Computing for Medicine

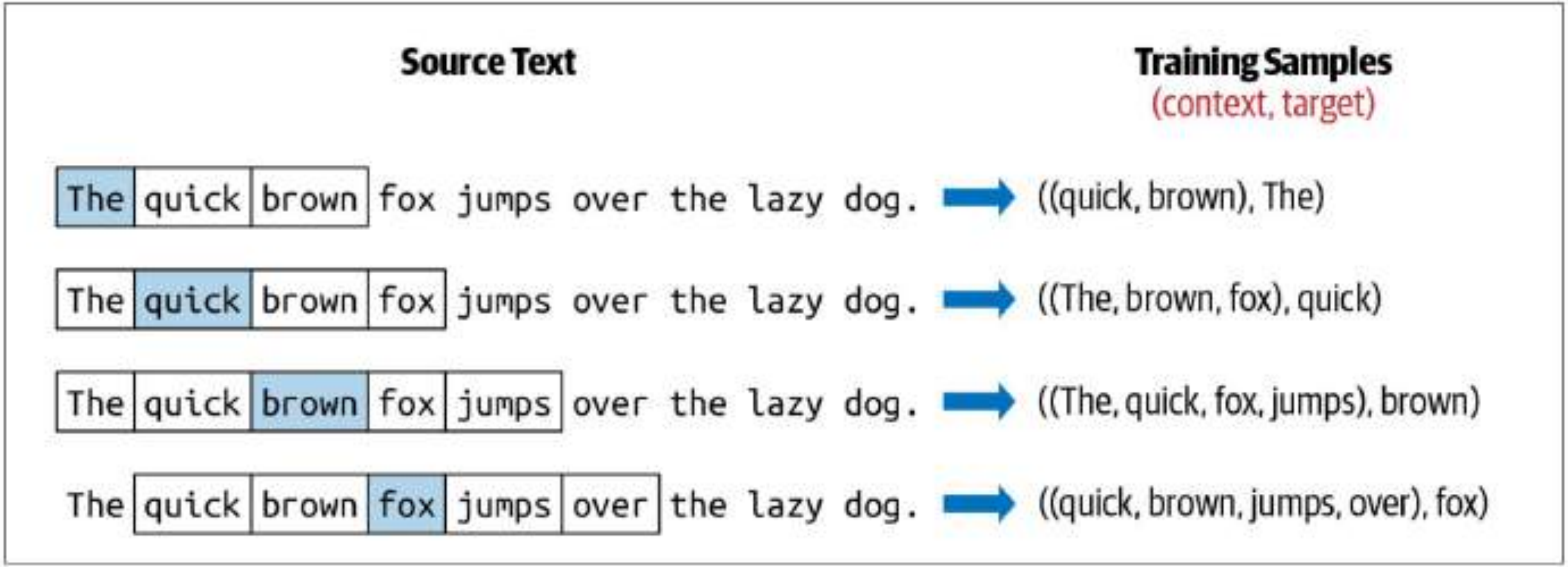
Google Classroom Code: dnd5qkt5

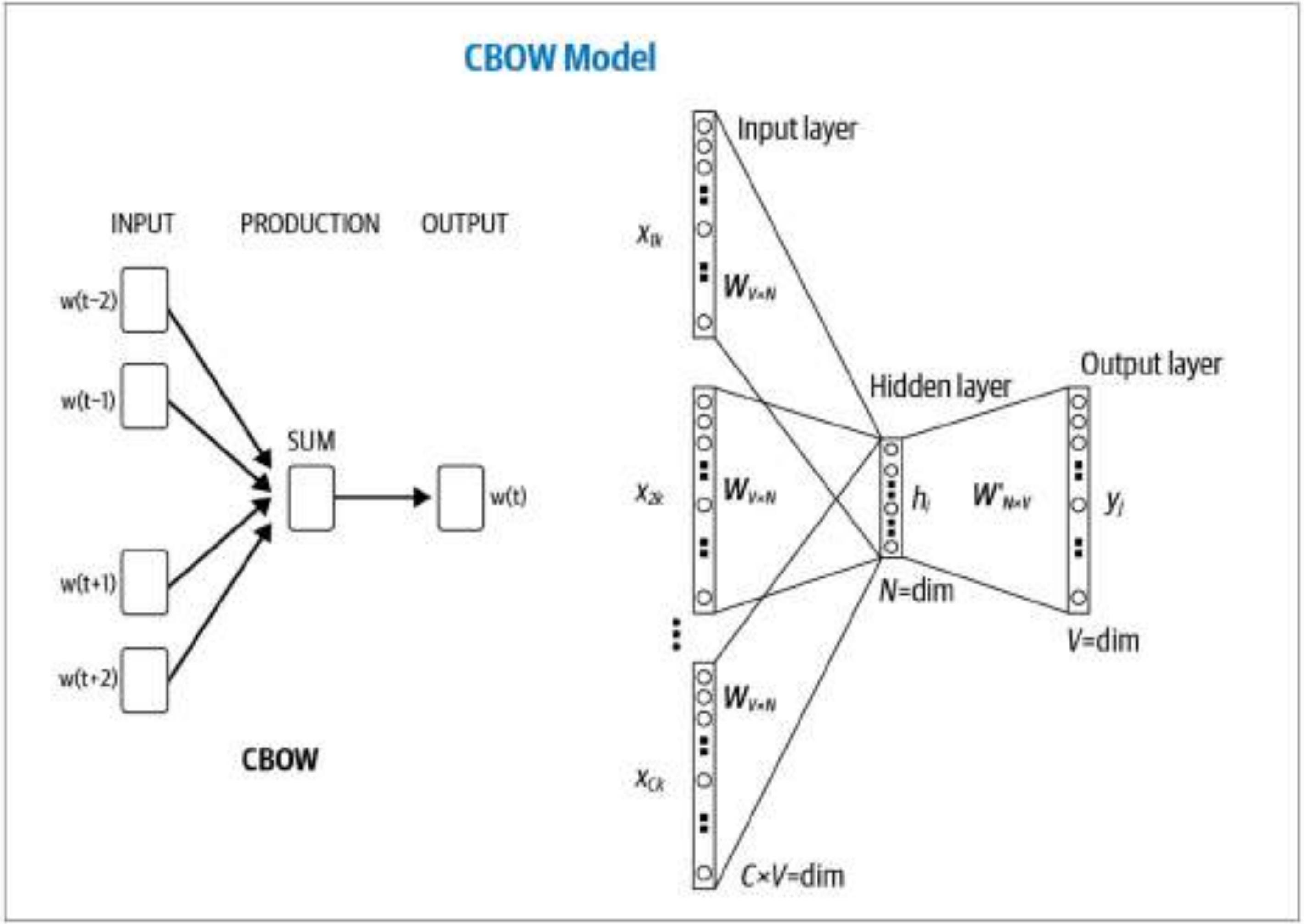
Monsoon 2025

Lecture 6

Word Vectors

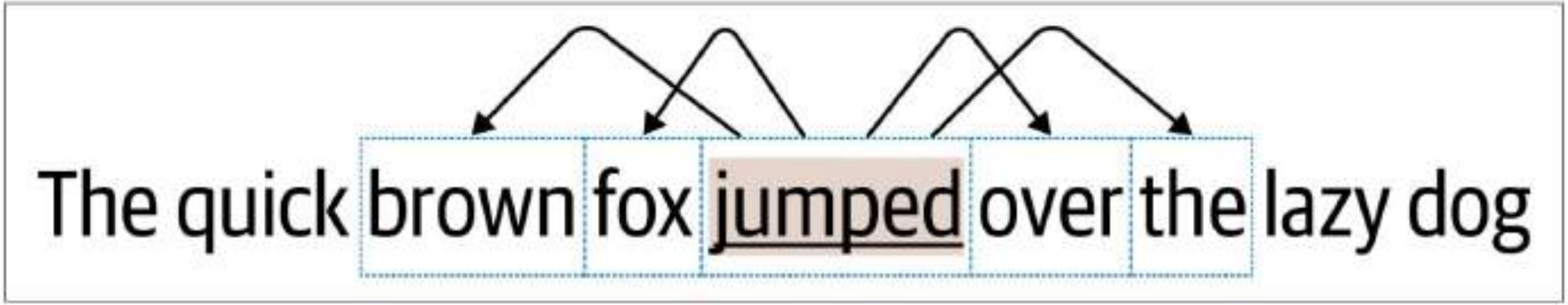
Training with CBOW



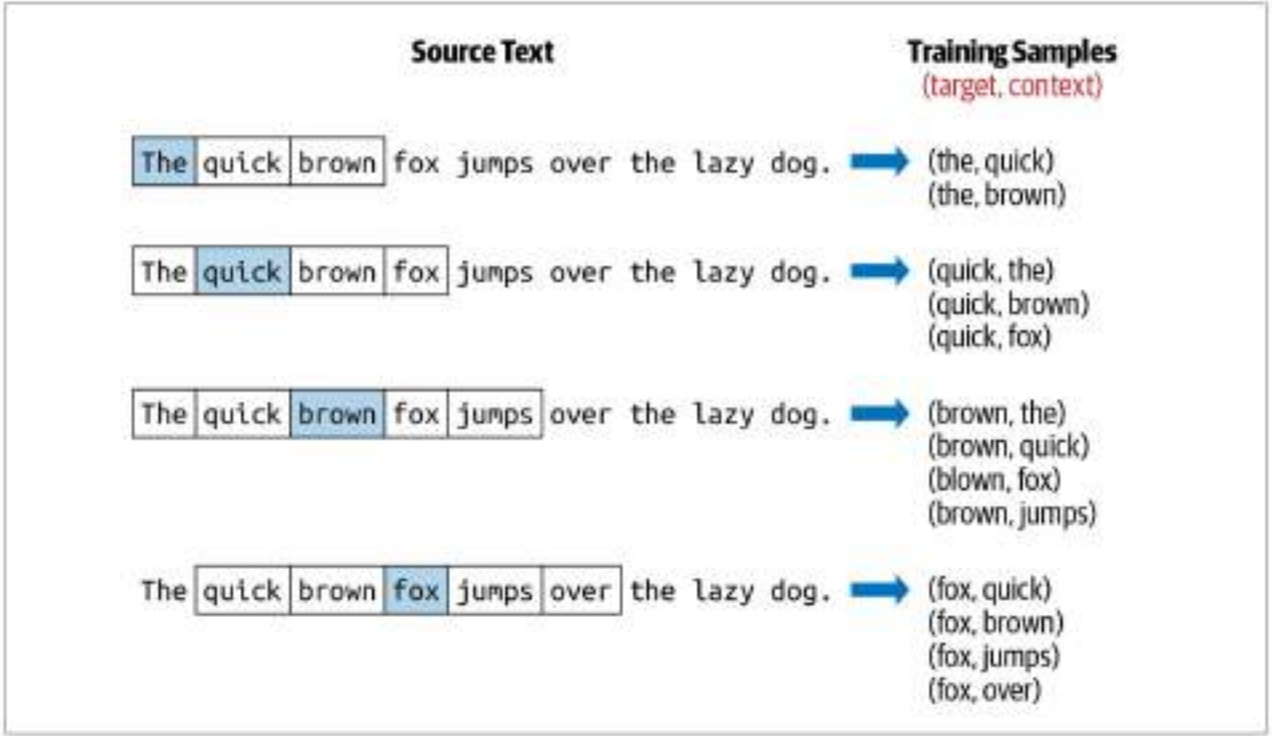

CBOW Model
INPUT
PRODUCTION
OUTPUT
w(t-2)
w(t-1)
w(t+1)
w(t+2)
SUM
w(t)
CBOW
Input layer
x_{1k}
$W_{V \times N}$
x_{2k}
$W_{V \times N}$
x_{Ck}
$W_{V \times N}$
$C \times V$=dim
Hidden layer
h_i
N=dim
$W'_{N \times V}$
Output layer
y_j
V=dim

word2Vec: Skip Gram Variant

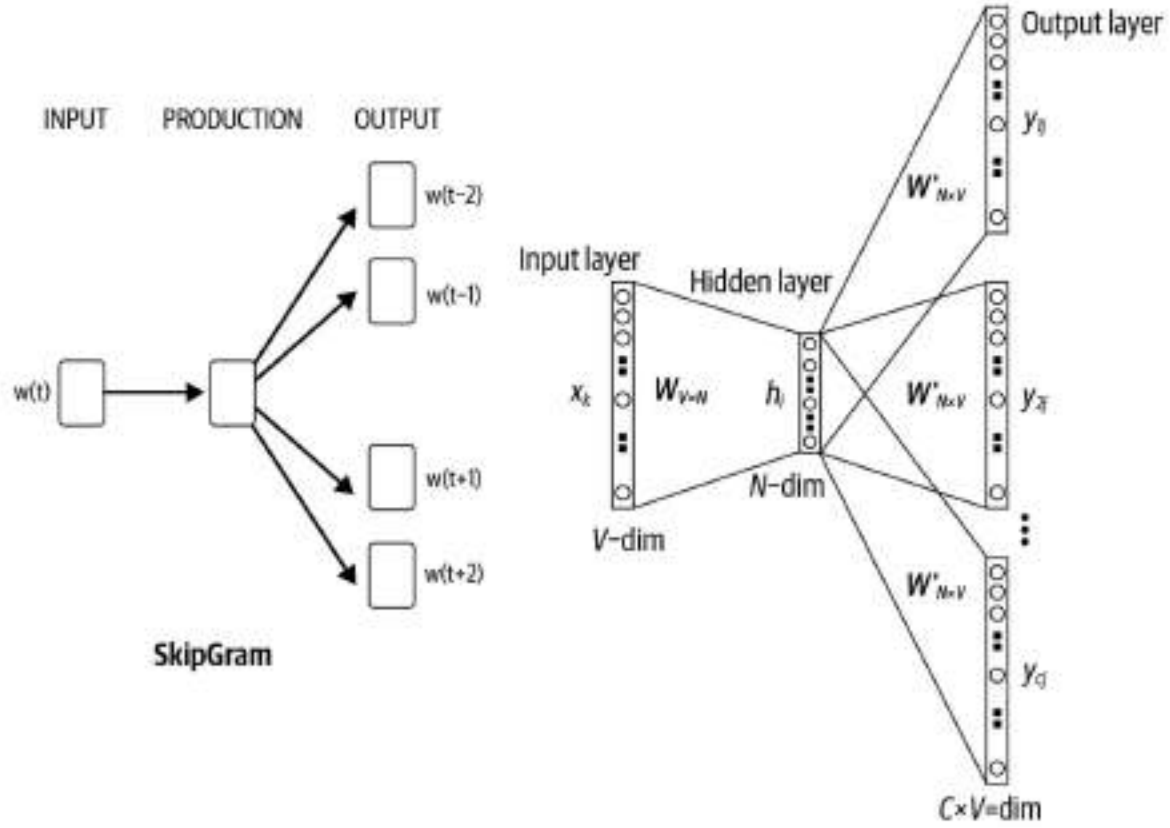
- Predicts the context given the middle word



Training with Skip Gram



Continuous SkipGram Model



Two Papers that changed the field:

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov
Google Inc.
Mountain View
mikolov@google.com

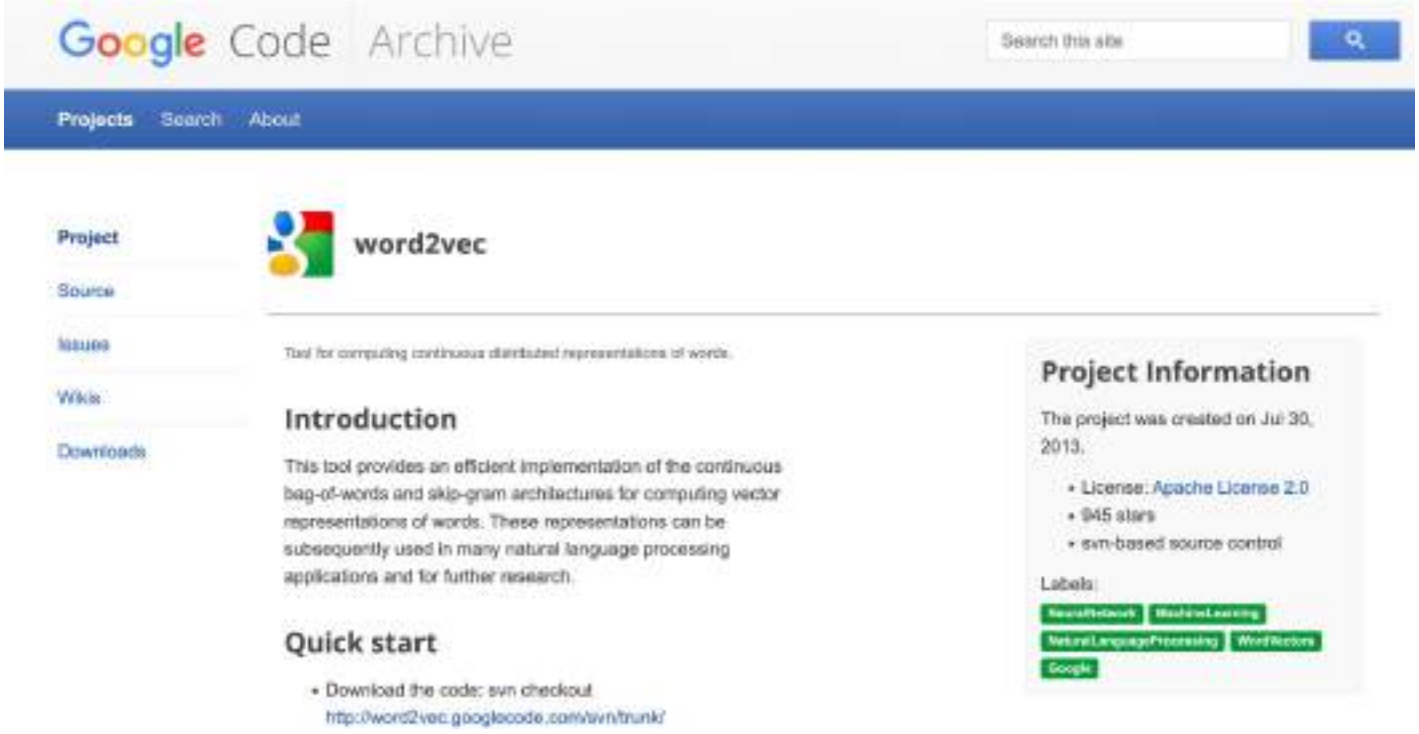
Ilya Sutskever
Google Inc.
Mountain View
ilyasu@google.com

Kai Chen
Google Inc.
Mountain View
kai@google.com

Greg Corrado
Google Inc.
Mountain View
gcorrado@google.com

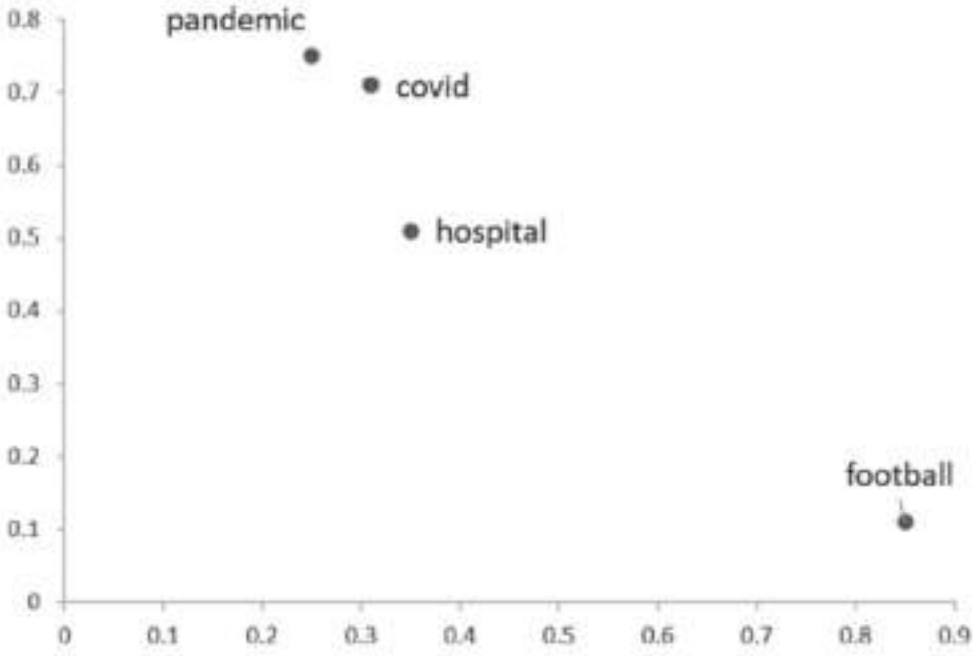
Jeffrey Dean
Google Inc.
Mountain View
jeff@google.com

Open Source Code, Multiple Implementations



Words in Vector Space

covid	[0.3 0.7]
pandemic	[0.3 0.8]
hospital	[0.4 0.5]
football	[0.9 0.1]



Summary of Embeddings

- Embeddings capture distributional similarities between words
- Allow efficient word algebra and analogies
- Word2Vec - Pre-trained Neural Network based low dimensional word embedding from Google (Atomic Unit = Word)
- GloVe- Pretrained word embedding from Stanford (Atomic Unit = Word)
- Fasttext- Pretrained word embedding from Facebook (Atomic Unit = Character)

Before Making your own Embedding

Available embeddings for clinical data and concepts. Since ELMo models use character information and BERT models use sub-word information, they can generate a representation for any concept.

Name	Model	Data/Concepts	Terms	Dim.
PubMed-w2v.bin[a]	word2vec	PubMed	2.4 M	200
PMC-w2v.bin[b]	word2vec	PubMed Central	2.5 M	200
PubMed-and-PMC-w2v.bin[c]	word2vec	PubMed, PubMed Central	4.1 M	200
wikipedia-pubmed-and-PMC-w2v.bin[d]	word2vec	PubMed, PubMed Central, Wikipedia	5.5 M	200
drug word embeddings[e]	word2vec	PubMed, DrugBank	553,195	420
AWE-CM [49]	word2vec	UMLS CUI (concepts)	265 M	300
claims_codes_hs_300 [55]	word2vec	ICD-9 codes (concepts)	51,327	300
claims_cuis_hs_300 [55]	word2vec	UMLS CUI (concepts)	14,852	300
cui2vec [56]	word2vec/GloVe	UMLS CUI (concepts)	108,477	500
concept embeddings [58]	AiTextML	MeSH ID (concepts)	26,103	100
word embeddings [58]	AiTextML	PubMed	513,196	100
ELMo (PubMed model) [11]	ELMo	PubMed	NA	1024
BioBERT [15]	BERT	PubMed	NA	768/1024
ClinicalBERT [16,17]	BERT	MIMIC III	NA	768

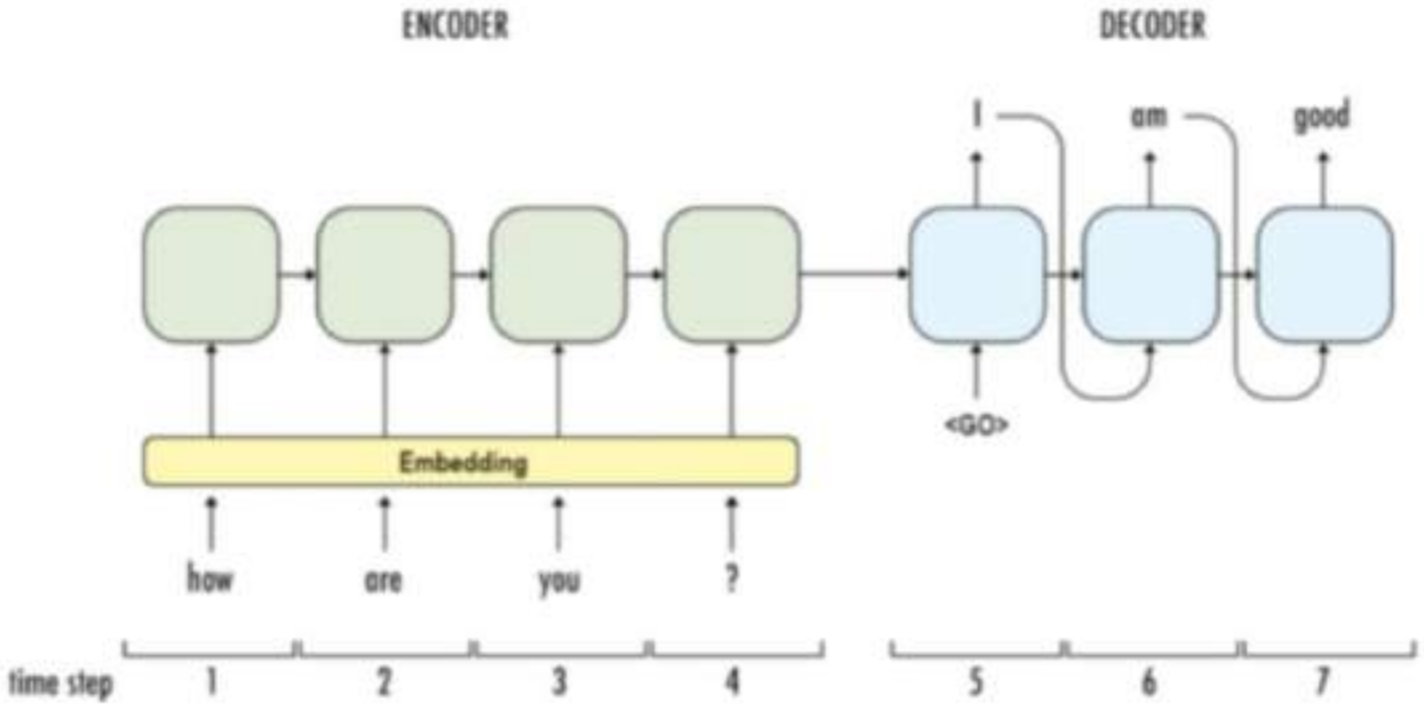
[a] http://evexdb.org/pmresources/ngrams/PubMed/.
[b] http://evexdb.org/pmresources/ngrams/PMC/.
[c] http://evexdb.org/pmresources/vec-space-models/wikipedia-pubmed-and-PMC-w2v.bin.
[d] http://evexdb.org/pmresources/vec-space-models/wikipedia-pubmed-and-PMC-w2v.bin.
[e] https://github.com/chop-dbhi/drug_word_embeddings.

Evaluation Tasks

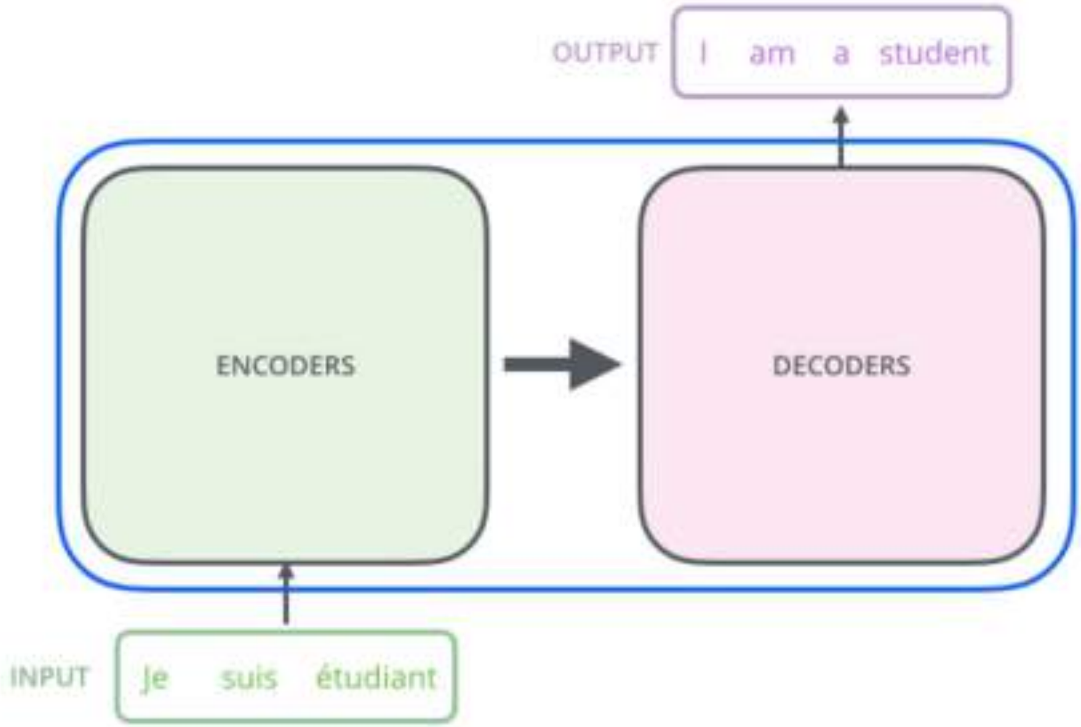
	Word embeddings		Evaluation	
Paper	Corpora	Model	Intrinsic	Extrinsic
De Vine et al., 2015 [30]		word2vec		clinical information extraction
Chiu et al., 2016 [72]	PMC, Pubmed, PMC + Pubmed	word2vec	relatedness, similarity	NER
Dubois et al., 2017 [32]	text notes, OHSUMED	GloVe	–	disease prediction, mortality prediction
Wang et al., 2018 [29]	Mayo Clinic text notes, PubMed, Wikipedia, Google News	GloVe	similarity (qualitative),	clinical information extraction, reaction extraction
Huang et al., 2016 [46]	MedHelp online forum, PubMed, Wikipedia		cluster quality evaluation	
Choi et al., 2016 [55]	OHSUMED, medical claims	word2vec	conceptual similarity, medical relatedness	–
Yu et al., 2016 [52])	UMLS and MeSH terms	LDA	UMLS-Similarity	–
Mencia et al., 2016 [58]	BioASQ, PubMed	All-in-text	MiniMayoSRS, UMNSRS similarity/relatedness	–
Bong et al., 2017 [49]	MIMIC-III	word2vec	similarity	–
Patel et al., 2017 [35]	PubMed, medical claims	word2vec	medical term similarity	medical coding review
Beam et al., 2018 [56]	PubMed, medical claims, UMLS semantic types	word2vec, Glove	comorbidity, causative, and drug-conditions relations and UMNSRS similarity/relatedness	
Zhao et al., 2018 [43]	PubMed, DrugBank	word2vec	UMNSRS similarity/relatedness	drug name recognition/classification
Craig et al., 2017 [37]	discharge notes	word2vec (Skipgram)	—	30-day unplanned prediction
Nguyen et al., 2017 [38]	hospital patient records	random init, word2vec	cluster evaluation	unplanned readmission within 6 months prediction
Pham et al., 2016 [39]	hospital patient records	random init + advanced techniques	—	unplanned readmission prediction, high-risk patient prediction
Escudié et al., 2018 [33]	electronic health records from hospital	three-layer stack of denoising autoencoders	—	disease prediction
Gehrmann et al., 2018 [36]	discharge summaries from MIMIC-III	word2vec	word similarity	phenotype classification
Wang et al., 2018 [29]	clinical notes from hospital, PMC, News, Wikipedia + Gigaword	word2vec (Skip-gram)	word/semantic similarity	information extraction, smoking status prediction, fracture detection,
Kholgi et al., 2016 [34]	i2b2/VA 2010 [81], ShARe/CLEF 2013 eHealth Evaluation Lab [82]	word2vec (Skip-gram)	—	disease prediction, term extraction

Attention and Transformer

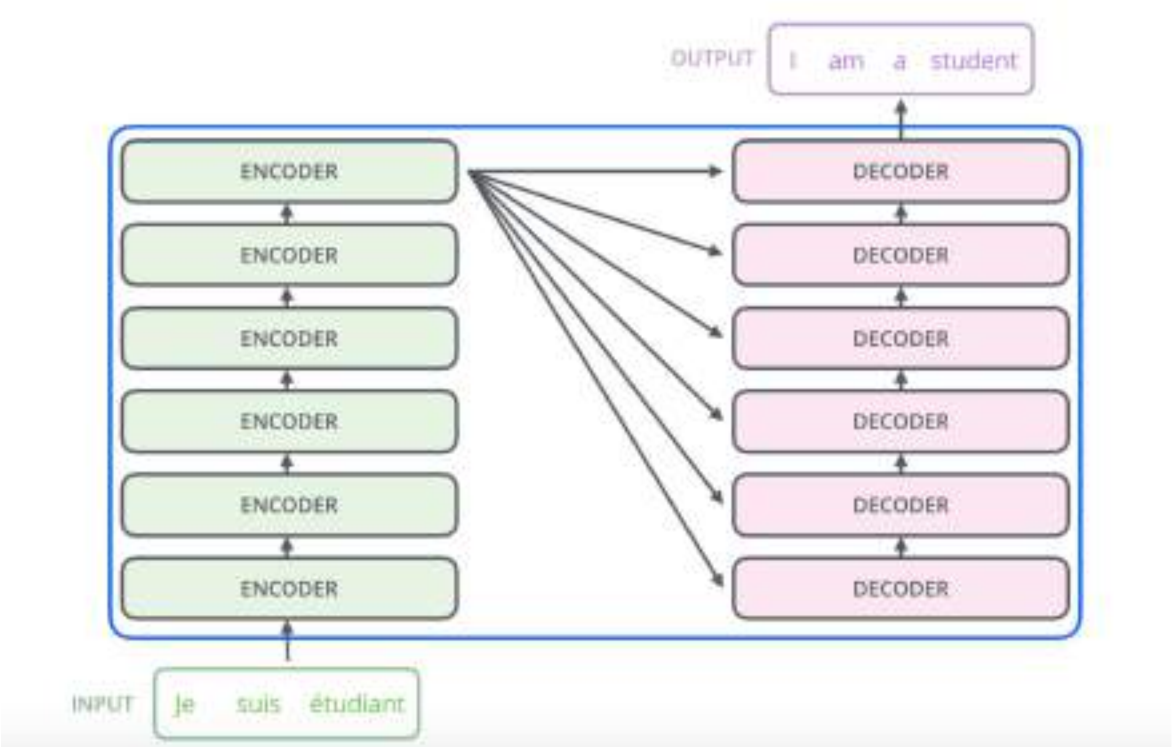
Sequence to Sequence Modeling



Sequence to Sequence Modeling



Sequence to Sequence Modeling



How do we model a whole sentence?

Attention Is All You Need

Ashish Vaswani[*]
Google Brain
avaswani@google.com

Noam Shazeer[*]
Google Brain
noam@google.com

Niki Parmar[*]
Google Research
nikip@google.com

Jakob Uszkoreit[*]
Google Research
usz@google.com

Llion Jones[*]
Google Research
llion@google.com

Aidan N. Gomez[*, †]
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser[*]
Google Brain
lukaszkaiser@google.com

Illia Polosukhin[*, ‡]
illia.polosukhin@gmail.com

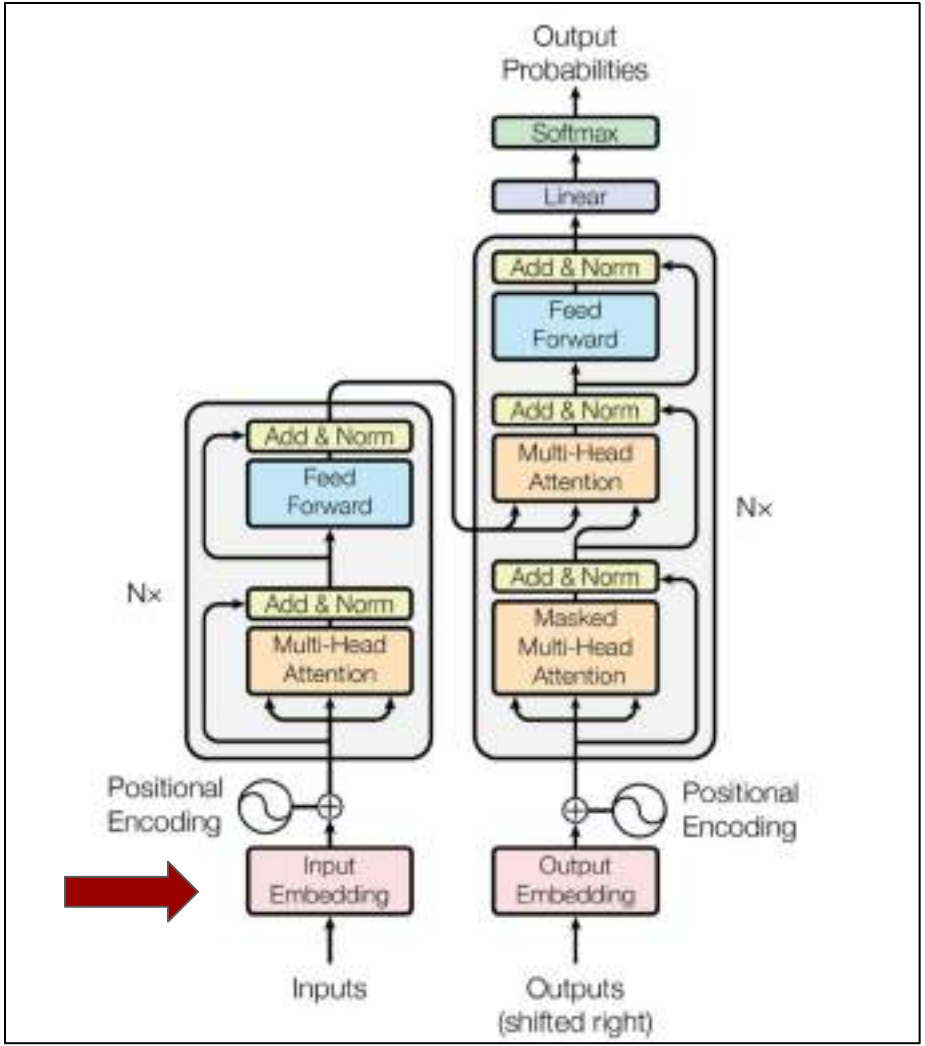
https://papers.nips.cc › paper › 7181-attention-is-all-yo...

Attention is All you Need - NIPS papers

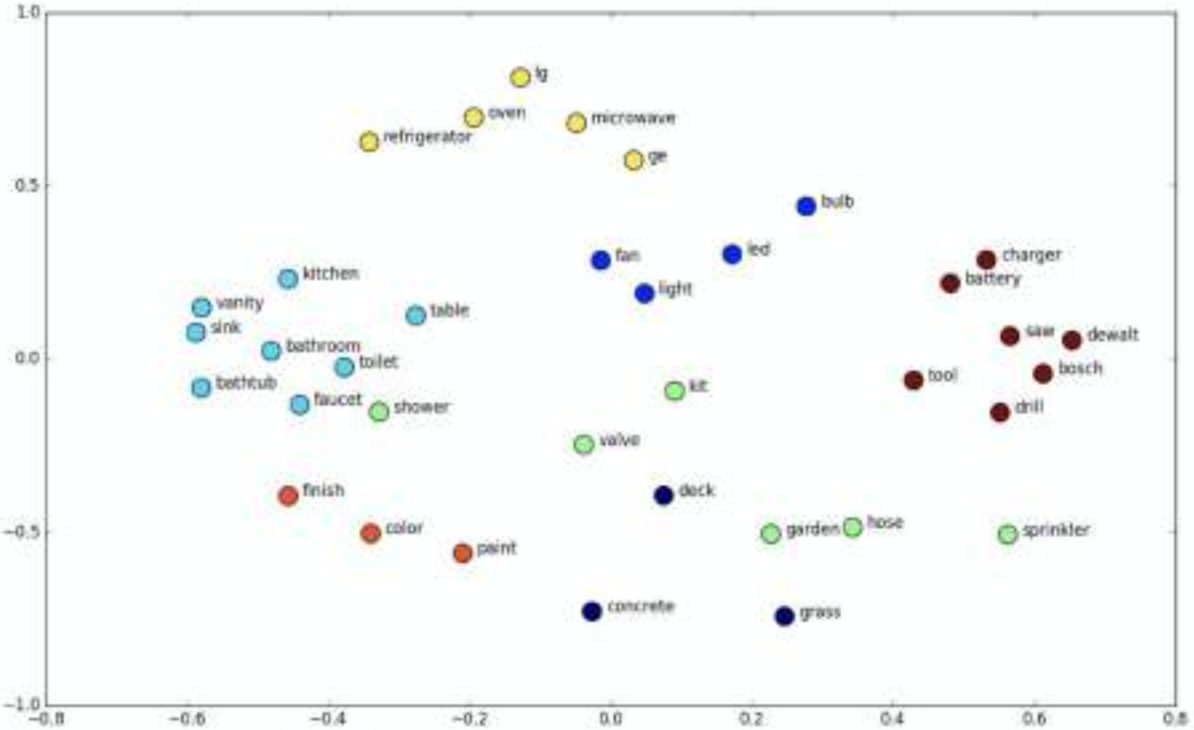
by A Vaswani · 2017 · Cited by 52845 — Authors. Ashish **Vaswani**, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin ...

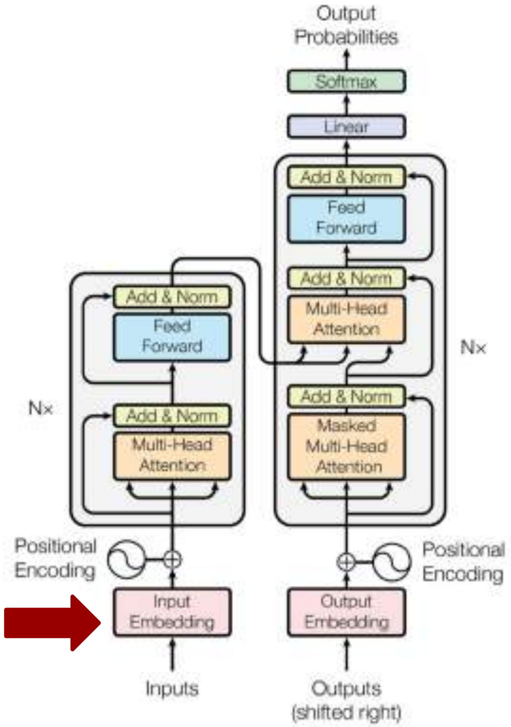
Transformer

Let's break it down

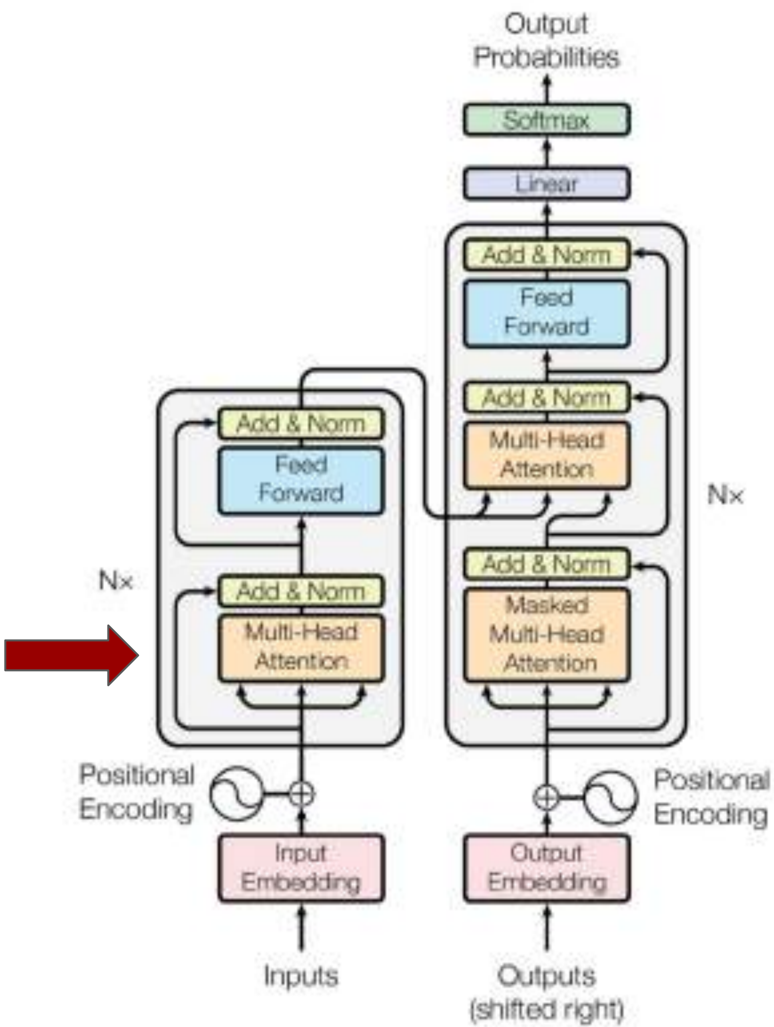


Input Embeddings



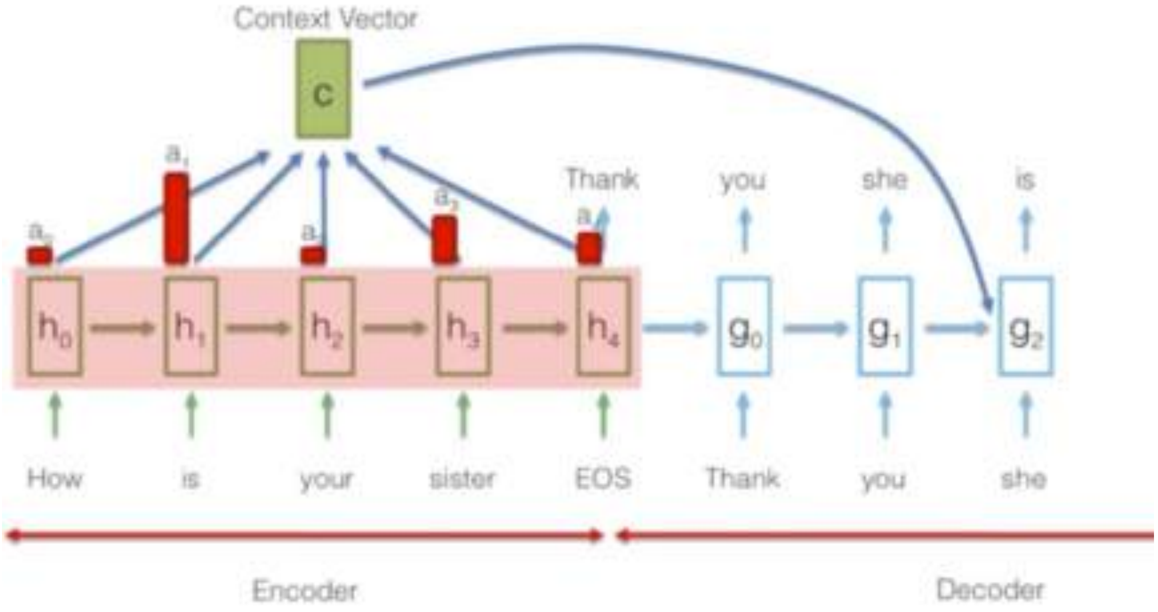


https://becominghuman.ai/attention-is-all-you-need-16bf481d8b5c



Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

Attention Mechanism



$$c_i = \sum_j \alpha_{ij} h_j$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}$$

$$e_{ij} = a(g_{i-1}, h_j)$$

Self Attention

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

Self Attention

https://jalammar.github.io/illustrated-transformer/
https://www.topbots.com/transformers-timesformers-and-attention/

Query, Key and Value

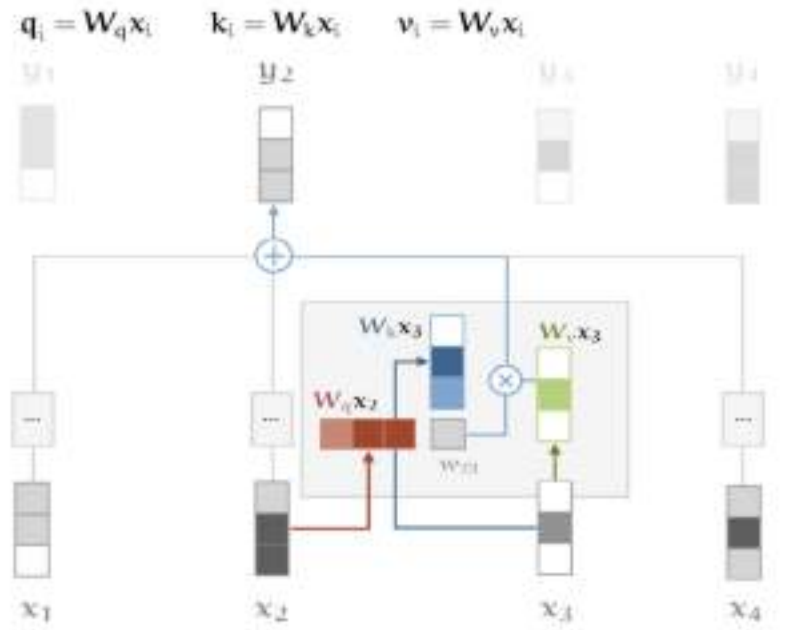


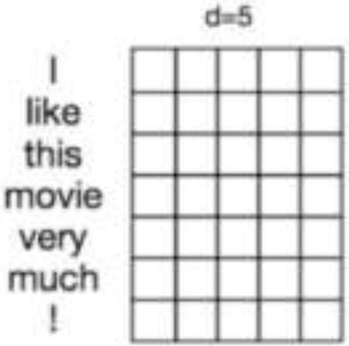
Illustration of the self-attention with key, query and value

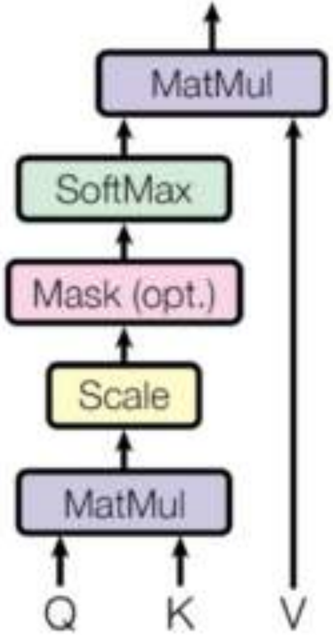
"Transformers from scratch" by Peter Bloem [2]

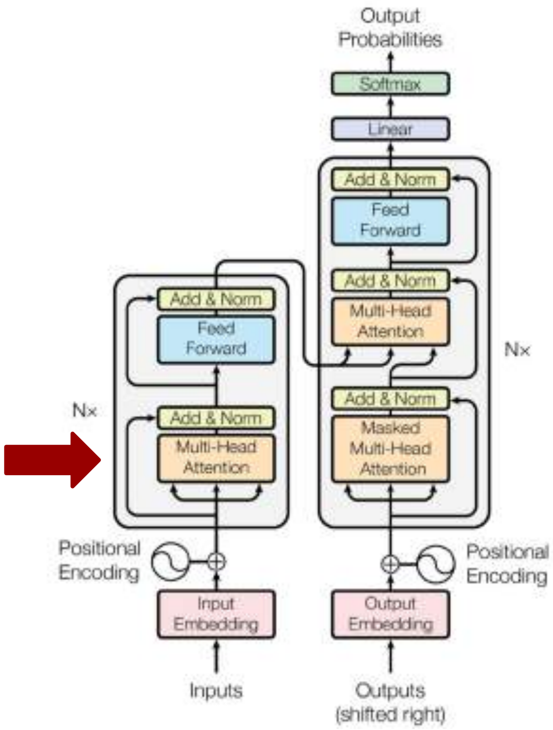
Then we use the Q, K and V matrices to calculate the attention scores. **The scores measure how much focus to place on other places or words of the input sequence w.r.t a word at a certain position.** That is, the dot product of the query vector with the key vector of the respective word we're scoring. So, for position 1 we calculate the dot product (.) of `q1` and `k1`, then `q1 . k2`, `q1 . k3` and so on,...

Scaled Dot Product (Similarity with Context)

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

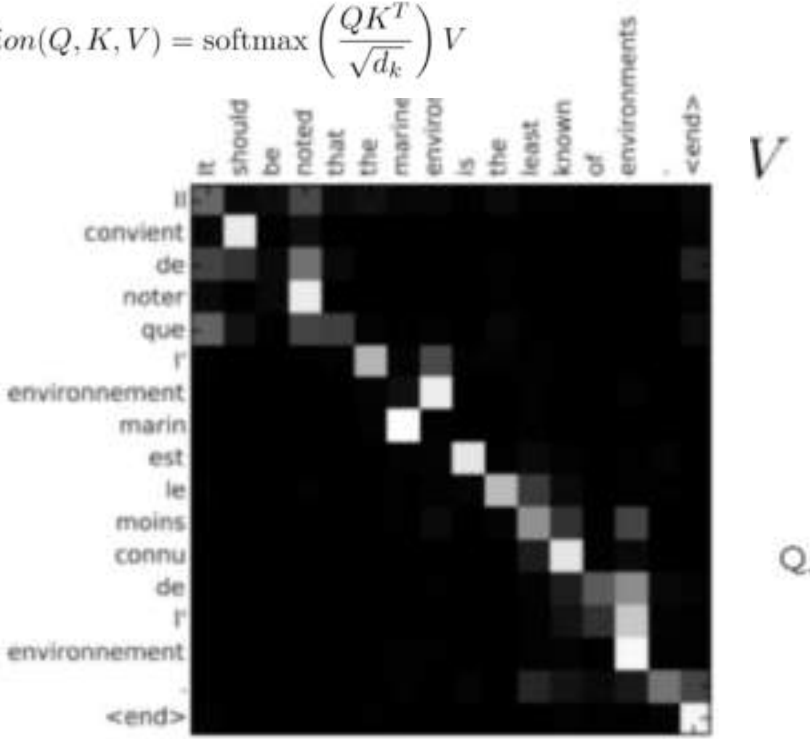


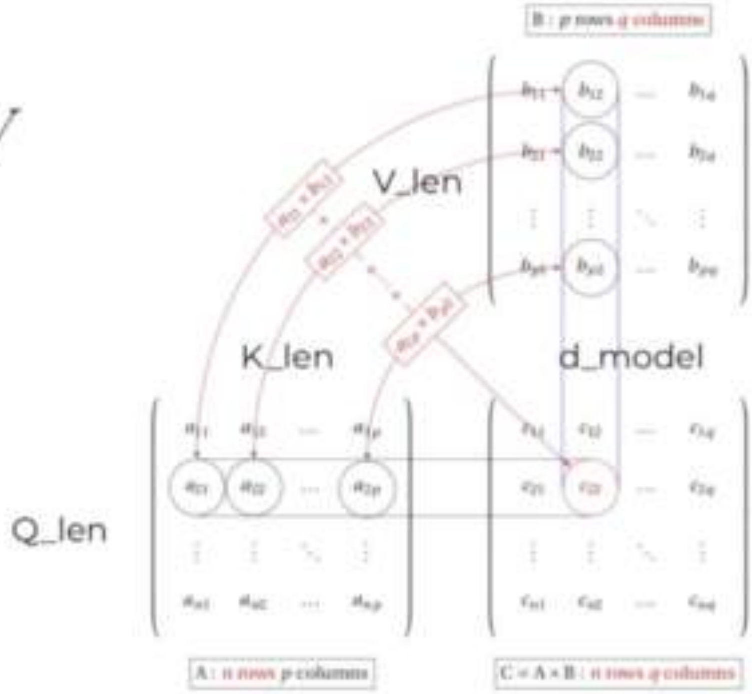




Mechanics of Attention

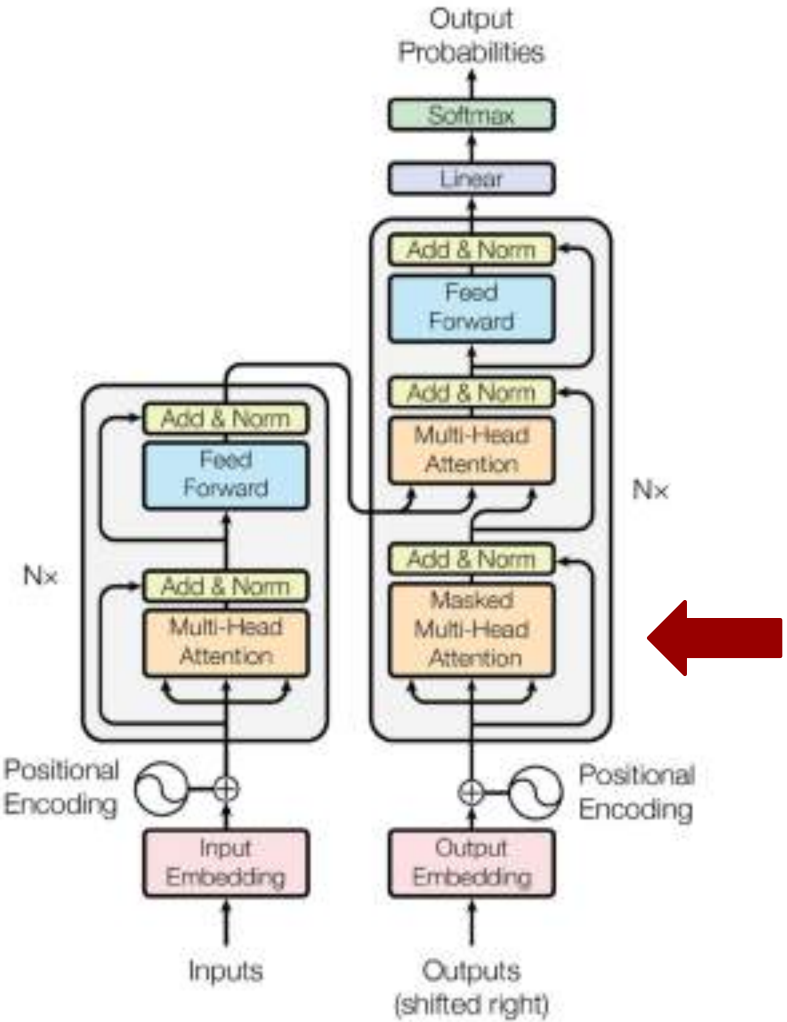
$$Attention(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



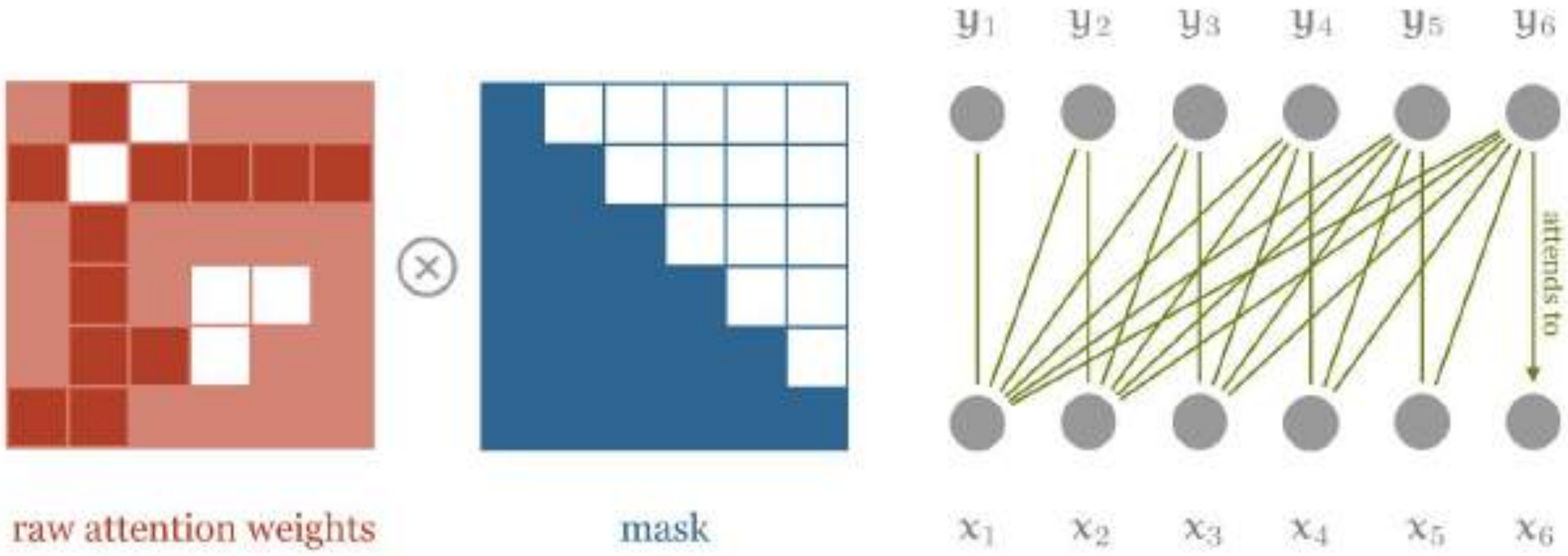


Masked Attention

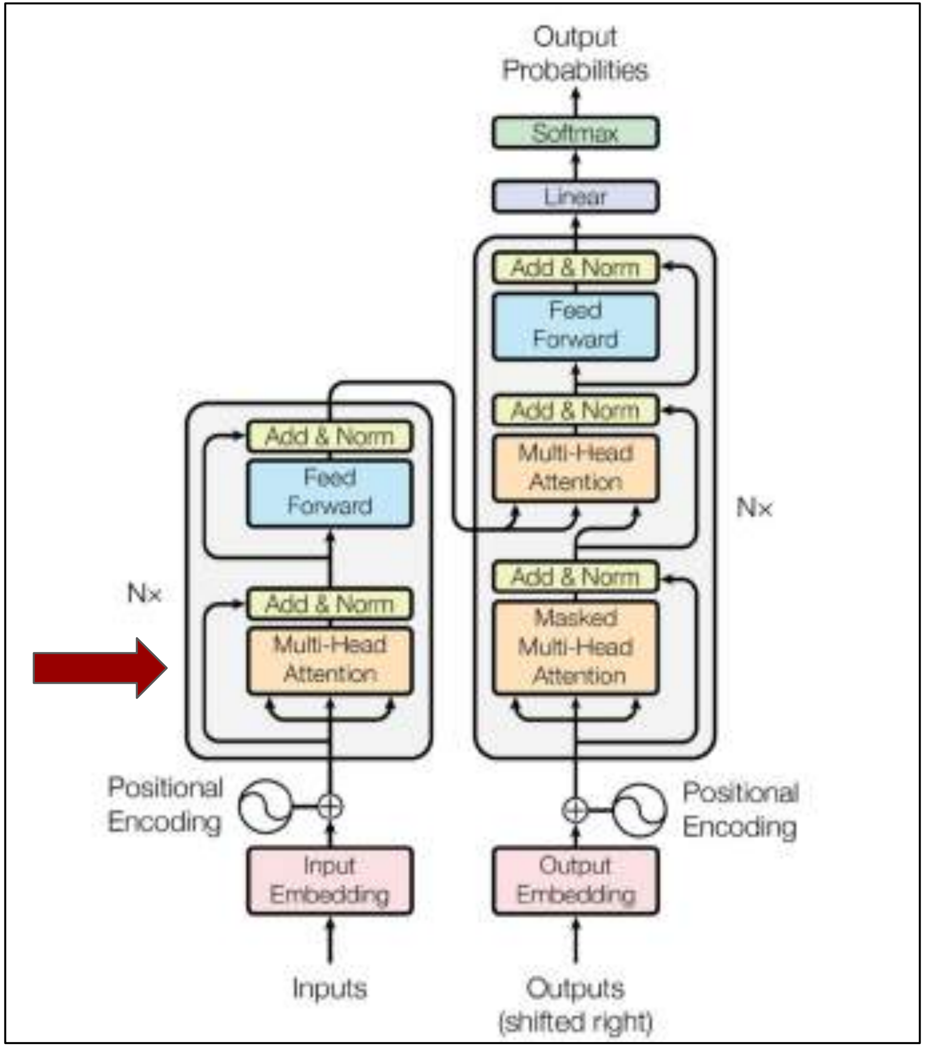
$$QK^T \quad * \quad \begin{array}{|c|c|c|c|c|} \hline 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 \\ \hline 1 & 1 & 1 & 0 & 0 \\ \hline 1 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$



Look ahead mask



http://peterbloem.nl/blog/transformers


Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Add & Norm
Masked
Multi-Head
Attention
Add & Norm
Feed
Forward
Nx
Add & Norm
Multi-Head
Attention
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)

Thanks for attending the class!