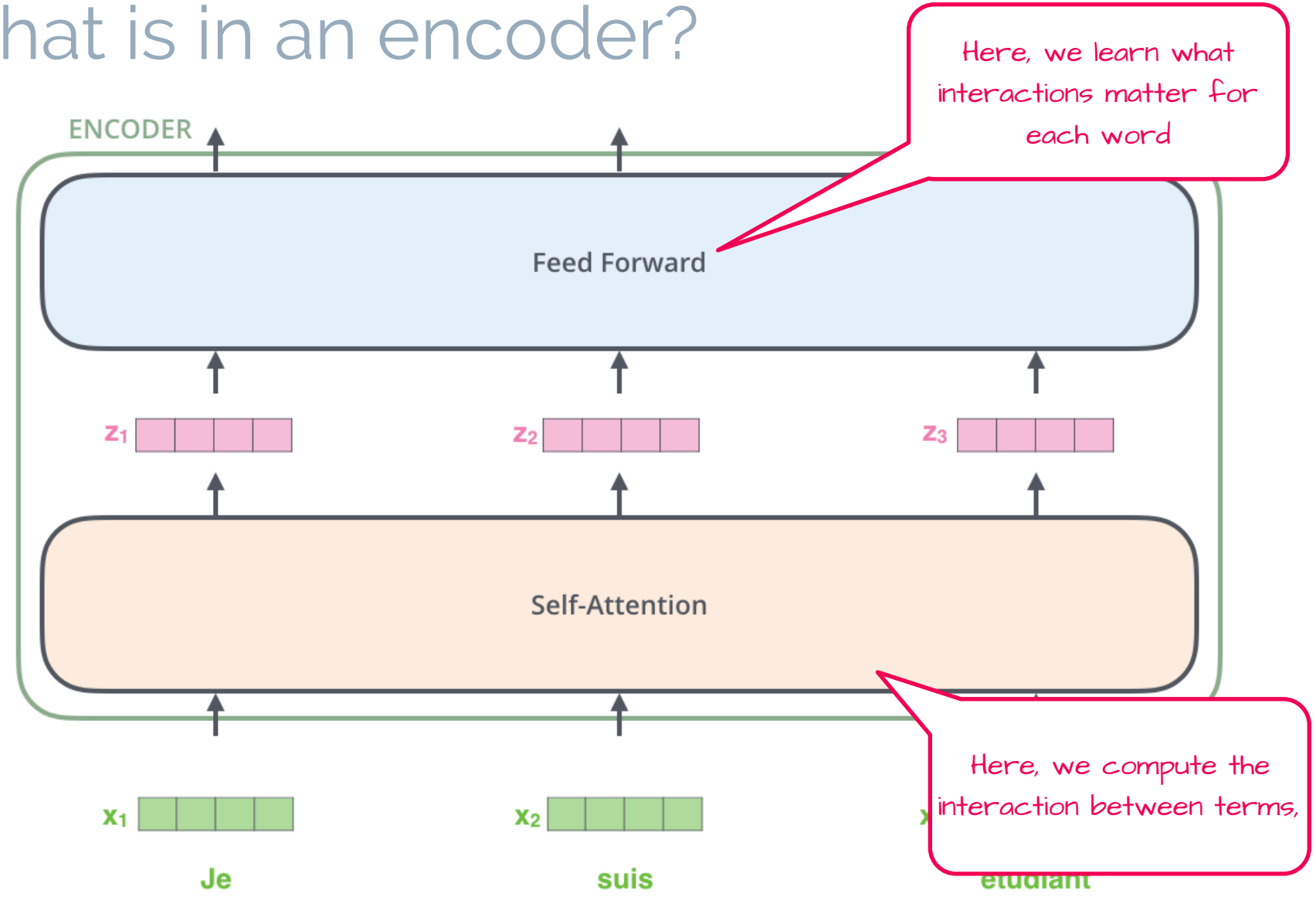
http://jalammar.github.io/illustrated-transformer/

What is a transformer?



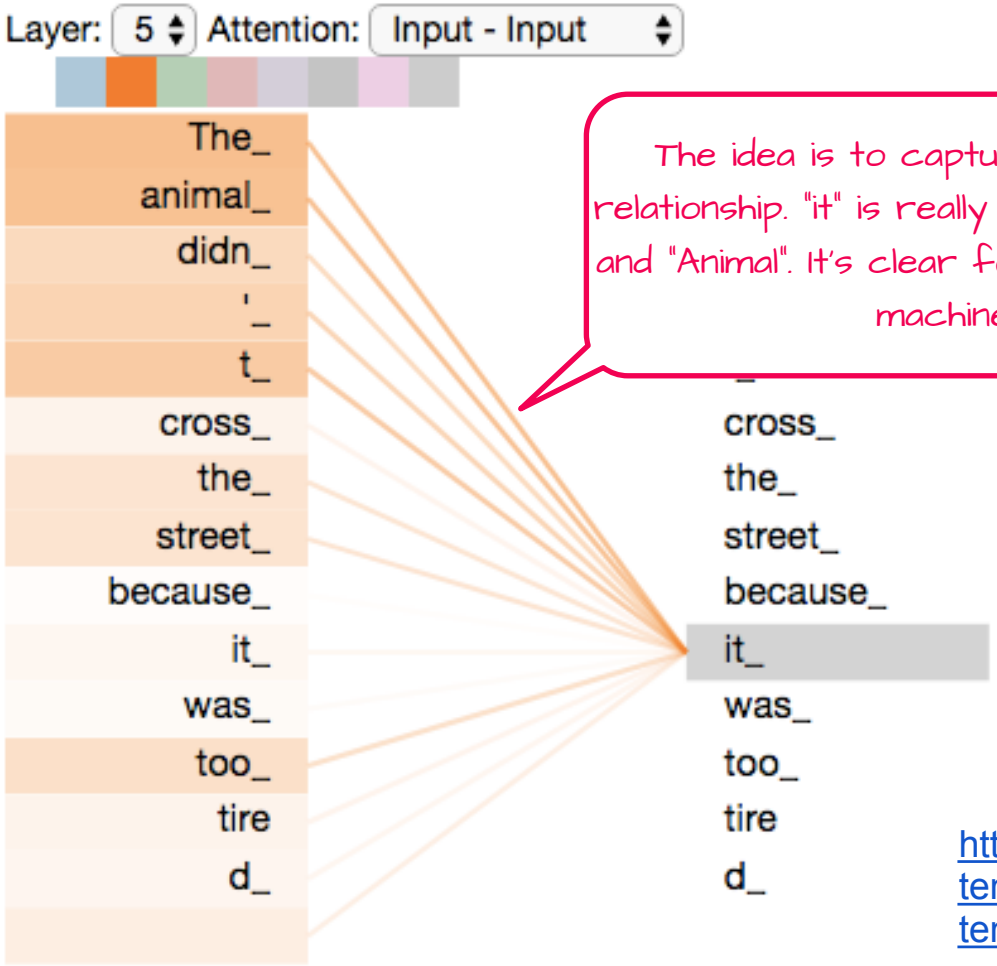
http://jalammar.github.io/illustrated-transformer/

What is in an encoder?



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

What is Self-Attention?



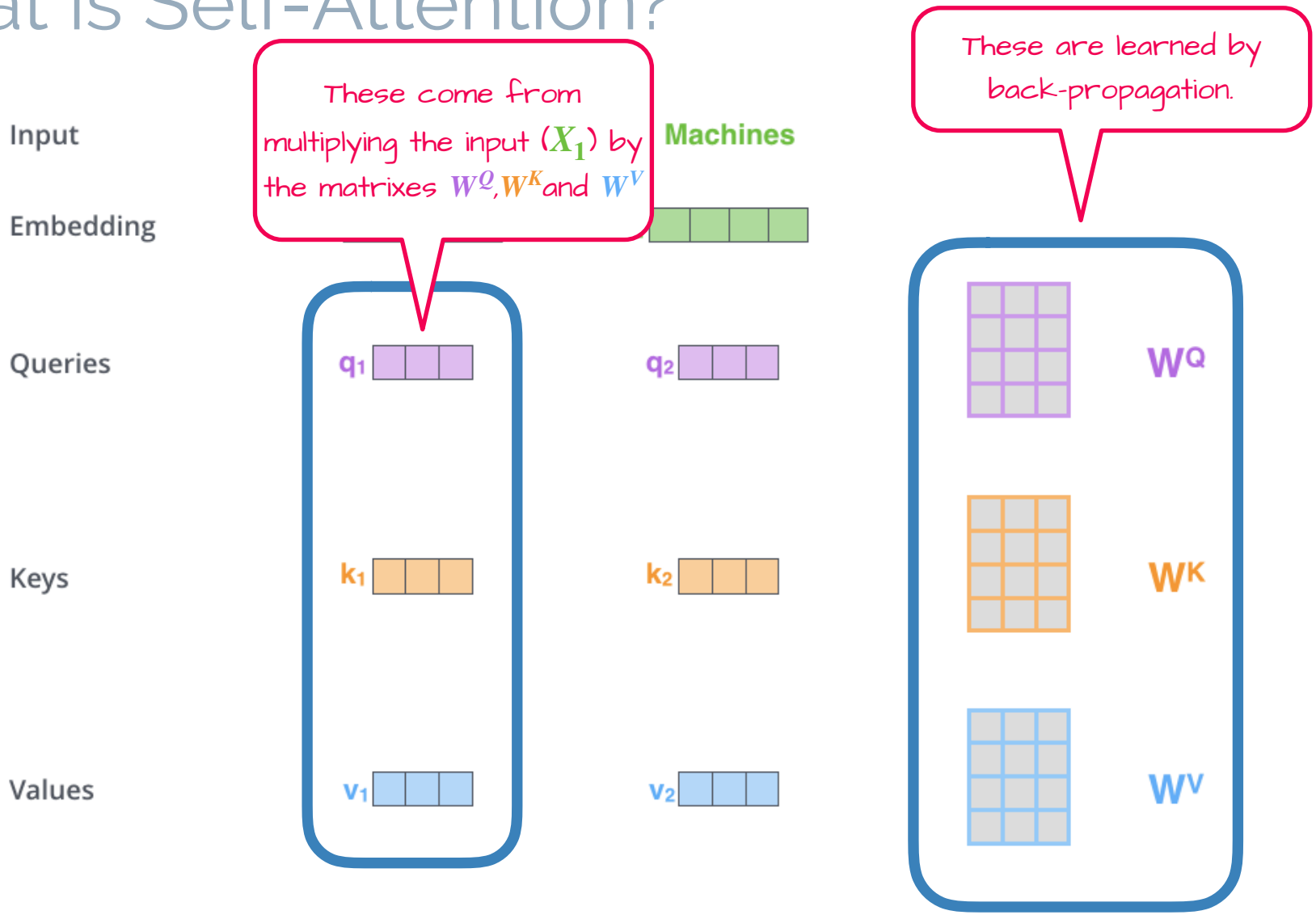
The idea is to capture this kind of relationship. "it" is really important to "The" and "Animal". It's clear for us, but not for machines.

Play with this idea here:

https://colab.research.google.com/github/tensorflow/tensor2tensor/blob/master/tensor2tensor/notebooks/hello_t2t.ipynb

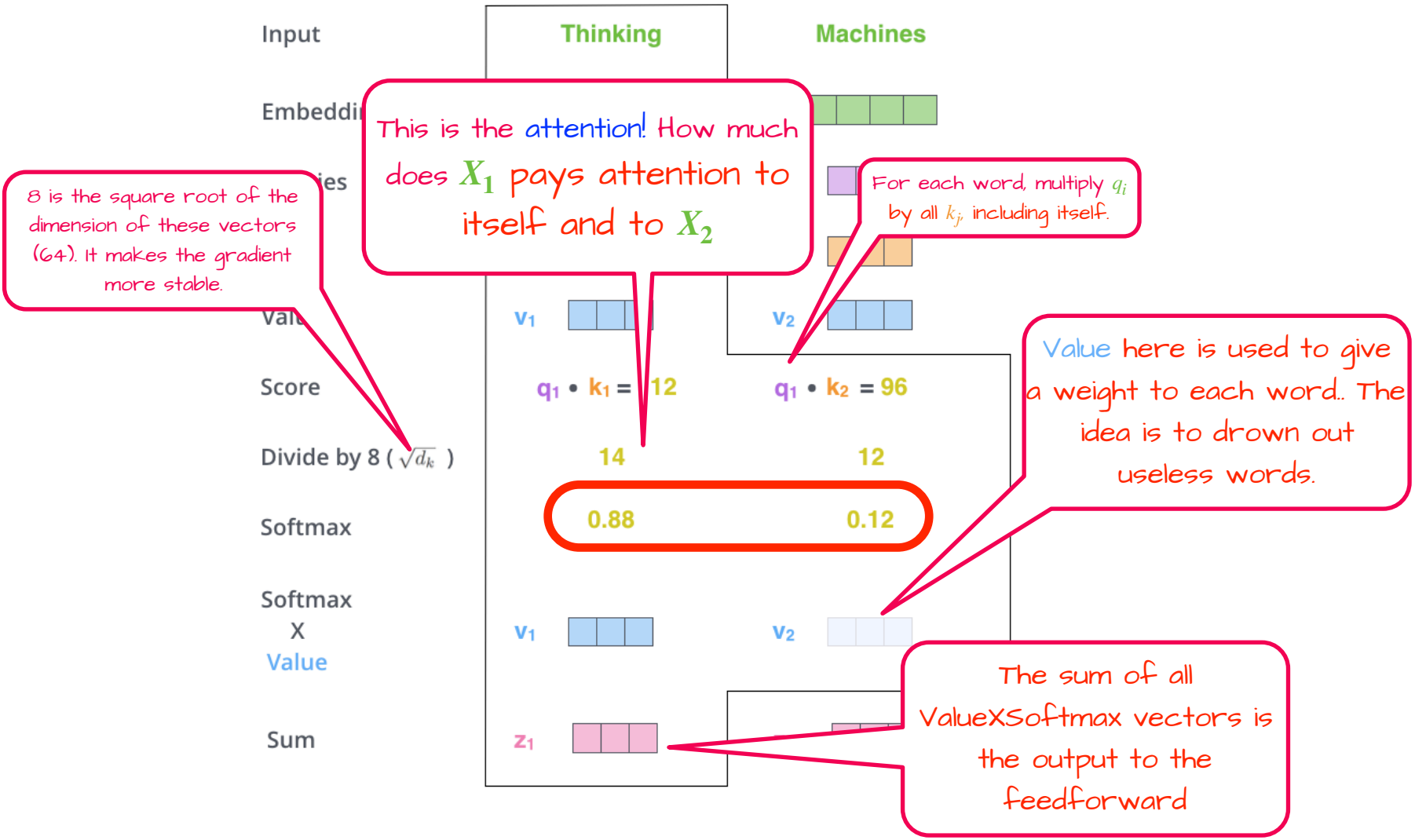
http://jalammar.github.io/illustrated-transformer/

What is Self-Attention?



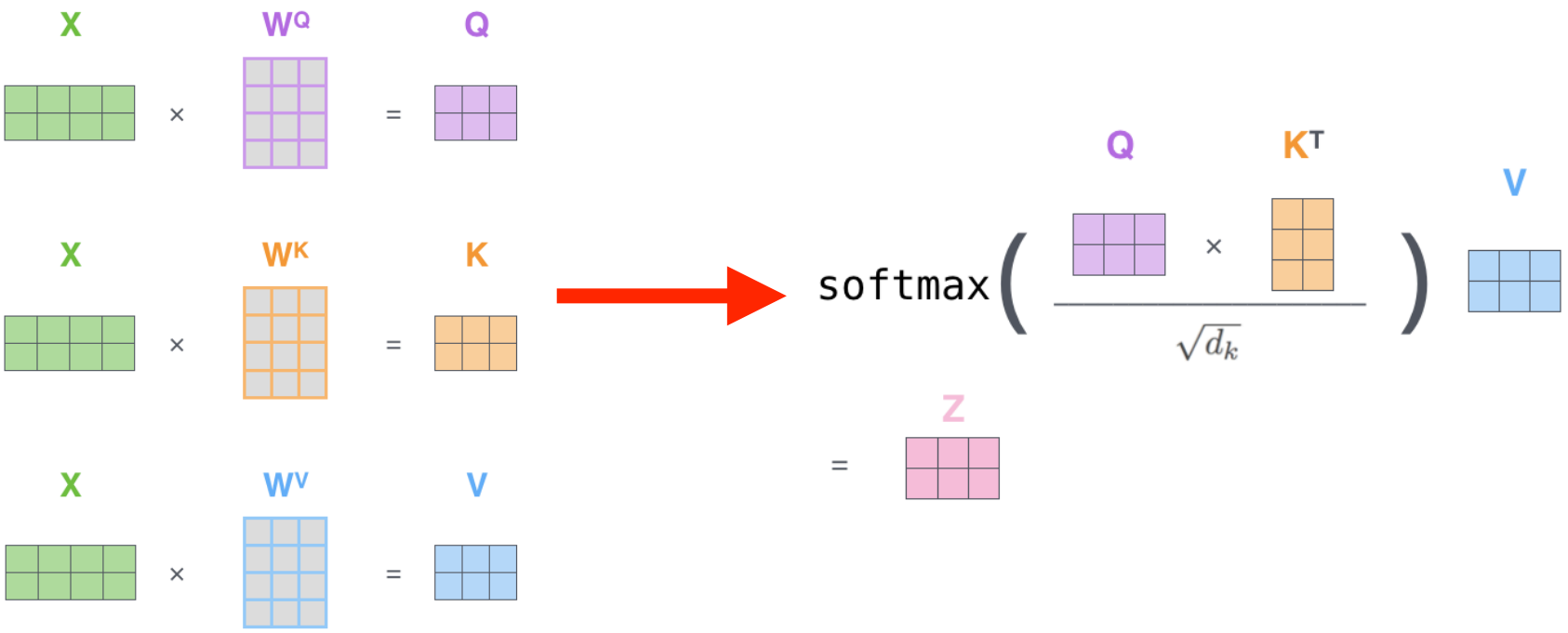
http://jalammar.github.io/illustrated-transformer/

What is Self-Attention?



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

What is Self-Attention?



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

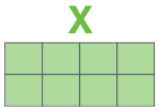
What is Se

Multiple heads means we do the same thing multiple times. It allows different "heads" to attend to different types of relationships

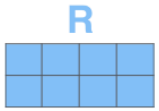
1) This is our input sentence*

2) We embed each word*

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



These are smaller representations (64d)

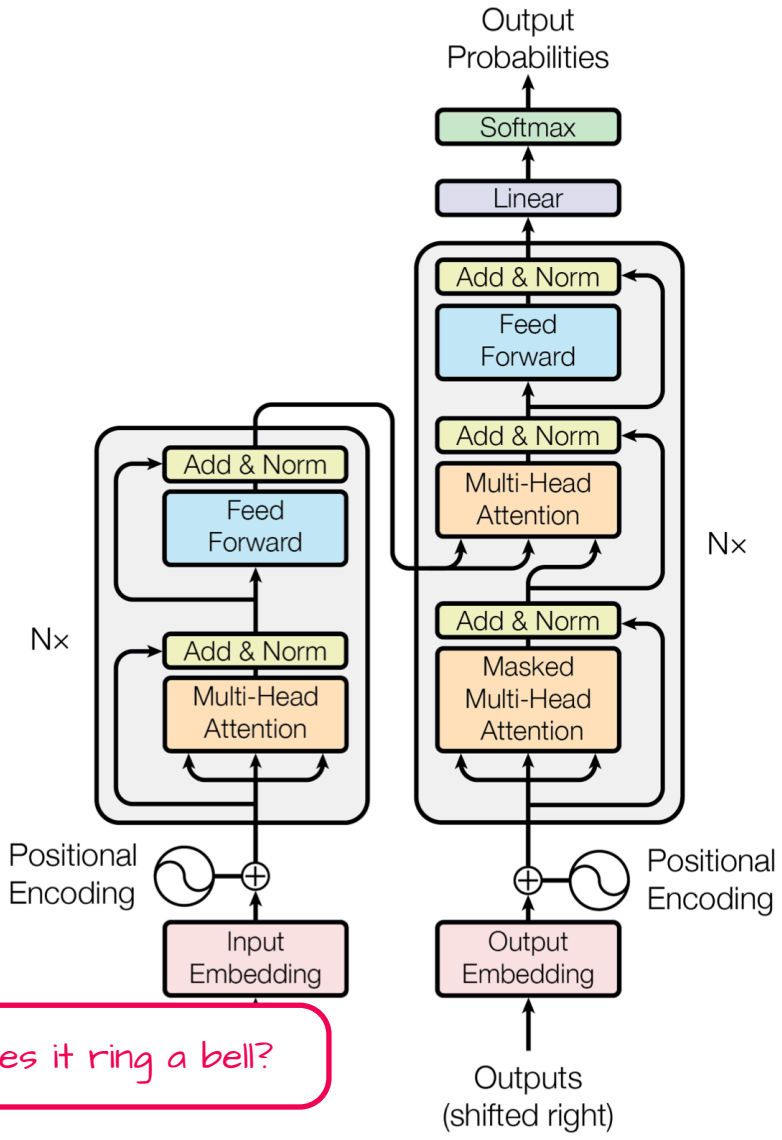
Project back into a smaller size, for an output in the

For all layers but the first, we use the previous one output

https://arxiv.org/abs/1706.03762

What is a transformer?

- There is more to it!
- How to represent the position of the word?
- How does the feed-forward look like?
- What is What else is in the decoder?
- We don't have the time to cover it all here.
- The main take-aways from transformers are:
  - It's a stack of self-attention layers
  - The output of the encoder is one representation for each word.



Does it ring a bell?

https://arxiv.org/abs/1810.04805

What is BERT?

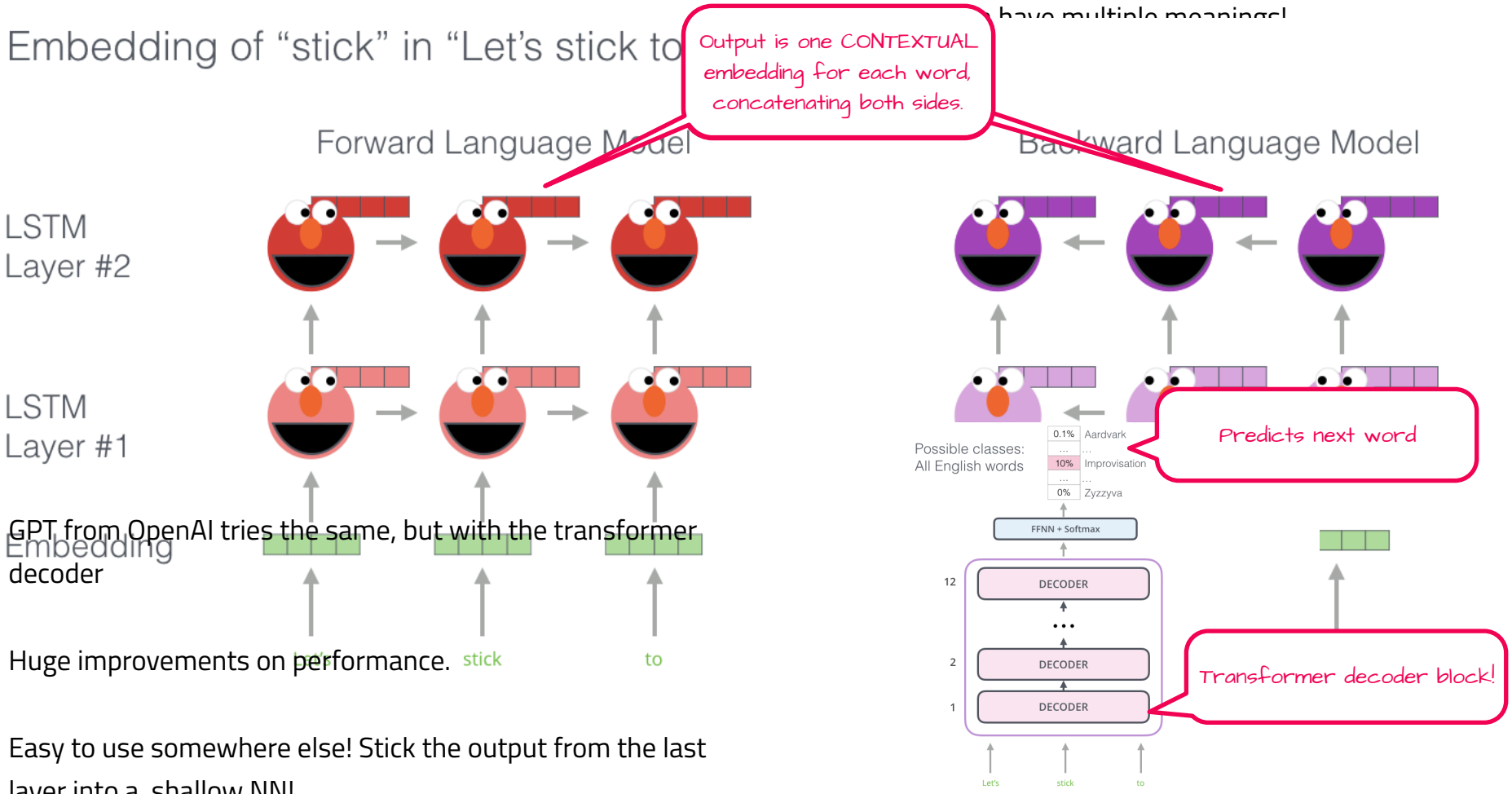
- Bidirectional Encoder Representations from Transformers
- Comes from a lineage of language models, like ELMo and GPT
- Generate one **CONTEXTUAL** embedding for each word
- It's basically a stack of Encoders from Transformers
- Two key things that made it popular:
  - Weakly Supervised Learning
  - Transfer Learning
- For more details, follow this link: http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

With ELMo and Transformers, probably the most important NLP work in recent years.
Some people call it the "ImageNet moment for NLP"

§

https://arxiv.org/abs/1810.04805

BERT - Overview (ELMO and GPT)



- GPT from OpenAI tries the same, but with the transformer decoder
- Huge improvements on performance.
- Easy to use somewhere else! Stick the output from the last layer into a shallow NN!

https://arxiv.org/abs/1810.04805

BERT - Overview and pre-train

A shallow Neural Network gets this embedding and predicts what it is

In here, a stack of Transformers encoders. (12/24), with more heads (12/16) and larger embeddings (768/1024 dimensions)

Output is the same size as input! (fixed at 512. Pad with 0s as needed

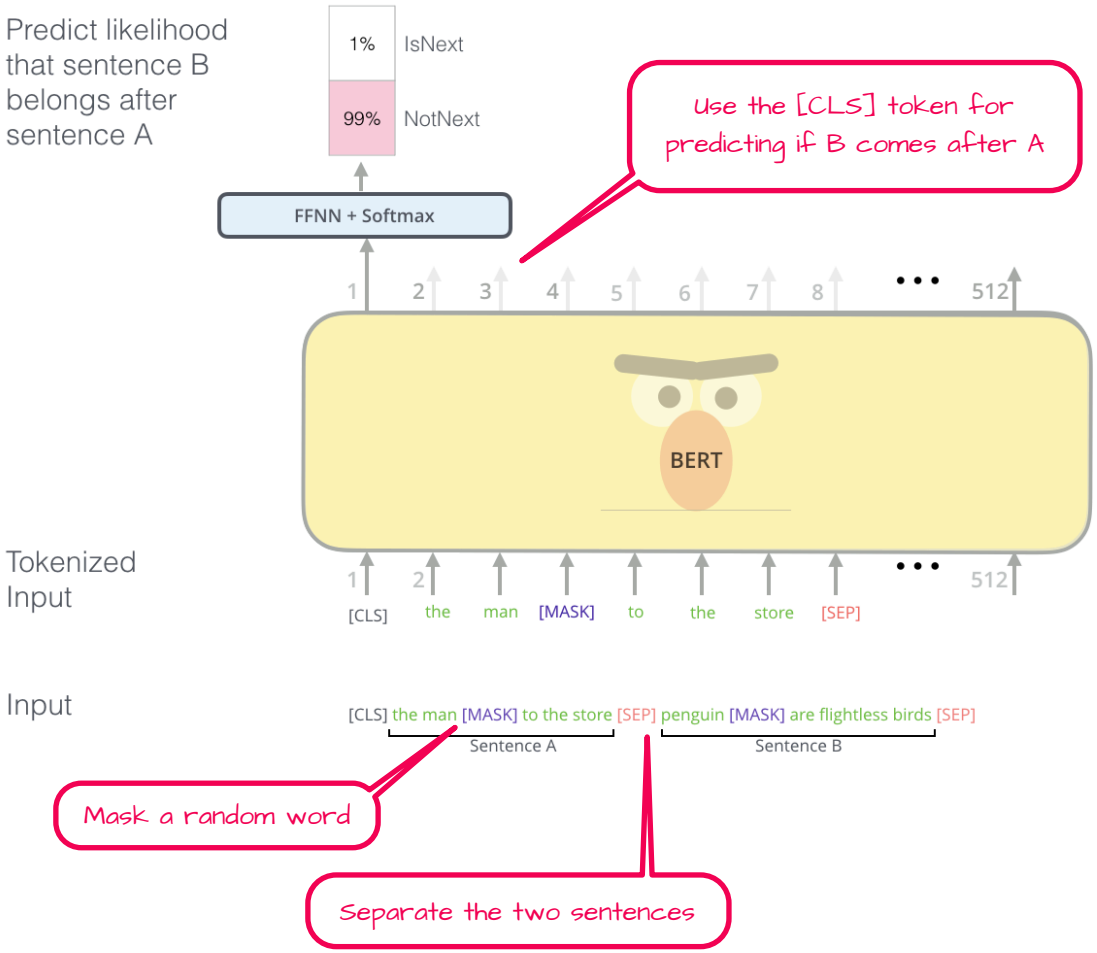
The goal is to predict this word/token. When pre-training, BERT randomly masks 15% of all tokens

Add [CLS] token to create a feature vector for this sentence.

https://arxiv.org/abs/1810.04805

BERT - Weakly supervised

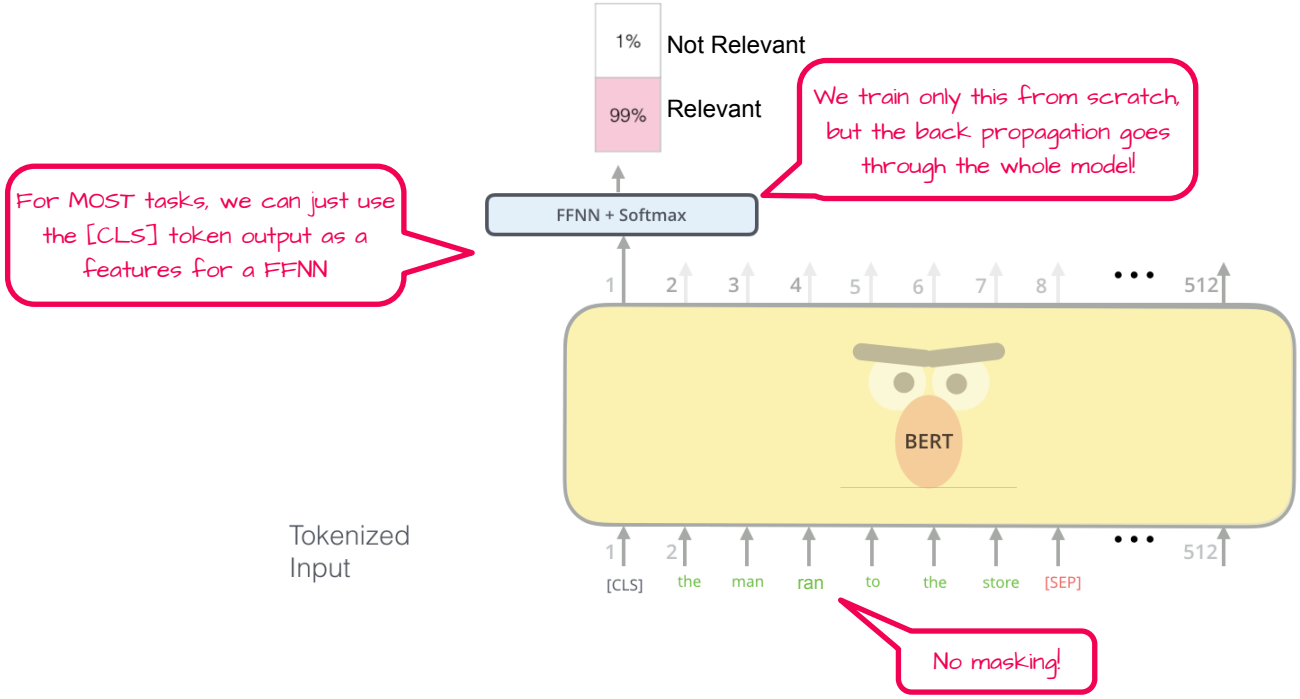
- What does it mean to be "Weakly supervised"?
  - It means that the model can create its own training data.
- For BERT, that means two things:
  - Randomly mask words from the input
  - Predict if two random sentences come right after each other.
- This makes it SUPER effective in using large amounts of data.
- And makes it "task agnostic"



https://arxiv.org/abs/1810.04805

BERT - Transfer Learning (AKA Fine tuning)

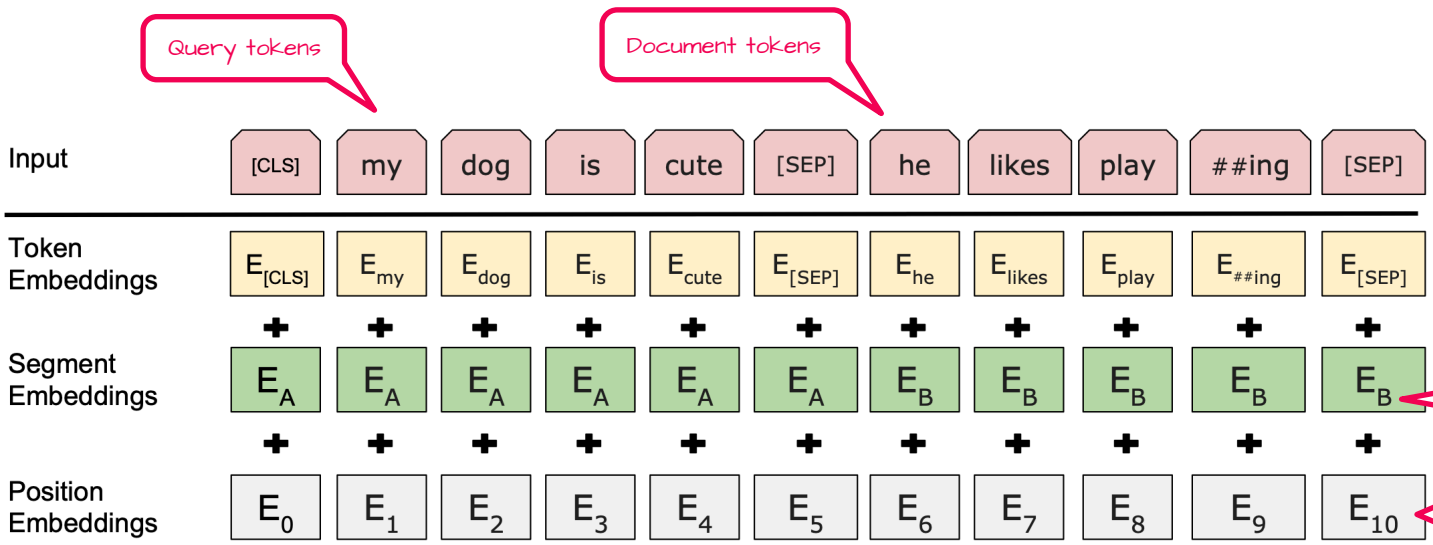
- A pre-trained BERT model already knows A LOT about language.
- After all, it was trained on 7000 books and the whole Wikipedia
- It means that it's REALLY good at "understanding" human language.
- We want to TRANSFER that knowledge to other tasks, like Question Answering, Sentence Classification, RANKING DOCUMENTS…



https://arxiv.org/abs/1810.04805

BERT - How to use in IR?

- You will need a base ranker (Say, BM25). It's NOT feasible (yet?) to score ALL documents in the collection with BERT.
- Fine-tune the model using a good retrieval collection. Say, MSMarco (https://microsoft.github.io/TREC-2019-Deep-Learning/)
- Input format is like this:



In practice, the Token ID. Most libraries will do it for you

Which sentence is this? 0 for query tokens, 1 for document tokens

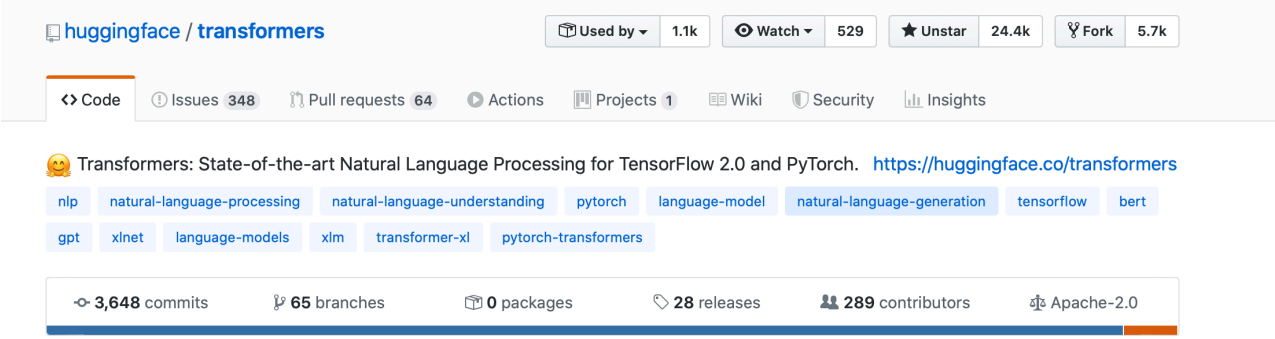
This is discussed in details in the BERT paper, as a sin function. In practice, don't worry. Library will probably take care of this too

BERT - Does it work for IR?

- Google uses it in production (https://www.blog.google/products/search/search-language-understanding-bert/)
- Bing uses (a version of) it in production (https://azure.microsoft.com/en-us/blog/bing-delivers-its-largest-improvement-in-search-experience-using-azure-gpus/)
- It dominates the MSMarco IR leaderboard. (https://microsoft.github.io/msmarco/)
- First model to be significantly better than traditional ones in Robust04 (https://www.aclweb.org/anthology/D19-3004.pdf)
- Overall, BERT is (specially) good for: (https://arxiv.org/pdf/1905.01758.pdf)
    - Rarer query terms
    - Documents with novel terms
    - Longer queries
- We have no clear idea WHY, though. (http://www.abcamara.com/documents/publications/ECIR_2020.pdf)

What else? (Other models)

- BERT inspired a number of other models. Some of the most important ones are:
  - ROBERTA (BERT with more training and tweaks) https://arxiv.org/abs/1907.11692
  - ALBERT (A little BERT, smaller and and almost as powerful) https://arxiv.org/abs/1909.11942
  - XLNet (Try to solve the problem of long documents. BERT is limited to 512 tokens) https://arxiv.org/abs/1906.08237
  - T5 (Multi-purpose, transfer-learning model) https://arxiv.org/abs/1907.11692
- Most models have a pretty good implementation, with lots of pre-trained models, for multiple languages (including Dutch) at 🤗 (no, really, that's the company name): https://github.com/huggingface/transformers



Many worthwhile issues to tackle

- BERT/Transformers is still NOT made-to-order for IR.
- Lack of large-scale real-life data for evaluation or training from scratch. (TREC-style evaluations will never scale up, we cannot leave the field to industry);
- It's not clear how to use **weak supervision** in IR
- So far, these models have been mostly applied to ad hoc retrieval and Web search, but of course IR is much bigger than that. **Conversational search**, **query suggestion, contextual search**... **How Bert (and friends) tackle these?**
- Clever tricks from the machine learning community are not yet often used (e.g. **curriculum learning, reinforcement learning...**)
- **Reproducibility** and benchmarking
- **Metrics** (of course!) for new tasks, e.g. answer generation

That's it for today!

Slack: in4325_2019.slack.com

Email: in4325-ewi@tudelft.nl

MSc project ideas? A.BarbosaCamara@tudelft.nl