Towards Common-Sense Reasoning with Advanced NLP architectures

Alex Movilă - November 2019

February 2019:

Researchers, scared by their own work, hold back “deep-fakes for text” AI

OpenAI Trains Language Model, Mass Hysteria Ensues

This New Storytelling AI Fools Humans 3 out of 5 Times With It’s Writing

Artificial Intelligence Can Now Write Amazing Content - What Does That Mean For Humans?

The first AI-generated textbook: "Lithium-Ion Batteries: A Machine-Generated Summary of Current Research"

Oct 2019:

Elon Musk-backed AI company releases its text-generating robot despite concerns it could be used to create fake news and spam

GPT-2 generates convincing fake news

Completion

The state legislature has passed a bill banning artificial intelligence in New York. This could be devastating for New York City-based software giants like Google and Amazon.

This bill will make it illegal to use artificial intelligence to drive cars in New York City.

There's more good news, too, and it's not just in the form of jobs. Artificial intelligence has already saved countless lives.

Artificial Intelligence Is Transforming Life in China

In December, the government of China announced a major project to transform their national infrastructure into smart cities. The goal of the project is to transform cities into smart, automated systems where everything is connected to the internet.

This means that people will have access to the internet, cars will drive themselves, and even the national government itself will be a computer.

That's because the goal of the program is to completely automate everything in a city, from power to garbage collection.

This isn't the first time the country has made such a move. Over the

Share

Humans find GPT-2 outputs convincing. Our partners at Cornell University surveyed people to assign GPT-2 text a credibility score across model sizes. People gave the 1.5B model a "credibility score" of 6.91 out of 10.

…

extremist groups can use GPT-2 for misuse, specifically by fine-tuning GPT-2 models on four ideological positions: white supremacy, Marxism, jihadist Islamism, and anarchism. CTEC demonstrated that it's possible to create models that can generate synthetic propaganda for these ideologies.

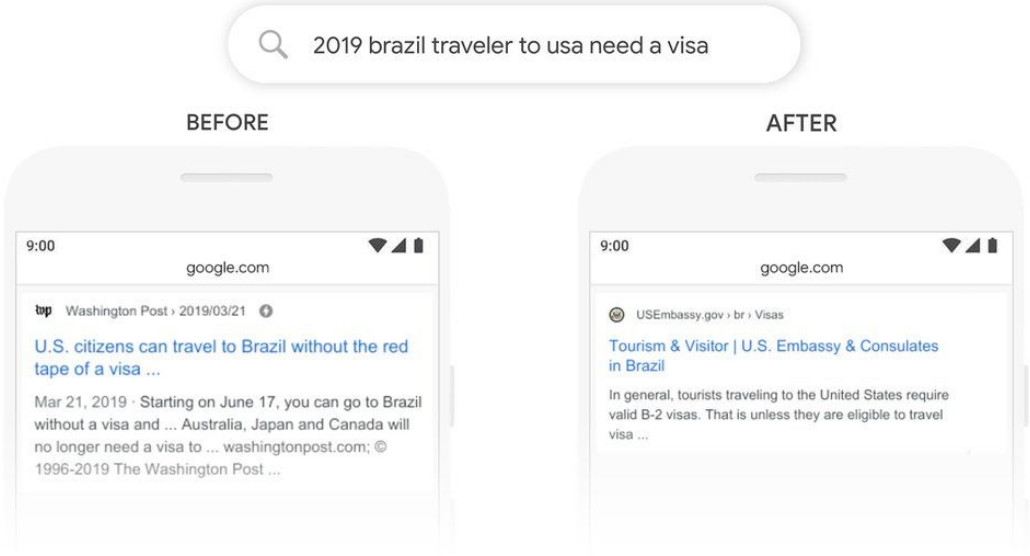
GPT-2: 1.5B Release

[IASI AI] Google improved 10 percent of searches by understanding language context (with BERT)

"*how to catch a cow fishing?*"

Even though I had purposely used the word "fishing" to provide context, Google ignored that context and provided **results related to cows**. That was on **October 1, 2019**.

Today, **October 25, 2019** the same query results in search results that are full of striped bass and **fishing related results**.



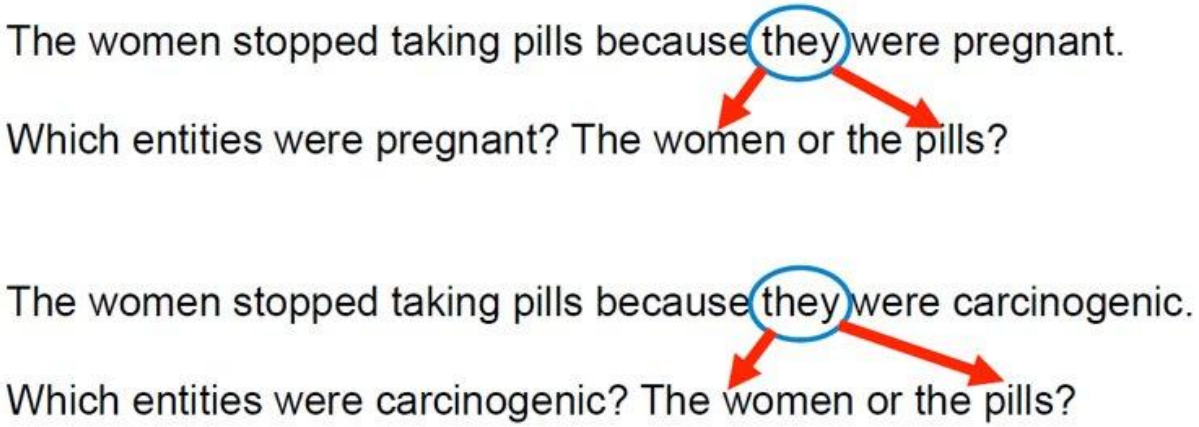
BERT = (Bidirectional Encoder Representations from Transformers)

Understanding searches better than ever before , Google is improving 10 percent of searches by understanding language context

Google Search Updated with BERT – What it Means

Winograd Schemas Challenge (WSC) (ambiguous text) - from 56% in 2018 to 93.8% in 2019

State-of-the-Art (SOTA) models in **Feb 8, 2018: 56% accurate**
Humans: 100% accurate, Chance: 50% accuracy

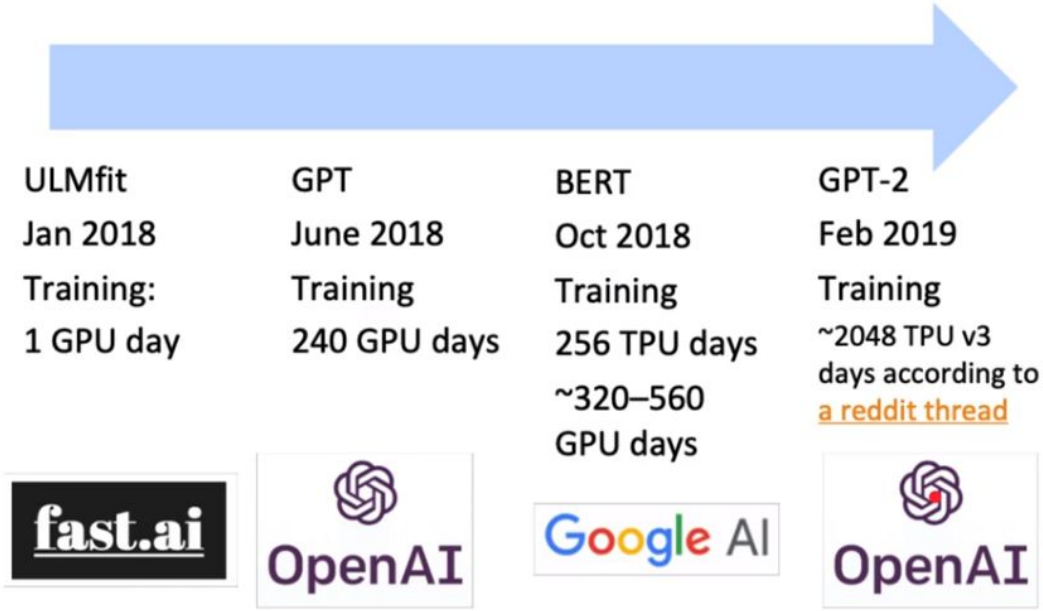


*4 Aug 2019: By fine-tuning the BERT language model both on the introduced and on the WSCR dataset, we achieve overall accuracies of **72.5%** and 74.7% on WSC273 and WNLI, improving the previous state-of-the-art solutions by 8.8% and 9.6%, respectively.*

Oct 2019: T5 model - 93.8%

A Better Turing Test - Winograd Schemas ,
Want to Know the Trick to Achieve Robust Winograd Schema Challenge Results? (Paper)

[IASI AI] Training time for BERT based models with transformer architecture - 2 years of training on 1 GPU



HOW LONG DOES IT TAKE TO PRE-TRAIN BERT?
BERT-base was trained on 4 cloud TPUs for 4 days and BERT-large was trained on 16 TPUs for 4 days. There is a recent paper that talks about bringing down BERT pre-training time – Large Batch Optimization for Deep Learning: Training BERT in 76 minutes.

HOW LONG DOES IT TAKE TO FINE-TUNE BERT?
For all the fine-tuning tasks discussed in the paper it takes at most 1 hour on a single cloud TPU or a few hours on a GPU.

Parameter size of latest NLP models (BERT based mostly) - up to 8 billion params

Here's are some of the latest large models and their **size in millions of parameters**.

Smaller, faster, cheaper, lighter: Introducing DistilBERT, a distilled version of BERT

GLUE [Wang et al., 2018] and **SuperGLUE** [Wang et al., 2019b]
each comprise a collection of text classification tasks meant to test general language understanding abilities:

- Sentence acceptability judgment (CoLA)
- Sentiment analysis (SST-2)
- Paraphrasing/sentence similarity (MRPC, STS-B, QQP)
- Natural language inference (MNLI, QNLI, RTE, CB)
- Coreference resolution (WNLI and WSC)
- Sentence completion (COPA)
- Word sense disambiguation (WIC)
- Question answering (MultiRC, ReCoRD, BoolQ)

SuperGLUE was designed to comprise of tasks that were
"beyond the scope of current state-of-the-art systems, but solvable by most college-educated English speakers"

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

SuperGLUE leaderboard

Rank	Name	Model	URL	Score	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC	AX-b	AX-g
1	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8	89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0	76.6	99.3/99.7
2	T5 Team - Google	T5		88.9	91.0	93.0/96.4	94.8	88.2/62.3	93.3/92.5	92.5	76.1	93.8	65.6	92.7/91.9
3	Facebook AI	RoBERTa		84.6	87.1	90.5/95.2	90.6	84.4/52.5	90.6/90.0	88.2	69.9	89.0	57.9	91.0/78.1

Google - T5 Team code (trained on 750GB text):
Our text-to-text framework provides a simple way to train a single model on a wide variety of text tasks using the same loss function and decoding procedure.

Facebook AI:

Congratulations to our AI team for matching the top GLUE benchmark performance! We believe strongly in open & collaborative research and thank @GoogleAI for releasing BERT. It led to RoBERTa, our robustly optimized system that was trained longer, on more data.

Example AX-b:
1001 Jake broke. Jake broke the vase. => not_entailment
1002 He is someone of many talents. He has many talents. => entailment

T5 - Unified Text-to-text Transfer Transformer model - one model to rule them all

We perform a systematic study of transfer learning for NLP using a unified text-to-text model, then push the limits to achieve SoTA on GLUE, SuperGLUE, CNN/DM, and SQuAD.

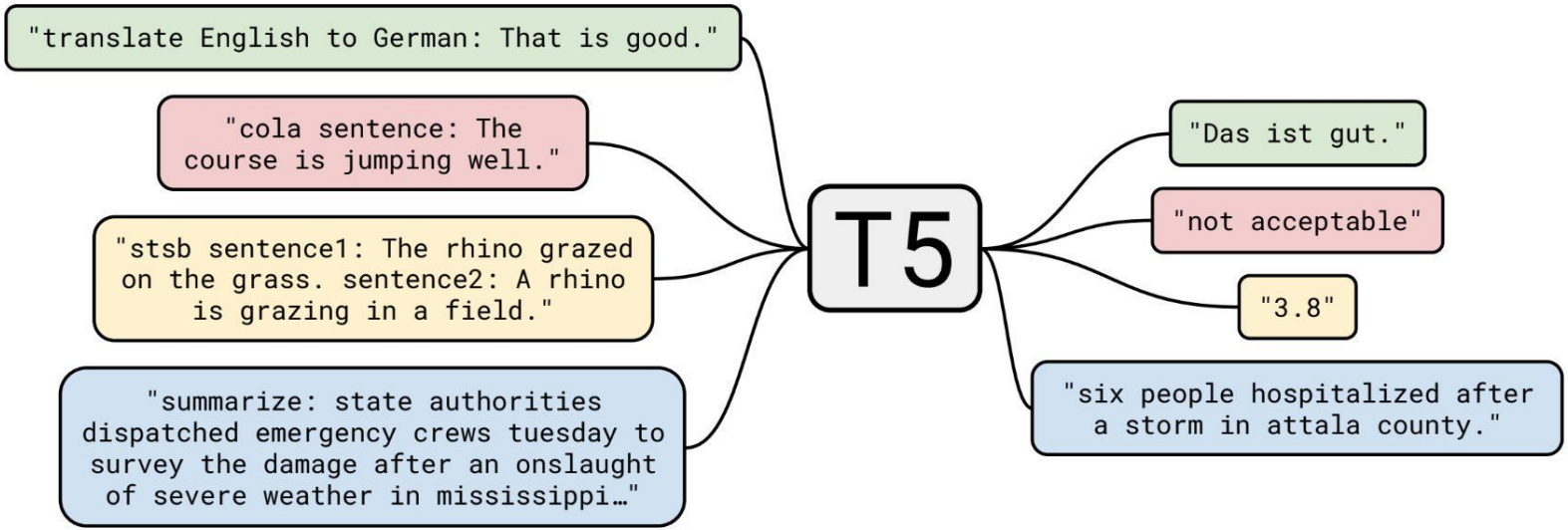


Figure 1: A diagram of our text-to-text framework. Every task we consider – including translation, question answering, and classification – is cast as feeding our model text as input and training it to generate some target text. This allows us to use the same model, loss function, hyperparameters, etc. across our diverse set of tasks. It also provides a standard testbed for the methods included in our empirical survey. "T5" refers to our model, which we dub the "**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer".

T5 - Code on Github

Reading comprehension (RC)—in contrast to information retrieval—requires integrating information and reasoning about events, entities, and their relations

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph. How will your system compare to humans on this task?

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Sep 18, 2019	ALBERT (ensemble model) *Google Research & TTIC* https://arxiv.org/abs/1909.11942	**89.731**	**92.215**
2 Jul 22, 2019	XLNet + DAAF + Verifier (ensemble) *PINGAN Omni-Sinitic*	88.592	90.859
2 Sep 16, 2019	ALBERT (single model) *Google Research & TTIC* https://arxiv.org/abs/1909.11942	88.107	90.902

Reading Comprehension Model Betters Humans, Tops SQuAD2.0 Leaderboard, SQuAD2.0 Dataset, State-of-the-art leaderboards, MS MARCO Leaderboard, Alibaba neural net defeats human in global reading test, NLP-progress - Question answering,

NLP in 2019 – Reasoning – Getting closer to humans - see leaderboard for RACE dataset

Why RACE dataset is more challenging and interesting?

- reading comprehension task designed for middle and high-school English exams in China, … consequently requires non-trivial reasoning techniques.

RACE has a wide variety of question types:

- What is the best title of the passage? (Summarization)
- What was the author's attitude towards the industry awards? (Inference)
- Which of the following statements is WRONG according to the passage? (Deduction)
- If the passage appeared in a newspaper, which section is the most suitable one? (Inference)
- The first postage stamp was made _ . (Context matching)

Model	Report Time	Institute	RACE	RACE-M	RACE-H
Human Ceiling Performance	Apr. 2017	CMU	*94.5*	*95.4*	*94.2*
Amazon Mechanical Turker	Apr. 2017	CMU	73.3	85.1	69.4
ALBERT (ensemble)	Sept. 26th 2019	Google Research & TTIC	**89.4**	**91.2**	**88.6**
ALBERT	Sept. 26th 2019	Google Research & TTIC	86.5	89.0	85.5
RoBERTa + MMM	Oct. 1st 2019	MIT & Amazon Alexa AI	85.0	89.1	83.3

RACE Leaderboard

DeepMind AI Flunks High School Math Test "They trained the machine on Arithmetic, Algebra, Comparisons, Calculus, Numbers, Measurement, Probability, and Manipulating Polynomials. It solved only some 35% of the 40 questions."

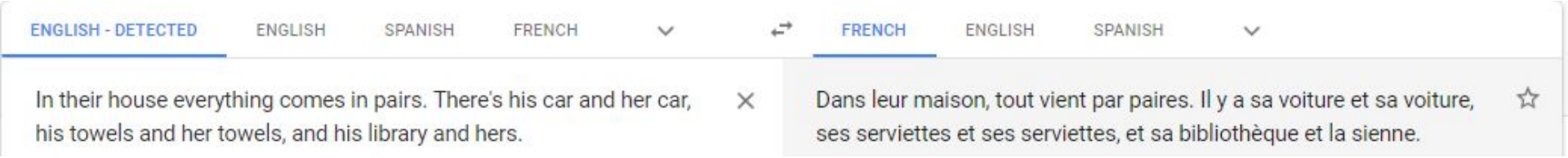
IBM's 'Project Debater' AI Lost to a Human - But Put Up Quite a Fight "I heard you hold the world record in debate competition wins against humans, but I suspect you've never debated a machine. Welcome to the future." During a twenty minute debate on a complex topic, Project Debater will digest massive texts, construct a well-structured speech, deliver it with clarity and purpose, and rebut its opponent.

Has BERT Been Cheating? Researchers Say it Exploits 'Spurious Statistical Cues'
Probing Neural Network Comprehension of Natural Language Arguments
"ARCT provides a fortuitous opportunity to see how stark the problem of exploiting spurious statistics can be. Due to our ability to eliminate the major source of these cues, we were able to show that BERT's maximum performance fell from just three points below the average untrained human baseline to essentially random. To answer our question in the introduction: BERT has learned nothing about argument comprehension. However, our investigations confirmed that BERT is indeed a very strong learner."

Language modeling: GPT-2 trained from text to choose words that maximize **P(next word | previous words) => generates fake stories => hopefully they need to learn to understand language to be able to do this**

• In theory, it would require complete understanding to obtain the best model, but the log-likelihood (perplexity) achieved by humans is not much better than that obtained by the best deep nets.

• Speech recognition and machine translation: **huge progress, but errors made by these systems show that they don't understand** what the sequences of words actually mean.



How well does BERT reason compared to LSTM? – let's see the feature importance

IMDB review positive or negative? :

"This was a good movie" :

BERT - positive (due to "good") , BiLSTM - positive (due to "this")

An adversarial attack:

"This this this ...this was a bad movie" :

BERT - negative, BiLSTM - positive

Let's try negation of positive:

"This was not a good movie" :

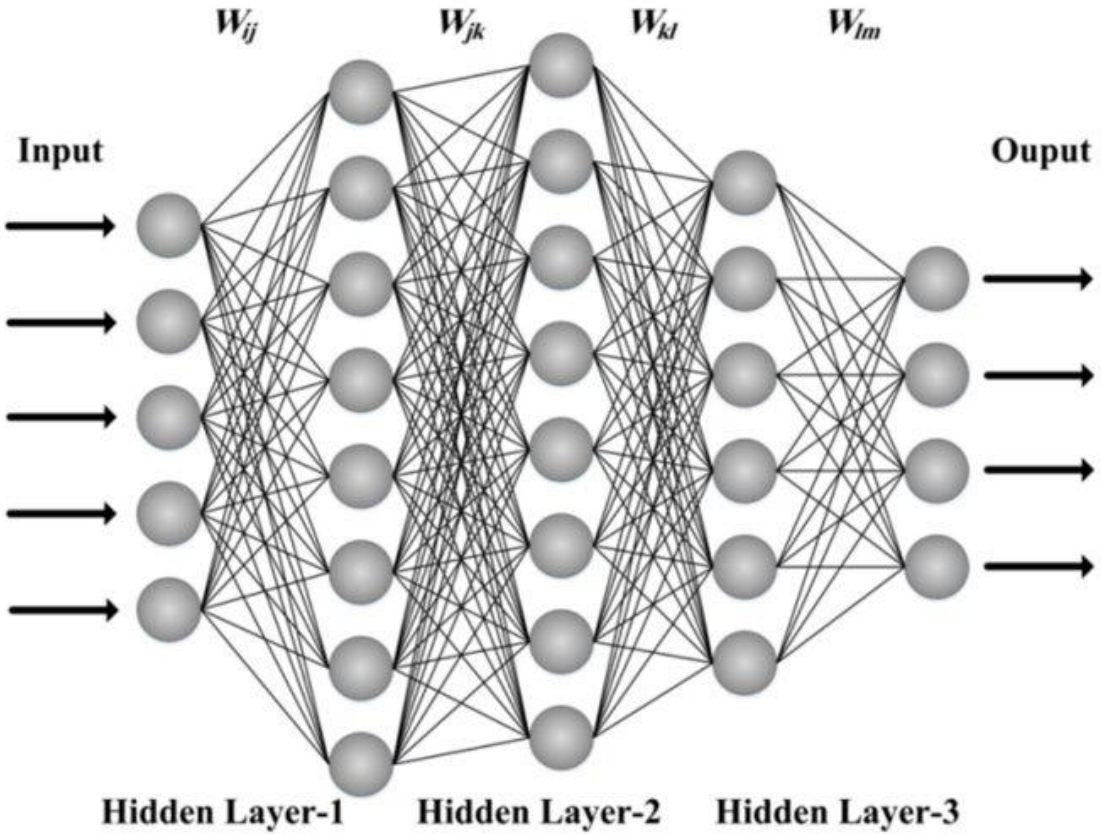
BERT - negative (due to "was not" more important than "good"), BiLSTM - positive

…

Let's try something more nuanced:

"This movie will not be a waste of your time": BiLSTM & BERT - negative (wrong both)

BERT is now part of Google Search, so let's understand how it reasons



Multilayer perceptron neural network

We have linear combination of linear combination of linear combinations...
Each neuron is a weighted sum of inputs.

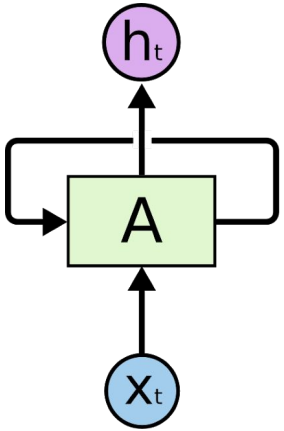
We have compound polynomial functions
=>
a linear function (but we need non-linear for learning any complex function
=>
add non-linear activation function)

Problems:
Input / output size is fixed and limited
Structure in data is no used

More layers => Vanishing gradient pb. - the error signal back-propagated can vanish due to multiplication of of many small numbers (almost zero derivatives of activation function)

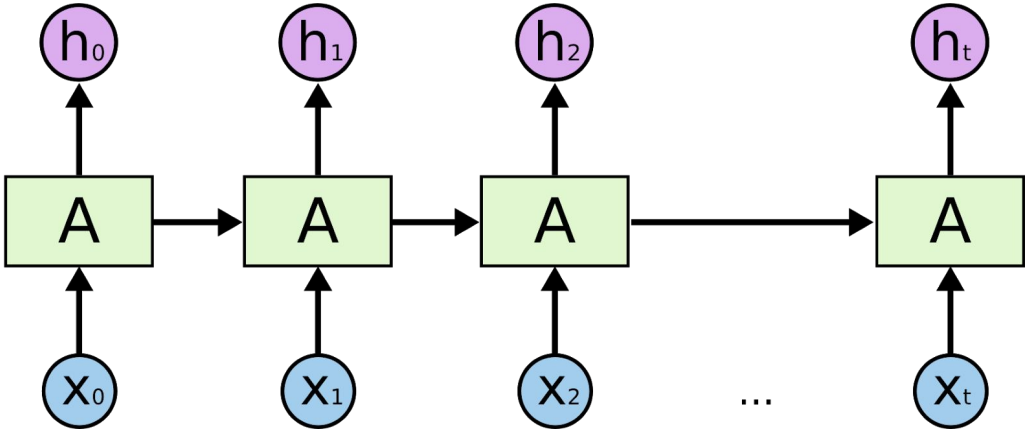
Traditional Recurrent Networks (RNN) - has a summarized memory of past inputs

With recurrent link:



=

Unrolled representation of RNN:



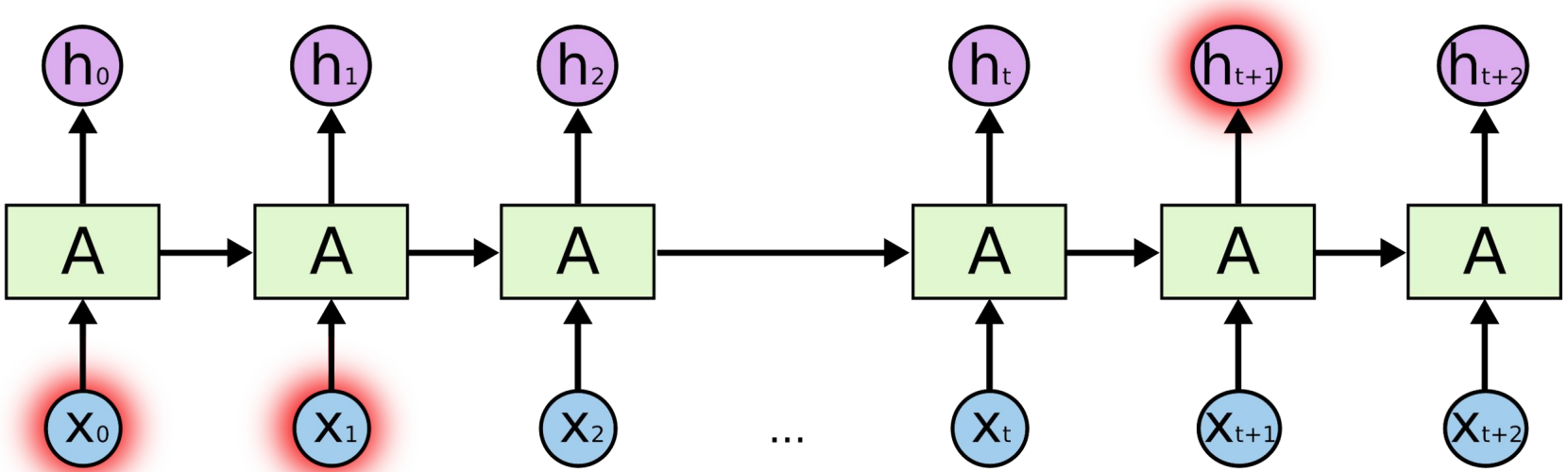
Current output depends on current input + previous output (we may say experience)
The unrolled version is equivalent , each cell parameters are shared.

Why RNN is good for NLP :

Sentences are variable length sequences.

We can infer meaning of one word in a sentence from the context. So we need a recurrent link for gathering the context info.

Why are the weights of RNN/LSTM networks shared across time?



"I grew up in France…… I speak fluent …"

Disadvantage:

A disadvantage of RNNs however is that the inherent **sequential aspect makes them difficult to parallelize**. Also, even with improvements from LSTMs or GRUs, **long-term dependencies are still hard to catch**.

Also - shallow architecture

How to deal with large input size = cause to explode the number of connections

Convolutional NN - images = **grid structure & local pixel correlations** =>
- keep large number of inputs but reduce number of connections , share parameters , down-sampling
- are parallelizable
- variable input size in theory at least

Recurrent NNs - have fixed number of inputs but allow to split large text in pieces and read piece by piece then output the result (use an internal memory to memorize a summary of all seen pieces) (problem - fixed memory size - 50 elements, difficulty to copy, slow, vanishing/exploding gradients)

LSTM (Long Short Term Memory) (or GRU) - add gates with layers of neurons for more intelligent control of memory access (basic internal attention = learn to filter out or add only relevant data in memory which is limited) (Problem: still short term memory, slow, can't stack too many layers, shallow architecture)

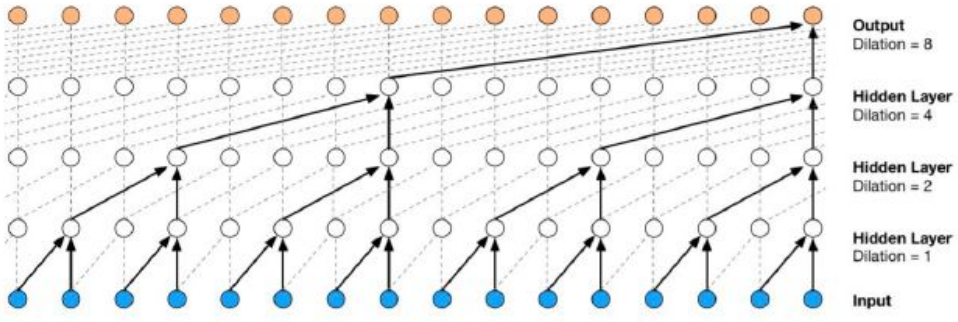
Other modernized **LSTMs:**

- **Relational Memory Core (RMC)** (200% better for Reinforcement Learning) (uses self attention = multi-head dot product attention) (Razvan Pascanu)
- **MAC (Memory, Attention and Composition) (CPU + LSTM + read/write Attention) (superhuman on VQA)**
- **AWD-LSTM (LSTM + special dropout for LSTM)**
- **Quasi-Recurrent Neural Network (LSTM + convolutions) = 16x faster, parallelizable LSTM**

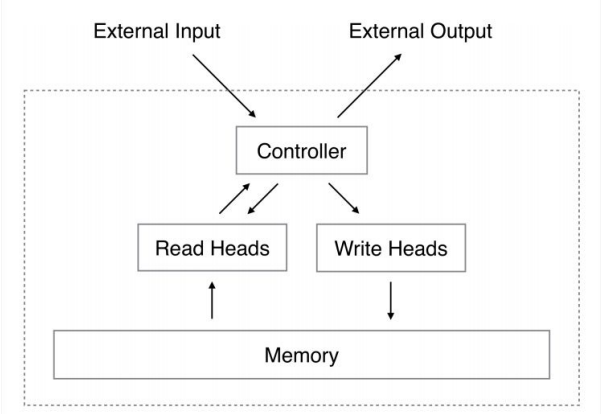
NTM LSTM architectures + Soft-Attention = revolutionary - precursor of transformer = 1 hop attention
- summarize and process only relevant data for the current word to translate

First attempts of attention networks:

ConvNets for NLP - easy to parallelize but still can't model long term dependencies - convolution is a local operation, need many layers - multi-hop attention
Examples : WaveNets / ByteNets



ConvNets



Neural Turing Machine

Memory Networks / Multi-Hop Attention = Differentiable Memory Computers , Precursors of Attention Networks
Attention to read / write in a classic memory
Neural Turing Machines introduced the idea of
Multiple-hop attention – stacking attention layers , Network keeps updated its memory to perform multi-step reasoning.

Transformer Architecture - first efficient version of attention networks!

Transformer - proposed improvements over memory networks:

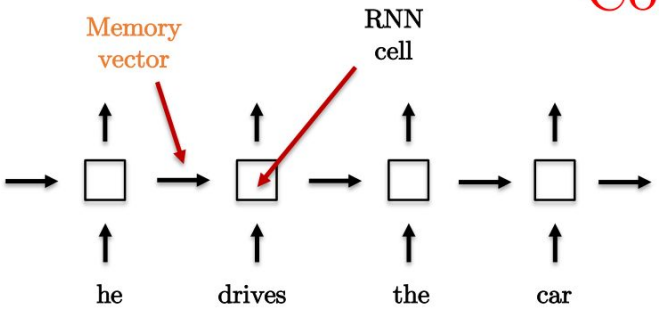
- Multiple hidden states (one per word)
- Multi-head attention (more learning capacity)
- Residual blocks (select specific attention layers, better backpropagation)

BERT - very powerful architecture for NLP - still fixed number of inputs (512 tokens) but large enough for a big paragraph

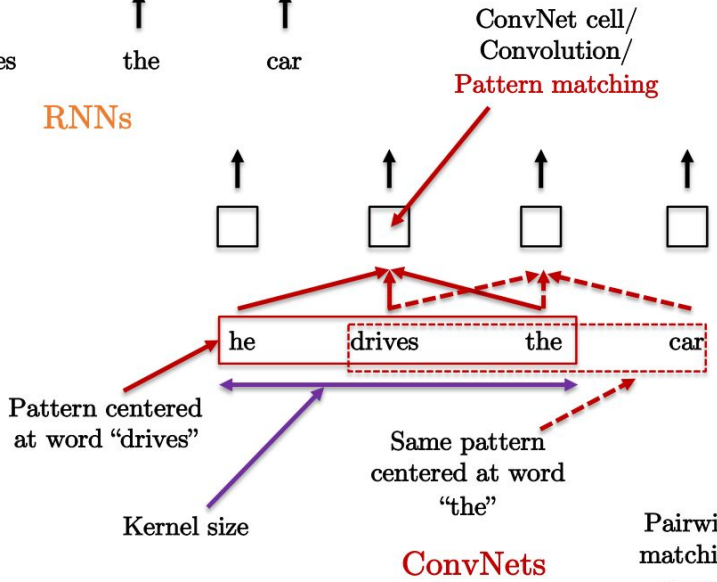
- deep architecture that transforms inputs into contextualized embeddings and process them in parallel,
- use self-attention to model dependencies between distant words

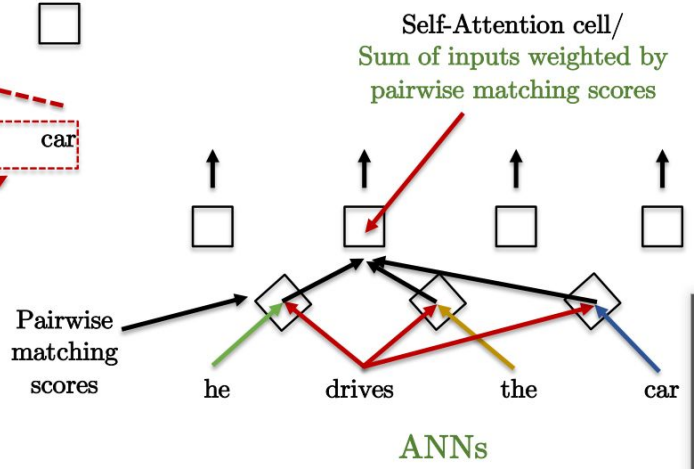
Transformer-XL - adds recurrent connection to BERT

Comparing Layers



All these layers can learn representation of sequences of different length/size and different context.





- RNN layer : $O(n.d^2)$
- ConvNet layer : $O(n.d^2.k)$
- Transformer layer : $O(n^2.d)$

Seems scary !

with n : sequence length, d : hidden feature size, k : kernel size

- Attention networks have actually less parameters as long as $d \geq n$!
  - Example 1 : n=100, d=1000, k=3

    RNN: $O(10^8)$, ConvNet: $O(3.10^8)$, Transformer: $O(10^7)$
  - Example 2 : n=1000, d=1000, k=3

    RNN: $O(10^9)$, ConvNet: $O(3.10^9)$, Transformer: $O(10^9)$

[IASI AI] 2018 - NLP's ImageNet moment - a huge NN learns "language"/"vision" => fast learn a specific task

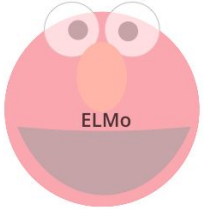
Training objective:
Machine Translation
(MT)

3 layer AWD-LSTM (ASGD Weight-Dropped LSTM)
First model with pretraining + fine-tuning, good fo classification only (Multi-Fit uses QRNN - Quasi-Recurrent Neural Network)

Bidirectional pretraining + fine-tuning
Training Objective:
Masked Language Modeling (MLM)

GPT-2 model:
Unidirectional - left to right

Training objective:
Language Modeling (LM)



2 layer LSTM based
Shallow bidirectional + extract embeddings (deep contextualized word representations)
Training objective: multiple tasks (sentiment classification etc)

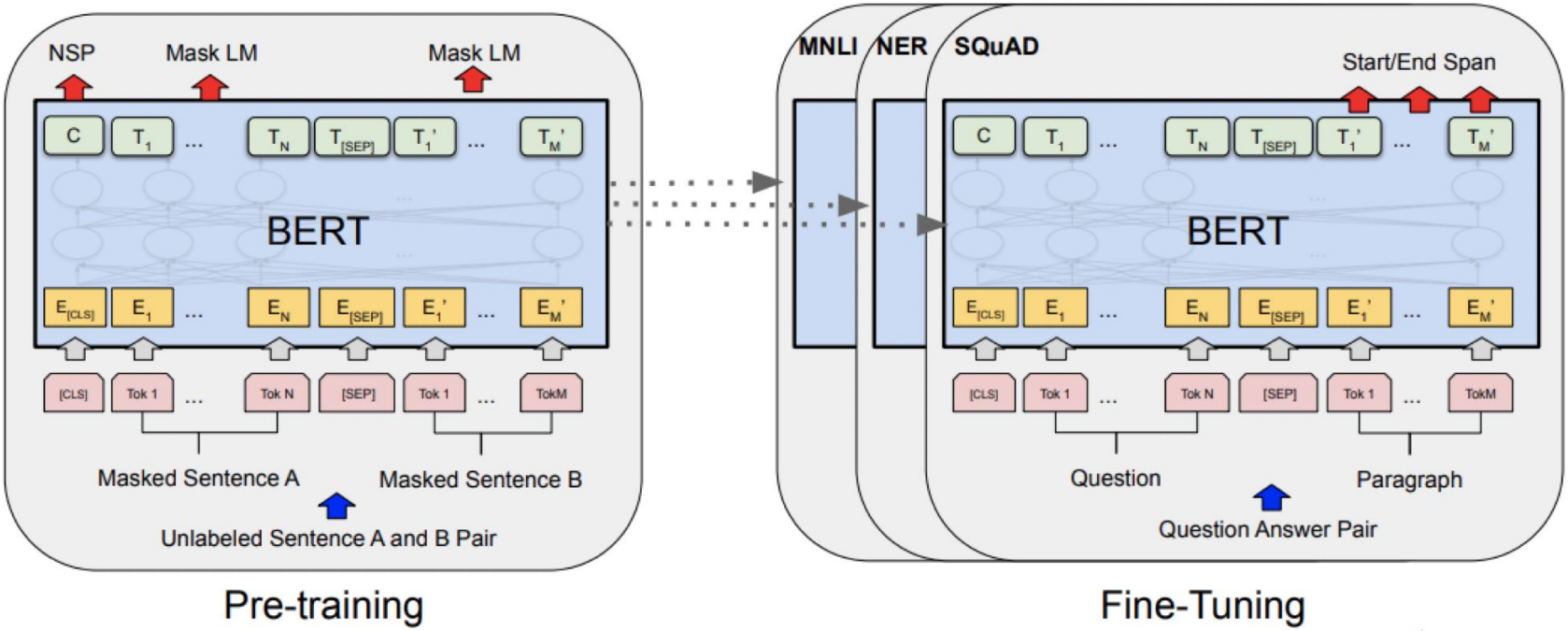
ImageNet moment = Use language modeling for (unsupervised) pre-training + fine-tuning (transfer learning)

Embeddings from Language Models (ELMo), Universal Language Model Fine-tuning (ULMFiT), and the OpenAI Transformer have empirically demonstrated how language modeling can be used for pretraining.
...
Empirical and theoretical results in multi-task learning (Caruana, 1997; Baxter, 2000) indicate that a bias that is learned on sufficiently many tasks is likely to generalize to unseen tasks drawn from the same environment.

The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning), NLP's ImageNet moment has arrived

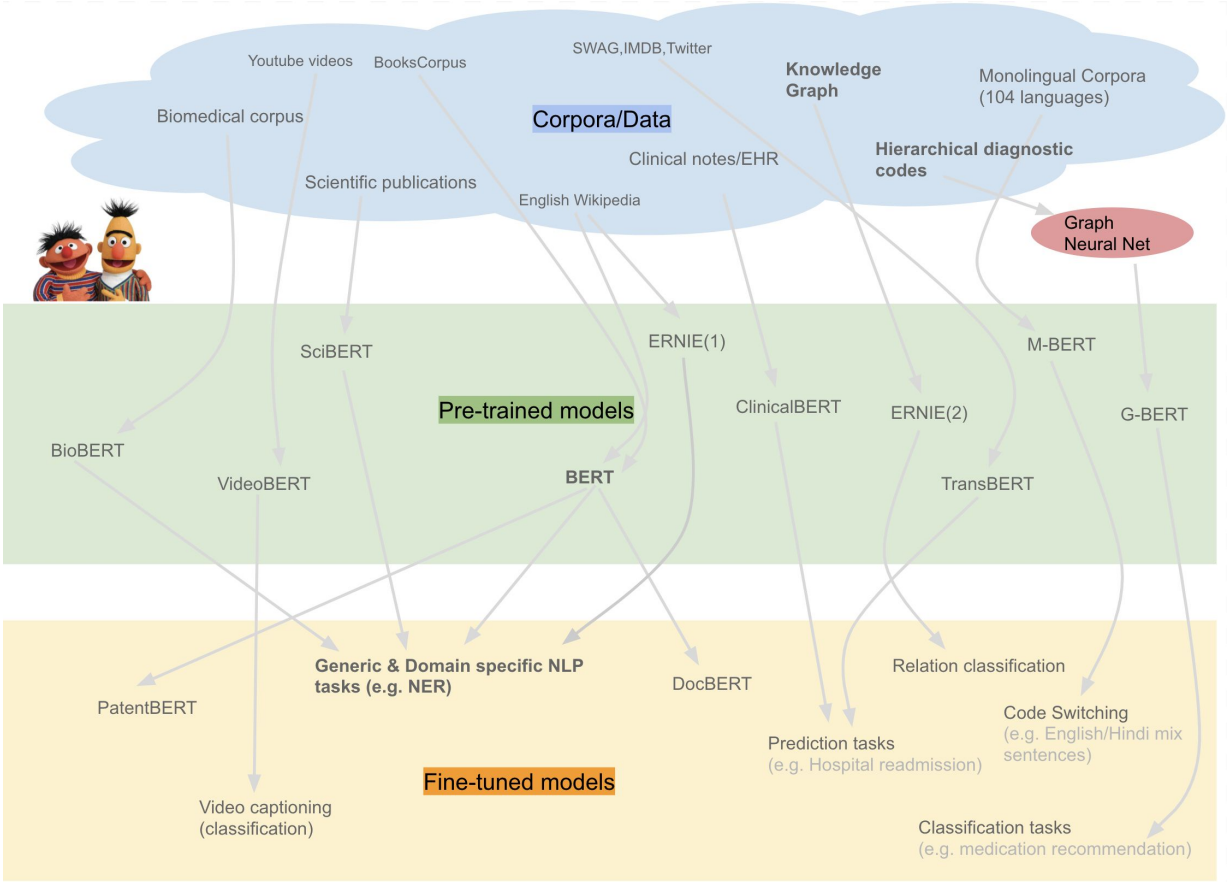
BERT - unsupervised pre-training + fine-tuning on different tasks

First pre-train the entire model on a data-rich task using unsupervised learning on unlabeled data. Ideally, this pre-training causes the model to develop general-purpose abilities and knowledge that can then be “transferred” to downstream tasks by fine-tuning (transfer learning + supervised learning).



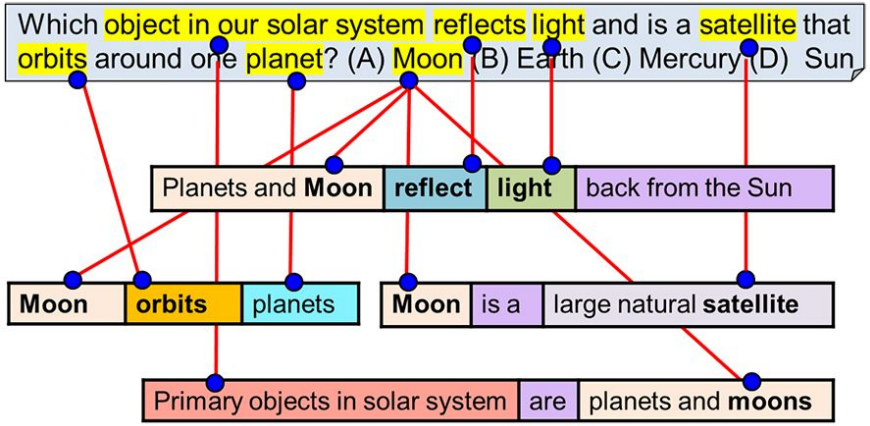
NAACL ’19 Notes: Practical Insights for Natural Language Processing Applications

BERT Zoo - pre-trained and fine-tuned on specific domain datasets



A review of BERT based models

Aristo system passes eighth-grade science tests - uses BERT based model

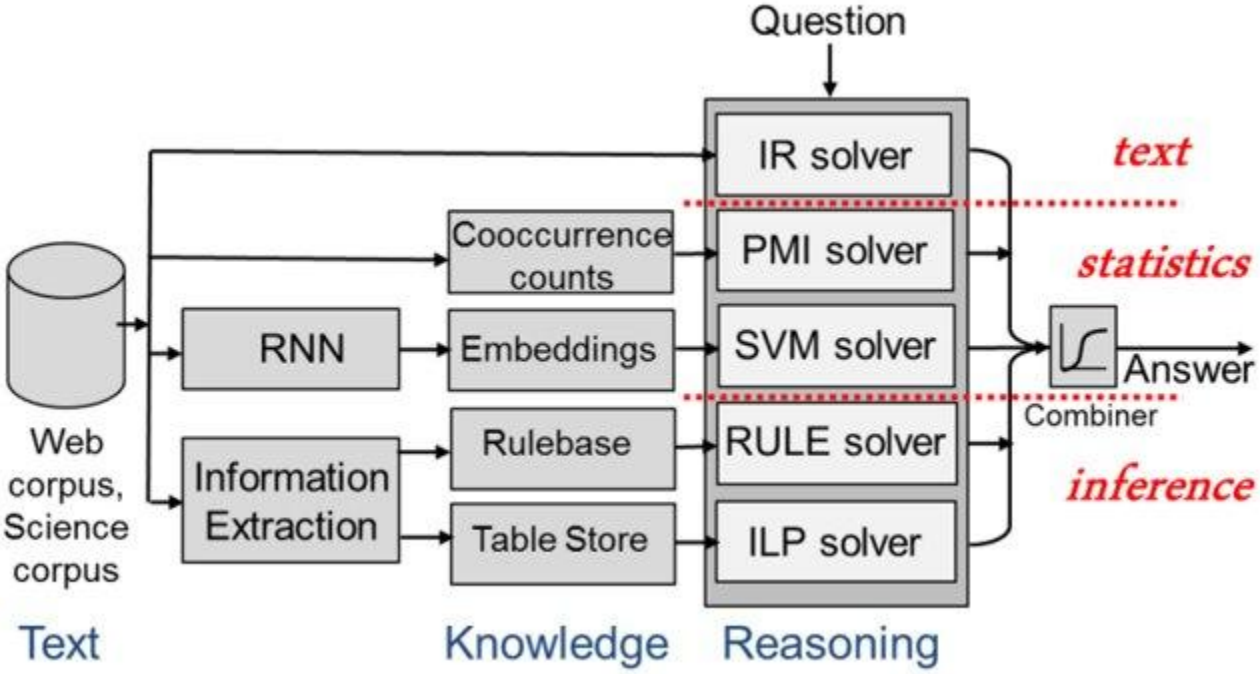


The Aristo Project aims to build systems that demonstrate a deep understanding of the world, integrating technologies for reading, learning, reasoning, and explanation.

Allen Institute's Aristo AI system finally passes an eighth-grade science test (ARISTO – Live Demo)
A Breakthrough for A.I. Technology: Passing an 8th-Grade Science Test

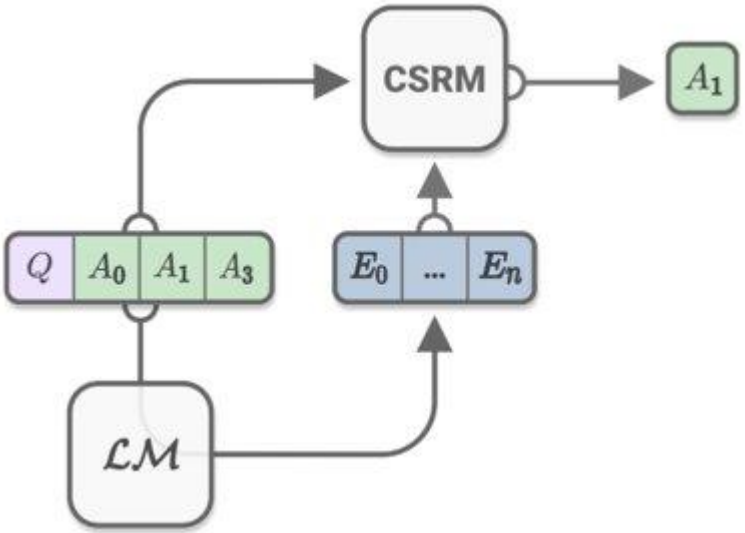
ARISTO - An intelligent system that reads, learns, and reasons about science.

Aristo uses five solvers, each using different types of knowledge, to answer multiple choice questions
In 2019 they added AristoRoBERTa in the mix.
Now is able to pass 8th grade science tests (previously 4th grade)
Issue : still have problems with complex reasoning and understanding.



ARISTO – Live Demo ,
AI Can Now Pass School Tests but Still Falls Short on the Turing Test

Implementations:
LM = GPT-2 (generates explanations)
CSRM = BERT (answers questions based on question , answers and generated explanation)



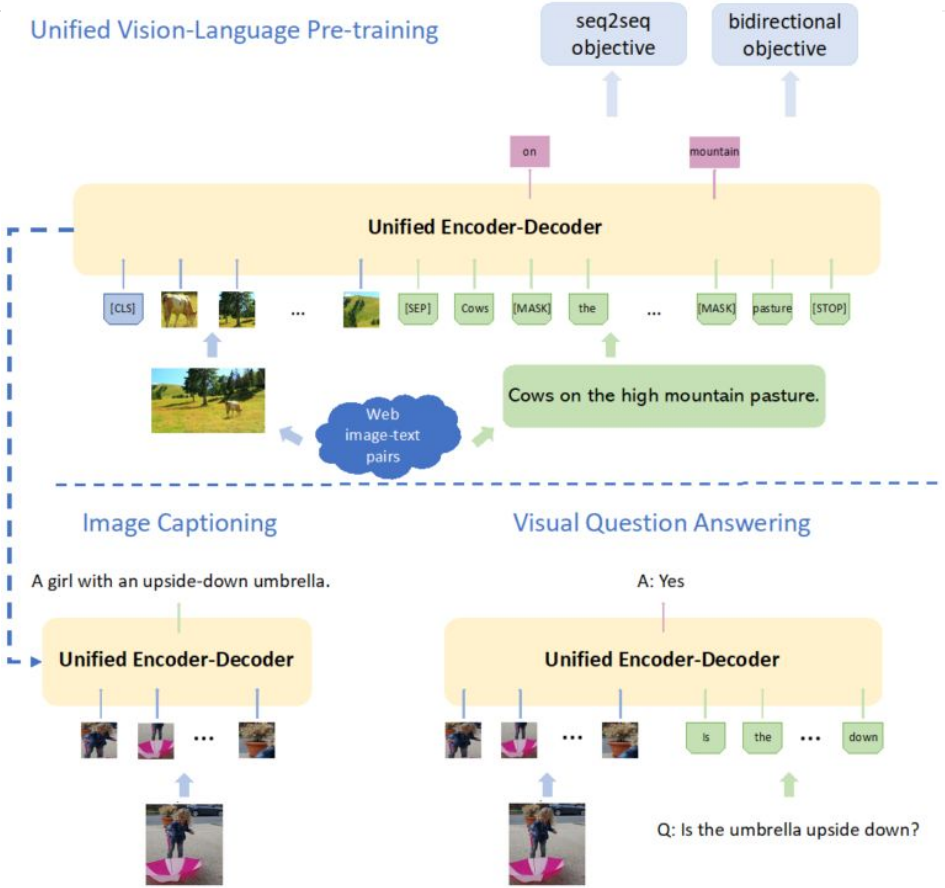
Method	Accuracy (%)
RC (Talmor et al., 2019)	47.7
GPT (Talmor et al., 2019)	54.8
CoS-E-open-ended	60.2
CAGE-reasoning	**64.7**
Human (Talmor et al., 2019)	95.3

A commonsense reasoning model (CSRM) uses the LM explanations to make a prediction.

Leveraging Language Models for Commonsense Reasoning in Neural Networks (from Salesforce)

Question:	What could people do that involves talking?
Choices:	**confession**, carnival, state park
CoS-E data:	confession is the only vocal action.
CAGE model:	people talk to each other
Question:	A child wants to play, what would they likely want?
Choices:	**play tag**, breathe, fall down
CoS-E data:	A child to play tag
CAGE model:	Children want to play tag, and they want to play tag with their friends.
Question:	They were getting ready for a really long hike, he put the food in his what?
Choices:	recycling center, house, **backpack**
CoS-E data:	Backpacks are used on hikes
CAGE model:	a backpack is a place to store food and supplies.
Question:	You can do knitting to get the feeling of what?
Choices:	**relaxation**, your, arthritis
CoS-E data :	Your are focusing on a repetitive task.
CAGE model:	knitting is the only thing that is relaxing.

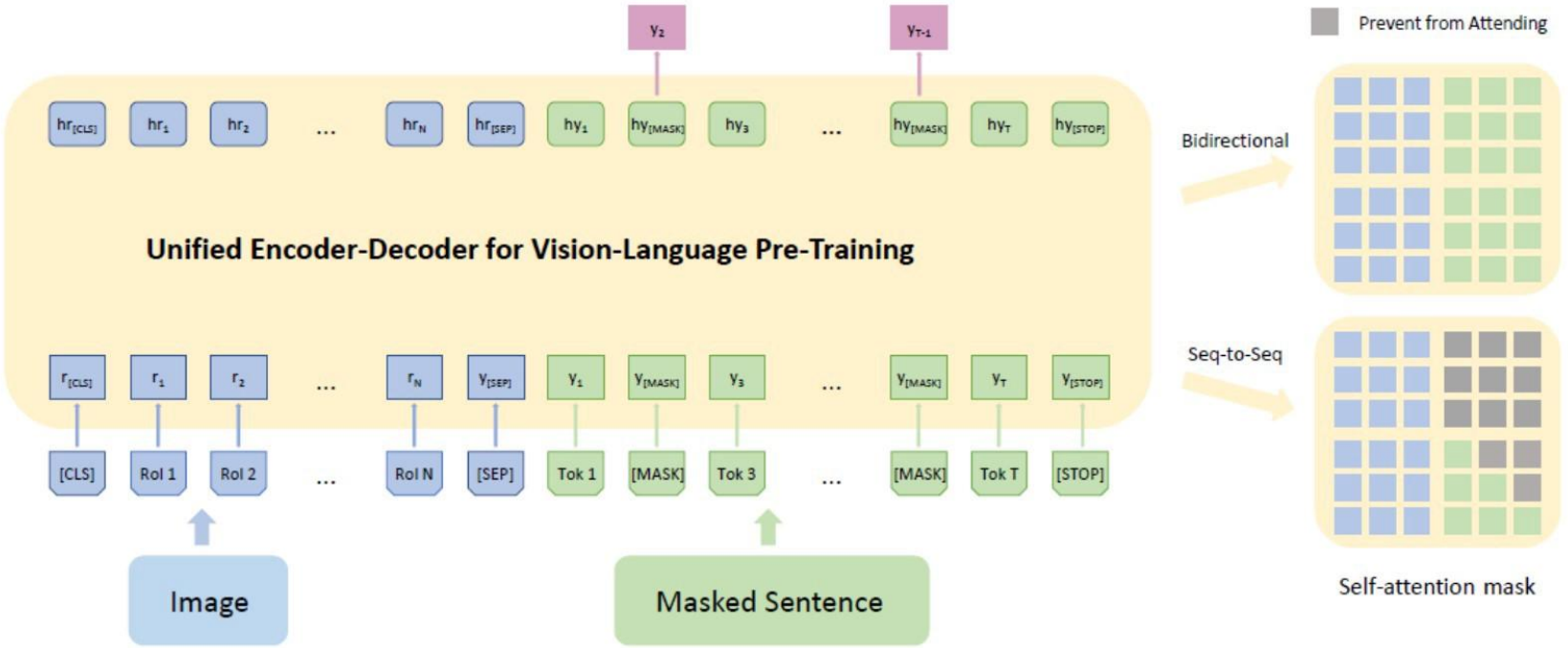
Microsoft Unified VLP (Vision-Language Pre-Training) for Image Captioning and VQA



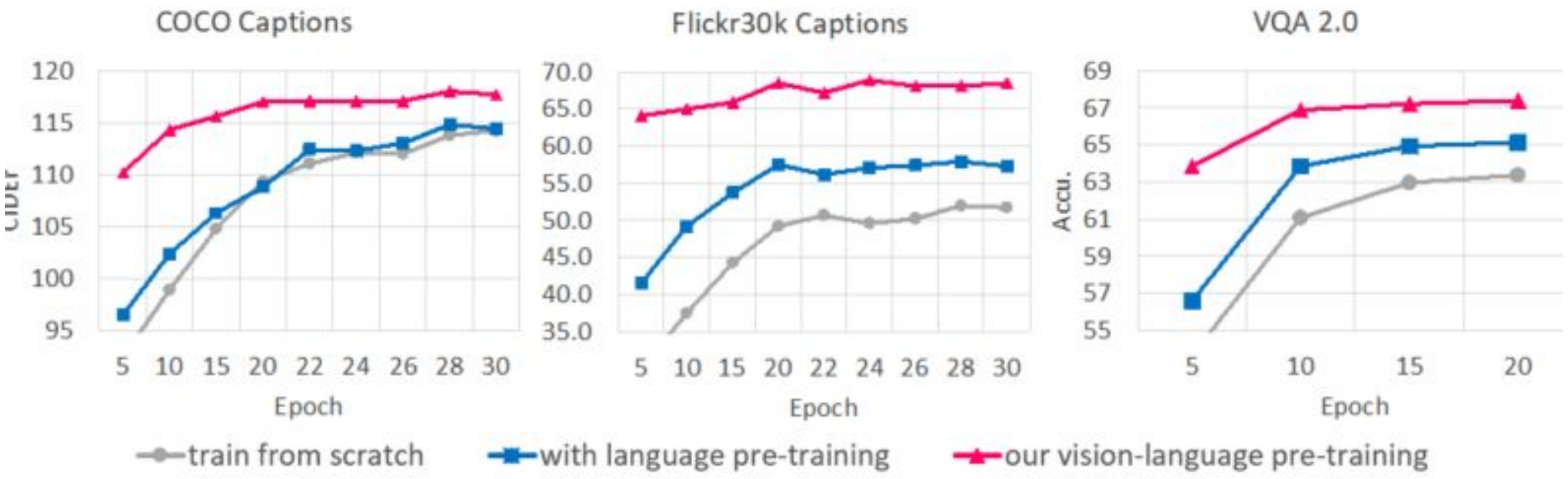
- Pre-training with pairs of image and text (images split in ROI)
- Fine-tuning on Image Captioning and VQA

Unified Vision-Language Pre-Training for Image Captioning and VQA,
This Microsoft Neural Network can Answer Questions About Scenic Images with Minimum Training

Microsoft Unified VLP - Architecture - BERT based



Microsoft Unified VLP - Qualitative Results - vision + language is better



Microsoft Unified VLP - Qualitative Results

	**GT sentences:** People in matching shirts standing under umbrellas in the sun People in the same colorful shirts have umbrellas. A large group of people with an umbrella outside. A group of men standing next to a lot of umbrellas A group of people that are under one umbrella	**Unified VLP** (159.8): A group of people standing under umbrellas in the rain. **Init from UniLM** (59.0): A group of people standing around each other. **Init from BERT** (59.0): A group of people standing around each other.	**Question**: Are they dressed the same? **Correct answer**: Yes **Unified VLP**: Yes **Init from UniLM**: No **Init from BERT**: No
	**GT sentences:** A man standing in front of a blue wall A man talks on a phone in a room with blue wallpaper A man holding a cell phone standing in front of blue wallpaper with designs and a large wall vent A man on a cell phone by a bright blue wall A man holding a phone to his ear	**Unified VLP** (180.6): A man talking on a cell phone in front of a blue wall. **Init from UniLM** (126.9): A man talking on a cell phone while standing next to a blue wall. **Init from BERT** (59.6): A man talking on a cell phone while wearing a gray shirt.	**Question**: Is the man taking his own picture? **Correct answer**: No **Unified VLP**: No **Init from UniLM**: Yes **Init from BERT**: Yes

Embeddings:

Visualise embeddings: Embedding Projector

ElMo embeddings: HMTL for NLP

LSTM based :

(BiDAF (trained on SQuAD), ELMo-BiDAF (trained on SQuAD), NAQANet (trained on DROP)):
AllenNLP Reading Comprehension

Ranker DrQA (facebook) + R-NET (microsoft)
DeepPavlov Open Domain Question Answering (ODQA)

BERT based:

GPT-2:

GPT-2 model - AllenAI Demo , Giant Language model Test Room (A tool to detect automatically AI generated text) , Write With Transformer - Get a modern neural network to auto-complete your thoughts.
Talk to Transformer

BERT:

Question And Answer Demo Using BERT NLP , Fortech chat bot (RASA + cdQA)

Multiple models used based on different architectures: ARISTO – Live Demo

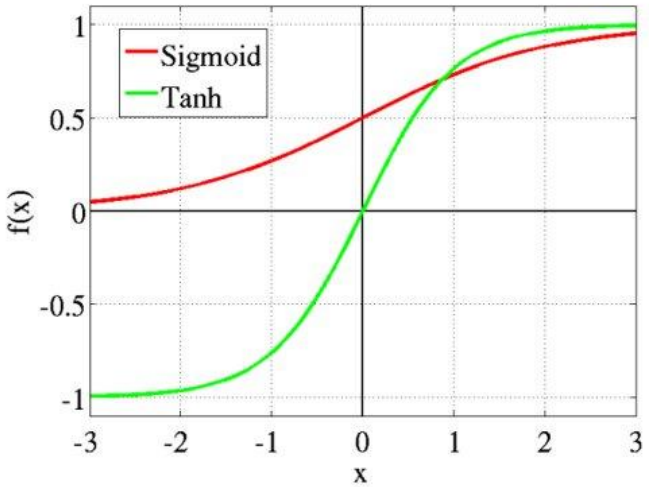
Beware! From here starts more technical slides!

Activation functions are used to

- add non-linearity to NN for better expressivity (NN to be able to approximate any function)
- perform some conversion or normalization

Sigmoid :
converts a real number (logit) to probability (0,1)

$$S(z) = \frac{1}{1 + e^{-z}}$$



Tanh :
squashes a real number into interval (-1, 1)

$$tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

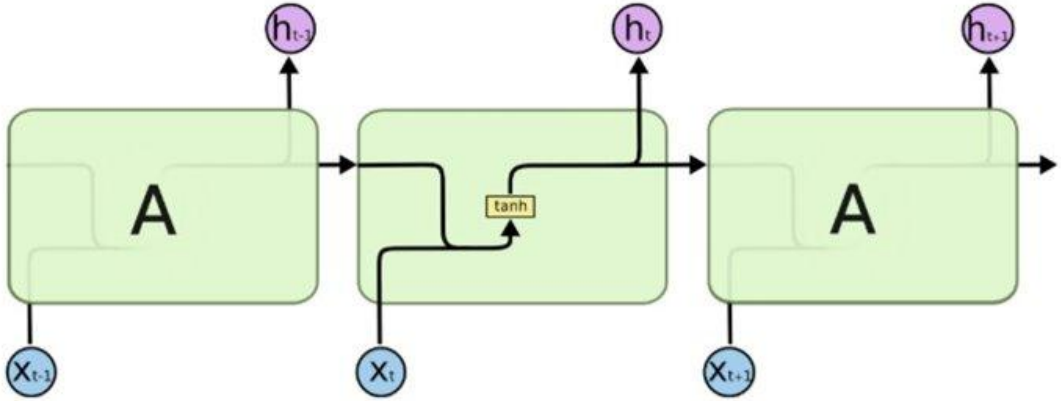
Activation Functions in Neural Network, ML Cheatsheet – Activation Functions

Traditional Recurrent Networks - Impractical : no long-range dependencies

RNN Training algorithm: **backpropagation through time**

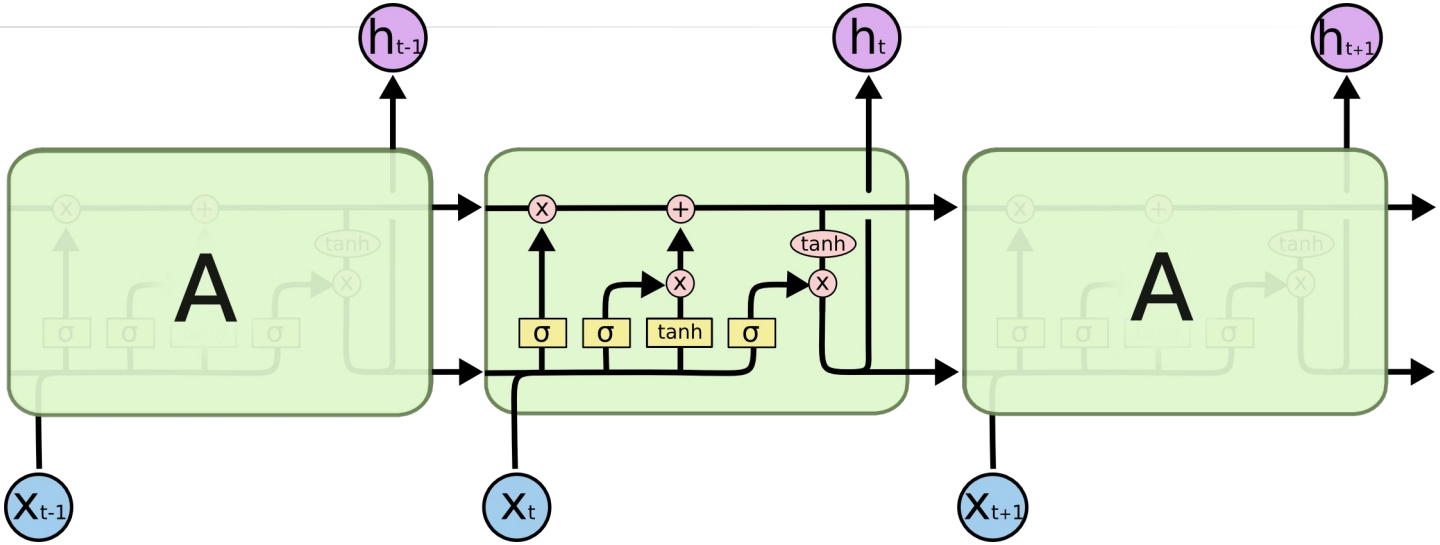
RNN Big Problem: The vanishing gradient problem is when the **gradient shrinks as it back propagates through time**. If a gradient value becomes extremely small, it doesn't contribute too much to learning.

In a standard non-gated recurrent neural network, the state at time step t is a linear projection of the state at time step t–1, followed by a nonlinearity. This kind of "vanilla" RNN can have **difficulty with long-range temporal dependencies** because it has to learn a very precise mapping just to **copy information unchanged** from one time state to the next.



The repeating module in a standard RNN contains a single layer.

LSTM (Long Short-Term Memory networks) for sequence modeling



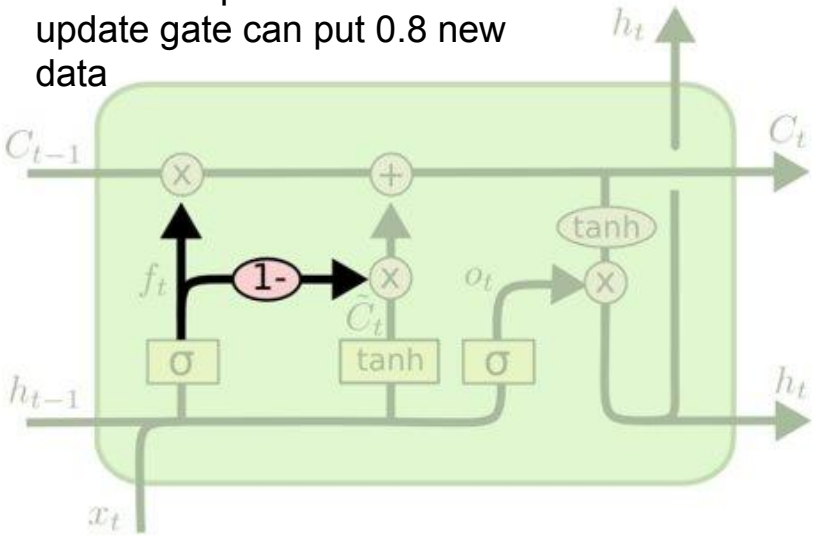
The repeating module of LSTM contains four interacting layers

- Popular type of RNN network which solve vanishing and exploding gradient problem.
- Works by copying its internal state (called the “cell state”) from each time step to the next. Rather than having to learn how to remember, it remembers by default.
- A memory accessed and modified by 3 gates (update, forget, output), each gate controlled by a layer of neurons
- Good for time - series where context counts and we have long-term dependencies between elements,
- Can deal with noisy time-series , it can bypass non-relevant input steps and thus remember for longer time steps
- Used for sequence modeling and time-series prediction (NLP, AI voice assistants, music creation, basic reasoning, stock market forecasting, programmer)

Illustrated Guide to LSTM’s and GRU’s, Understanding LSTM Networks, LSTM-nuggets for practical applications, The Magic of LSTM , LSTM Original paper, Introduction to LSTM Units While Playing Jazz

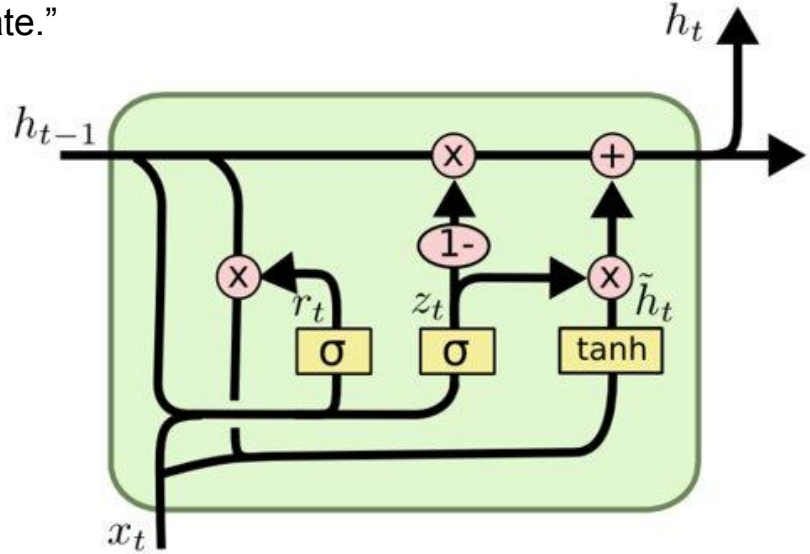
Coupled forget and input gates:

Forget gate multiplies by 0.2 - means 0.8 percent data lost so update gate can put 0.8 new data



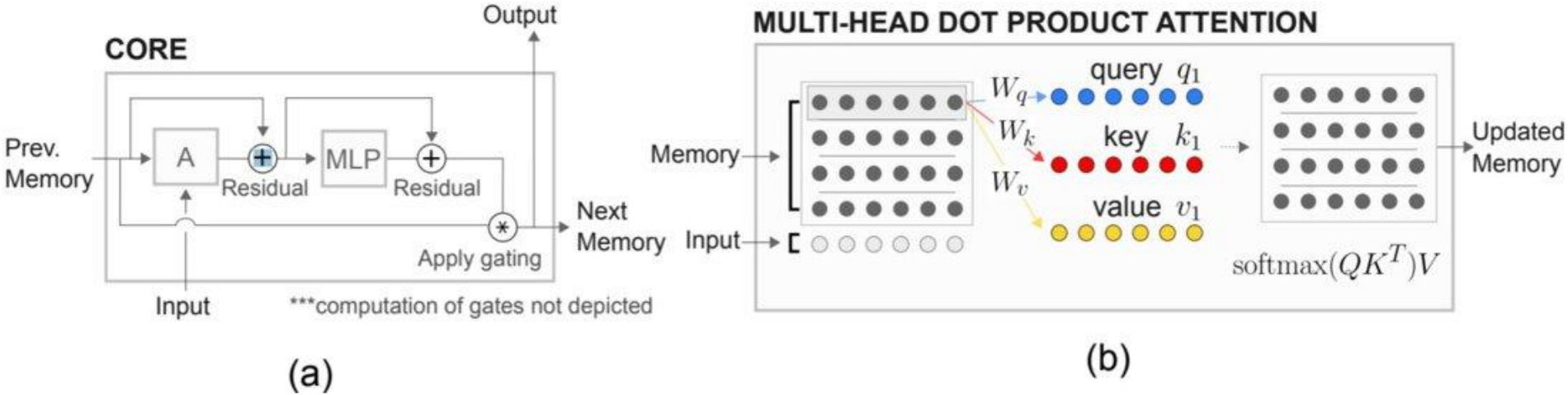
Gated Recurrent Unit (GRU)

- merges the cell state and hidden state
- combines the forget and input gates into a single “update gate.”



Training RNNs as Fast as CNNs: **SRU** is a recurrent unit that can run 10x faster than cuDNN LSTM

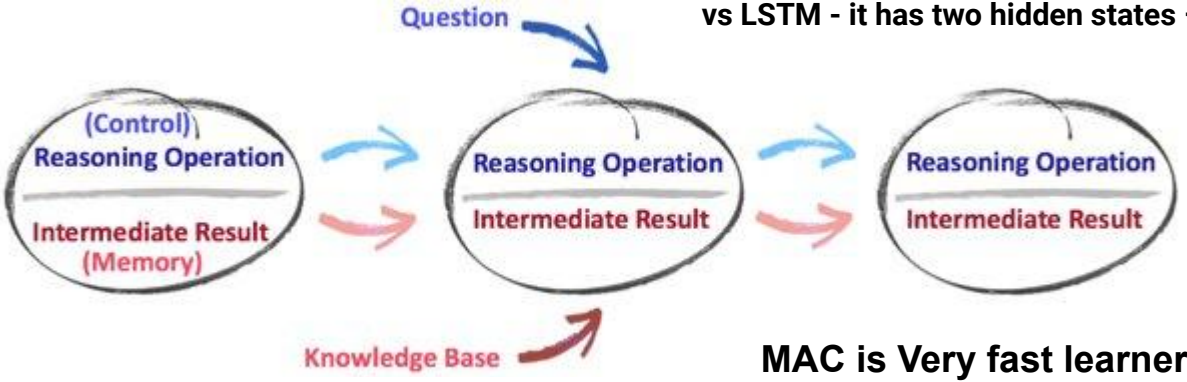
LSTMs pack all information into a common hidden memory vector, potentially making compartmentalization and relational reasoning more difficult => let's use a matrix instead, each row can be a separate memory and use attention between memories



We test the RMC on a suite of tasks that may profit from more capable relational reasoning across sequential information, and show **large gains in RL domains** (BoxWorld & Mini PacMan), **program evaluation**, and language modeling, achieving state-of-the-art results …
...In RL for games **nearly doubled the performance of an LSTM**

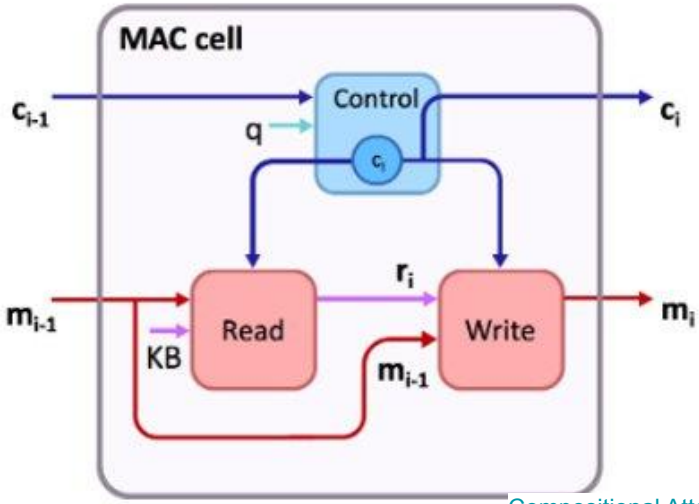
Relational recurrent neural networks (Razvan Pascanu, ...) (NIPS/NeurIPS 2018) (*Revolutionary*) (*DeepMind*), Paper

MAC (Memory, Attention and Composition) cell - superhuman for VQA = CPU + Attention LSTM

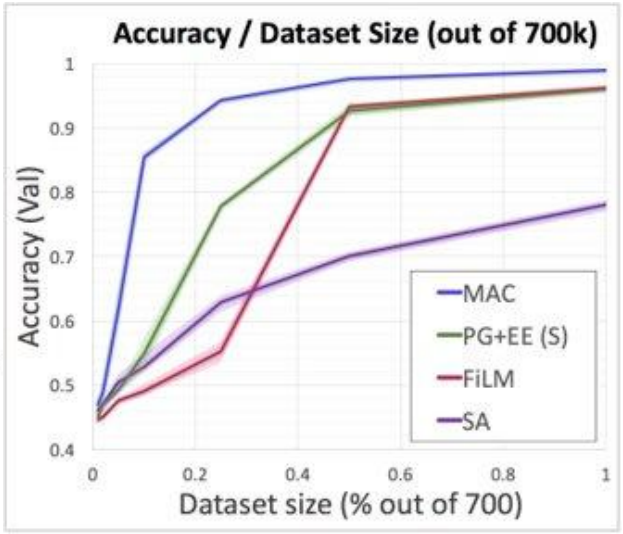


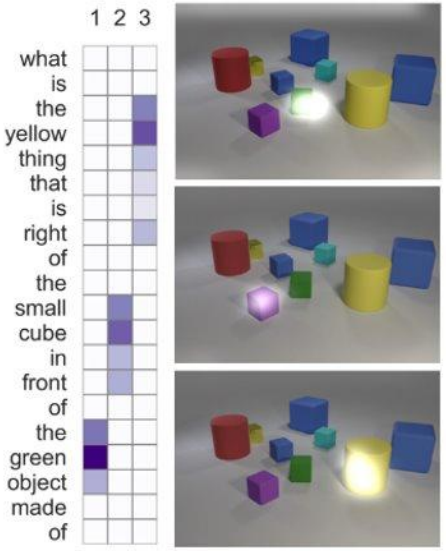
vs LSTM - it has two hidden states – control and memory, rather than just one

- a new state-of-the-art 98.9% accuracy, halving the error rate of the previous best model. More importantly, we show that the model is computationally-efficient and data-efficient, in particular requiring 5x less data than existing models to achieve strong results



MAC is Very fast learner:





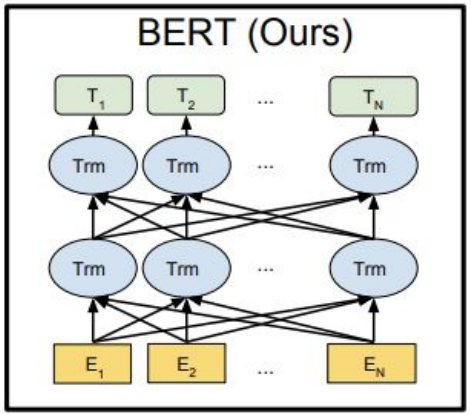
Compositional Attention Networks for Machine Reasoning, Code, Paper, Building Neural Network Models That Can Reason

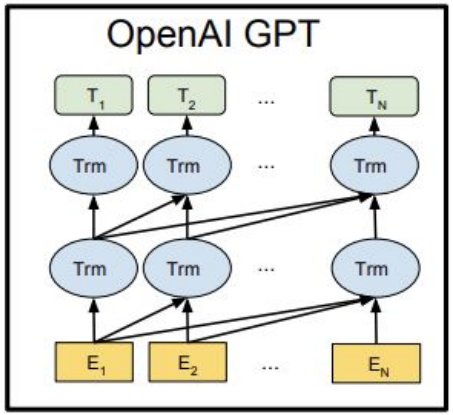
BERT vs GPT vs ELMo for predicting next word

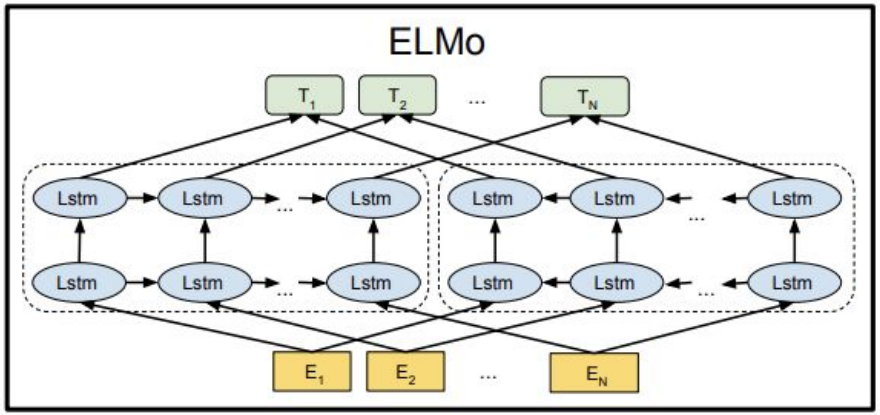
Bidirectional
Oups! Does it cheat?

Unidirectional - sees only left words

Bidirectional but... - a word sees other words via long paths







BERT is deeply bidirectional, OpenAI GPT is unidirectional, and ELMo is shallowly bidirectional.

BERT has 2 different training objectives:

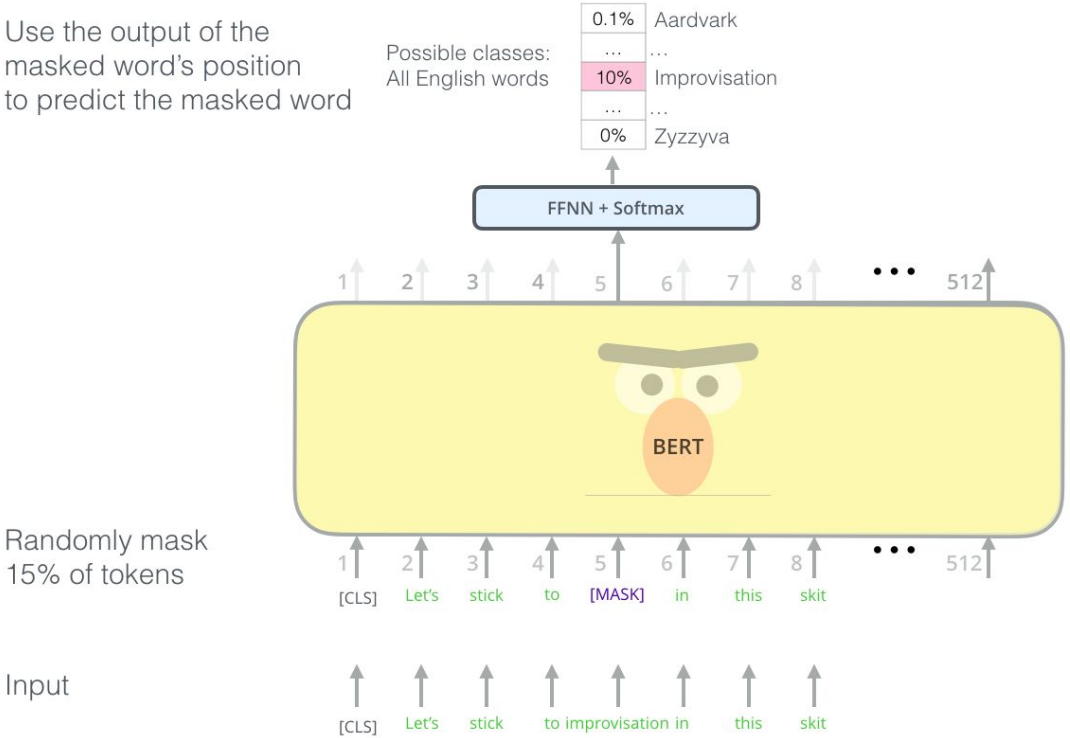
- Masked language modeling: mask 15% of tokens and predict them based on the whole text. (no more cheating)
- Is next sentence prediction

BERT Advantages - long-term dependencies , parallelizable (In contrast to other approaches, it discovers the context concurrent rather than directionally)
Issue: large but limited input size

Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing

BERT - Pre-training phase - masked language modeling - self-supervised training

Input: The man went to the $[MASK]_1$. He bought a $[MASK]_2$ of milk .
Labels: $[MASK]_1$ = store; $[MASK]_2$ = gallon



Note:
The masked words are not always replaced with the masked token – [MASK] because then the masked tokens would never be seen before fine-tuning.

Therefore, 15% of the tokens are chosen at random and –

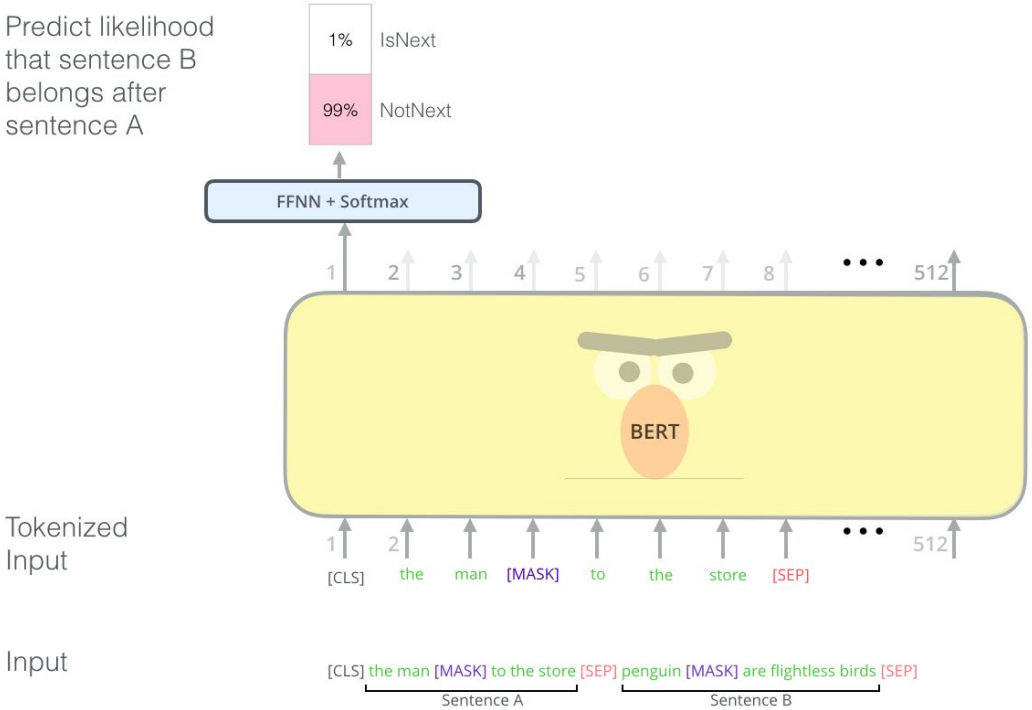
- 80% of the time tokens are actually replaced with the token [MASK].
- 10% of the time tokens are replaced with a random token.
- 10% of the time tokens are left unchanged.

The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

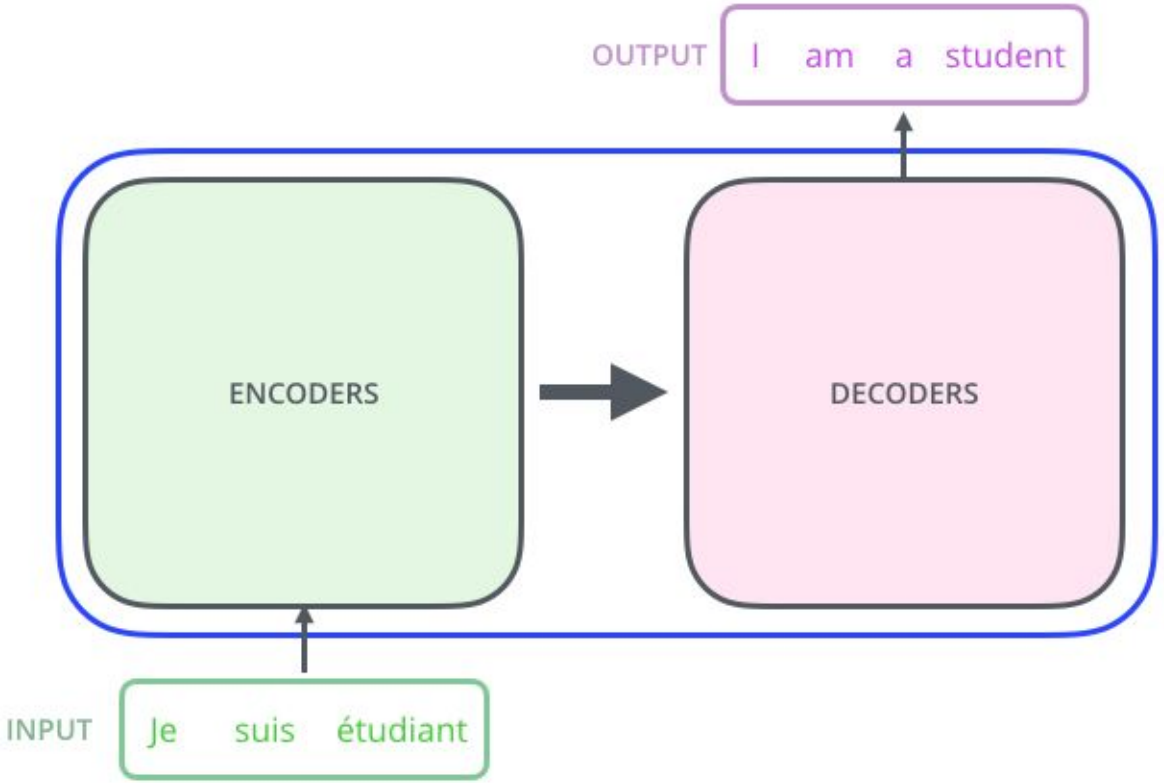
BERT - Pre-training phase - 2 sentence prediction - self-supervised training

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

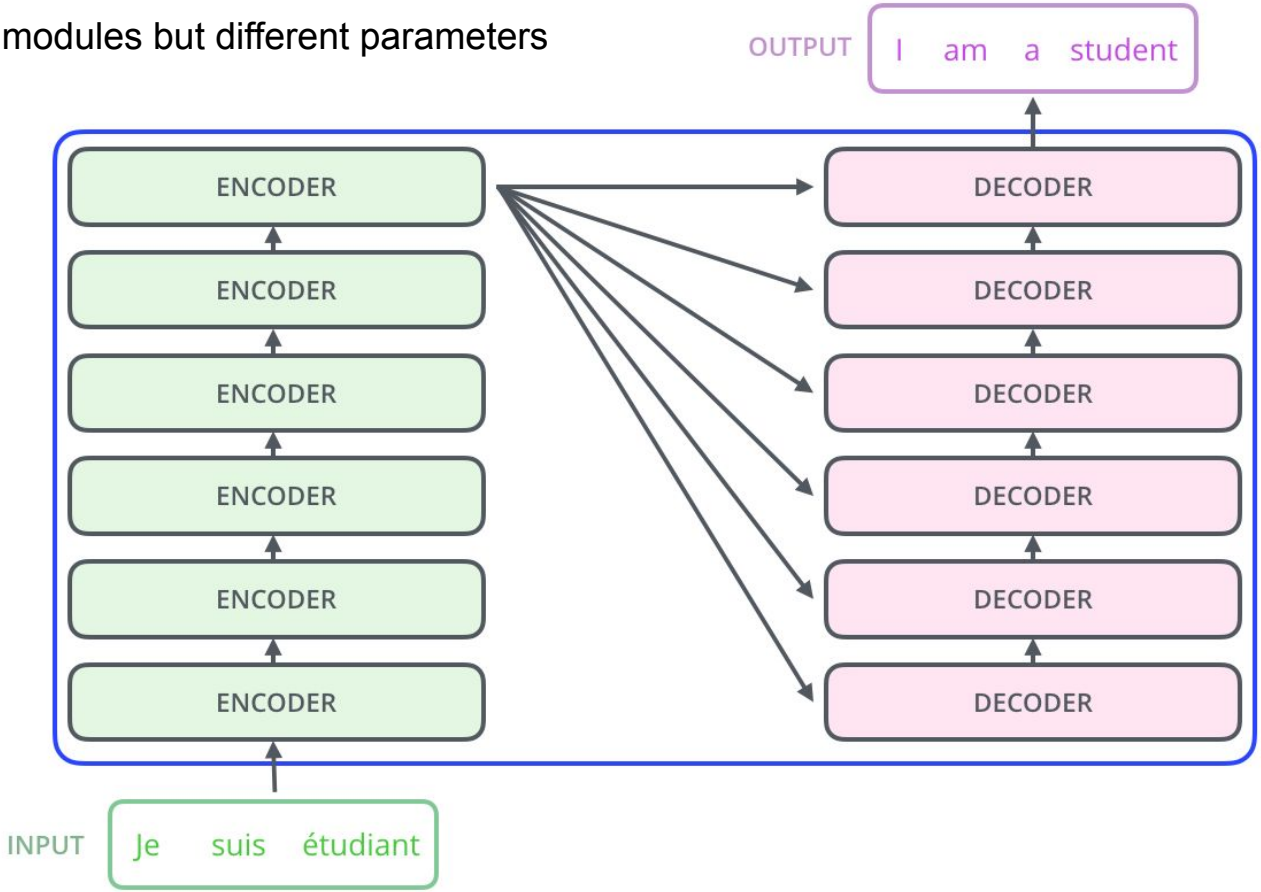
Sentence A = The man went to the store.
Sentence B = Penguins are flightless.
Label = NotNextSentence



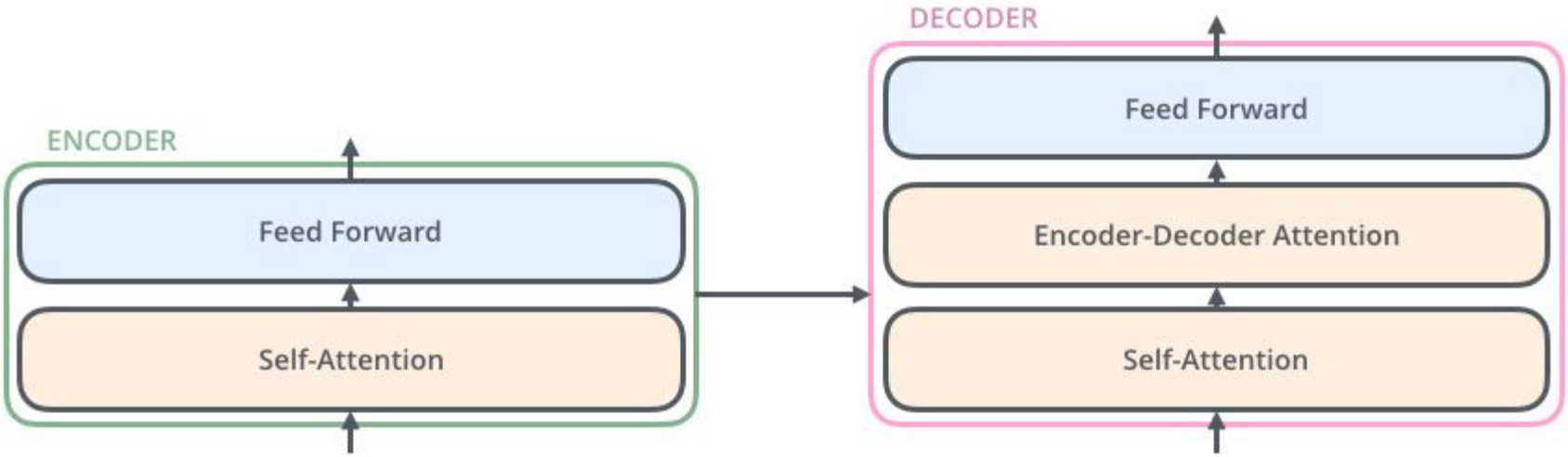
Transformer architecture for translation - encoder-decoder



Note : Identical modules but different parameters

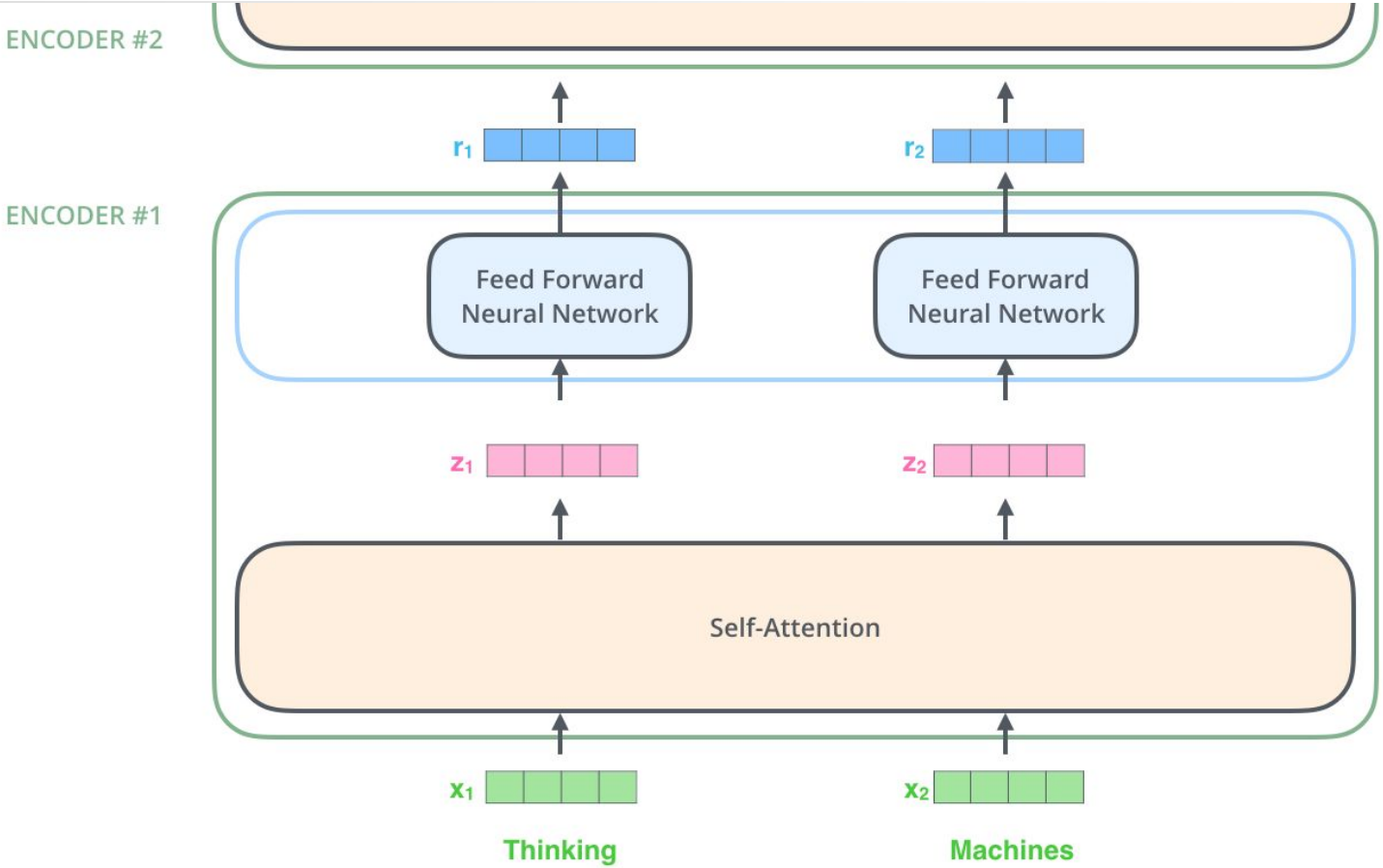


Encoder & Decoder sub-components



The Illustrated Transformer

Encoder - more detailed



- Problem: different meaning, same embedding vector

If you drive down the road and follow the river bank, you should find the Bank of America on your right.

(0.2, 0.5, 0.13, ...)

- Solution: modify embedding for each position taking into an account the nearby relevant context

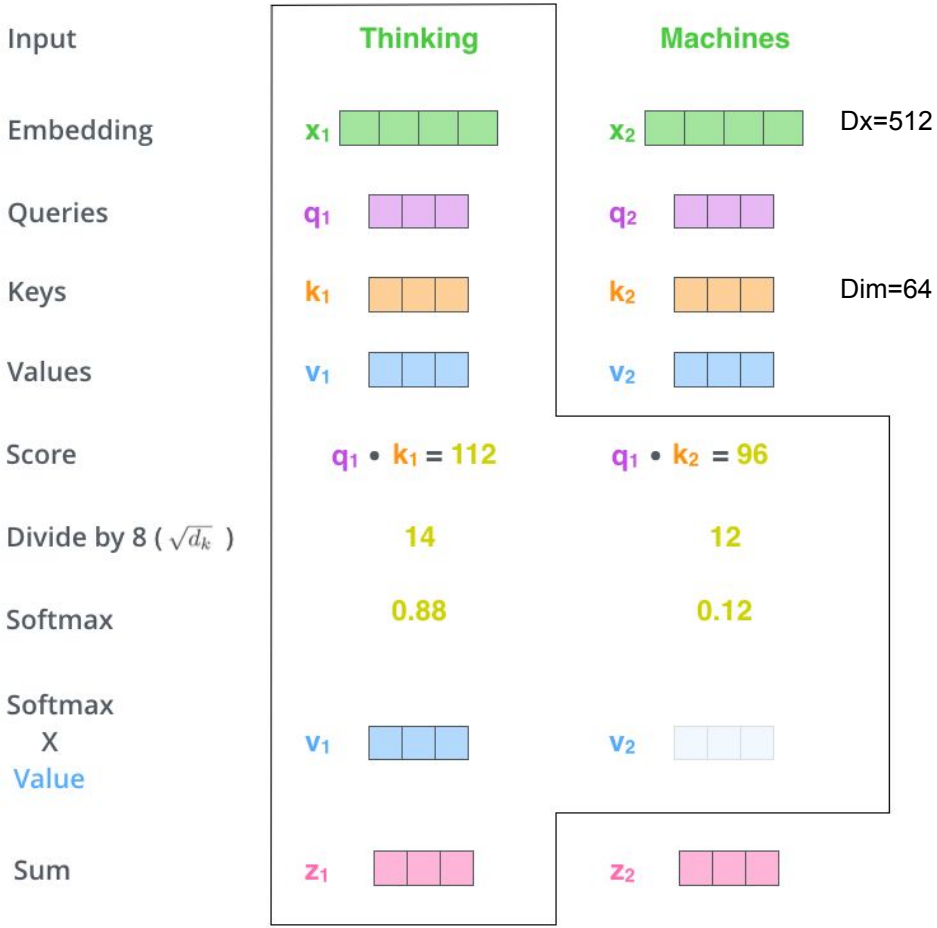
If you drive down the road and follow the river bank, you should find the Bank of America on your right.

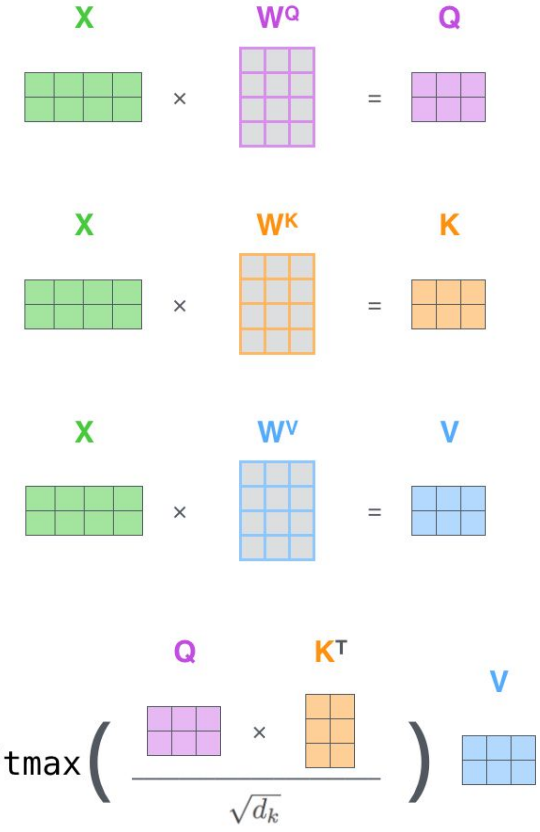
(river bank) (Bank of America)

→ (1.1, 0.3, ...) (0.4, 0.1, ...) (2.3, 1.2, ...) (0.3, 0.2, ...) ...

Self-attention is a variant of attention that processes a sequence by replacing each element by a weighted average of the rest of the sequence.
In Transformer (2017) is implemented as **scaled-dot product attention**

[IASI AI] Scaled Dot Product Attention in Detail



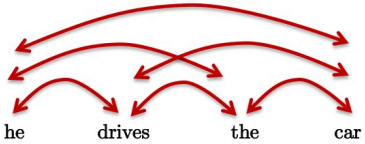


The Illustrated Transformer

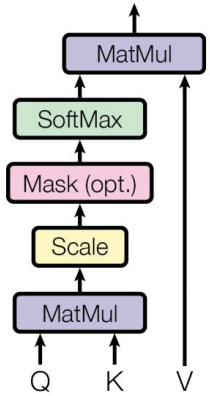
BERT - Multi-Head Attention = 8x Scaled Dot Product Attention

Properties :

- (Much) simpler layer than RNNs (matrix-matrix multiplications)
- Parallelizable (no recurrence/sequential process)

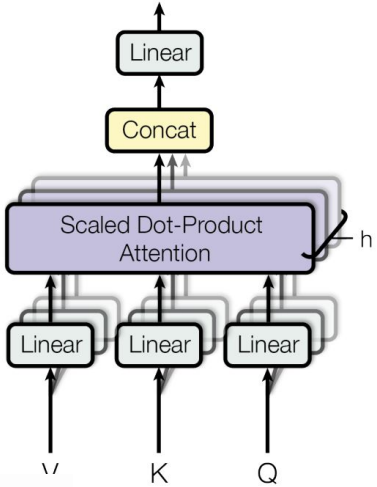


Scaled Dot-Product Attention



x 8 =

Multi-Head Attention



So 8 attentions allow us to view relevancy from 8 different "perspectives". This eventually pushes the overall accuracy higher, at least empirically. The transformation also reduces their output dimension so even 8 attentions are used, the computational complexity remains about the same.

*"multi-head attention allows the model to jointly attend to information from different representation **subspaces** at different positions. With a single attention head, averaging inhibits this."*

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{n}})\mathbf{V}$$

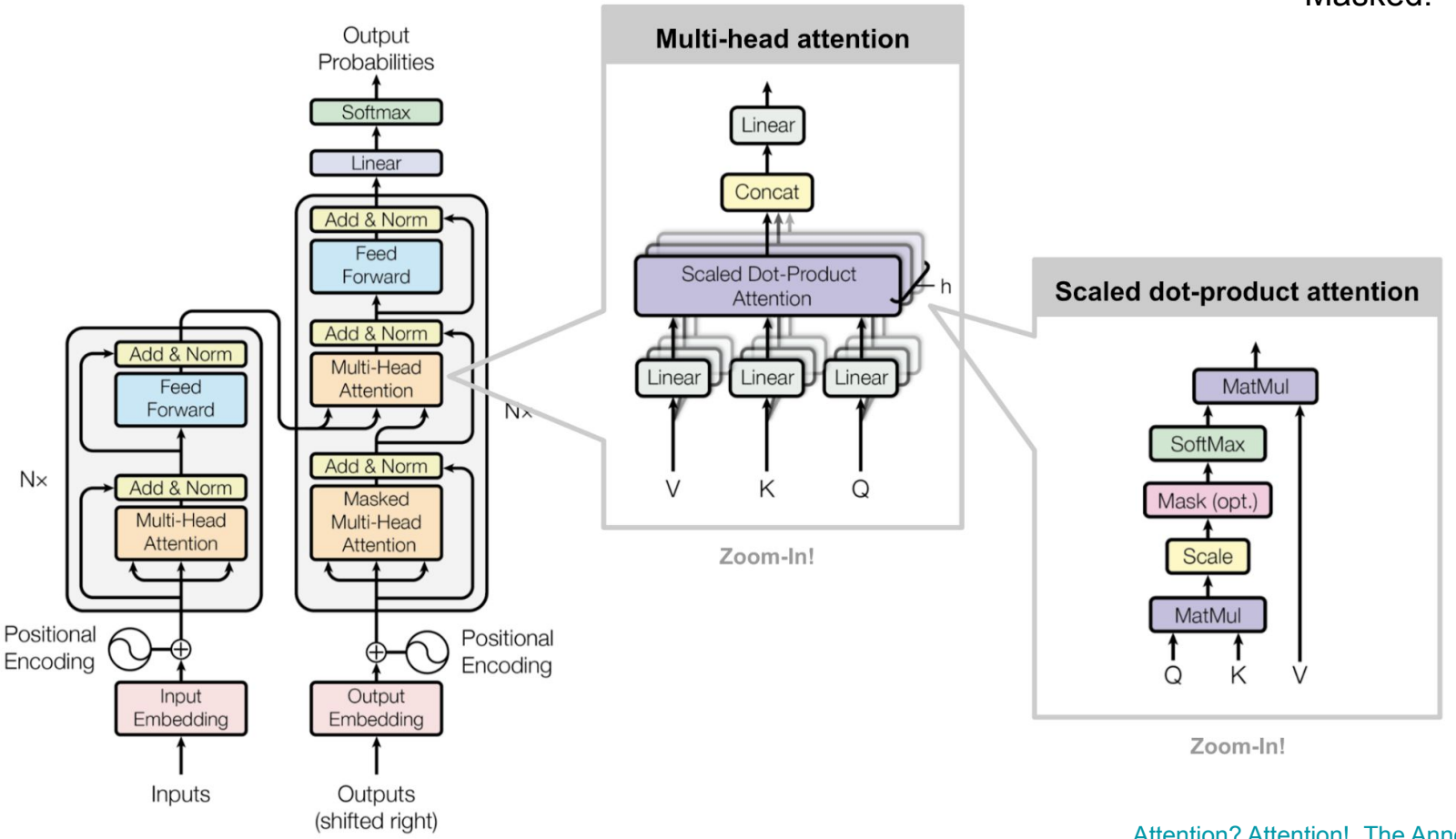
Note: scale to avoid gradients too small pb

$$\text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = [\text{head}_1;\ldots;\text{head}_h]\mathbf{W}^O$$
$$\text{where head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$$

Attention? Attention!, Google's Transformer solves a tricky problem in machine translation

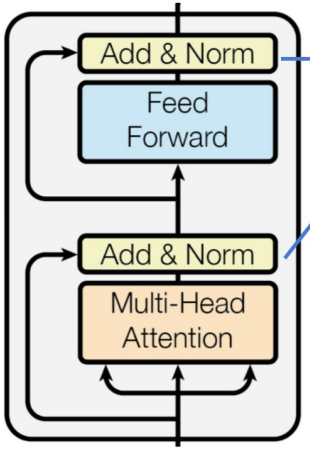
Transformer Architecture (BERT - encoder only, GPT-2 - decoder only)

Masked:



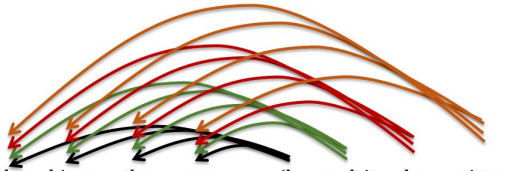
Attention? Attention!, The Annotated Transformer

Encoder and Decoder with Layer normalization and Skip (residual) connection

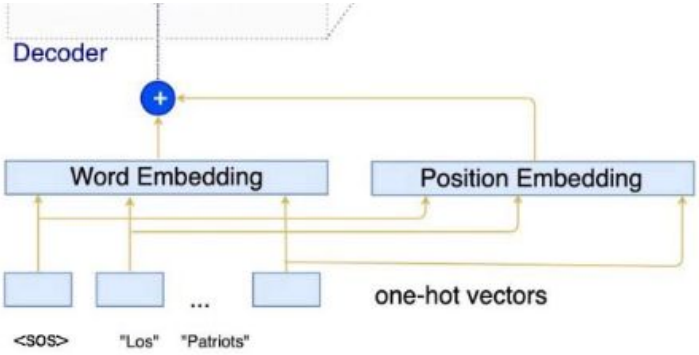


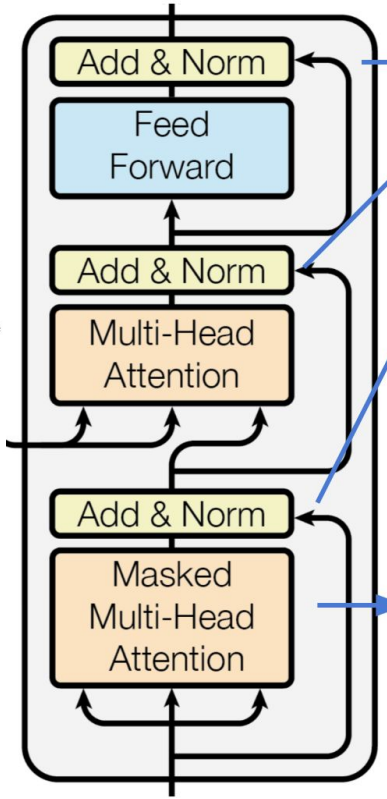
Nx
= 6

LayerNorm(x + SubLayer(x))



$$K = V = H_{\text{Encoder}}$$

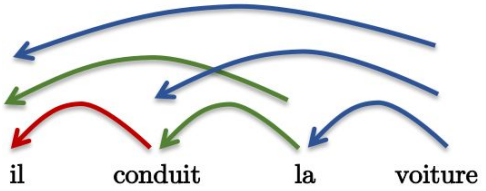




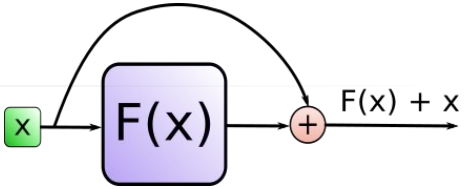
LayerNorm(x + SubLayer(x))

Nx
= 6

Masked: not to use the information in the future.

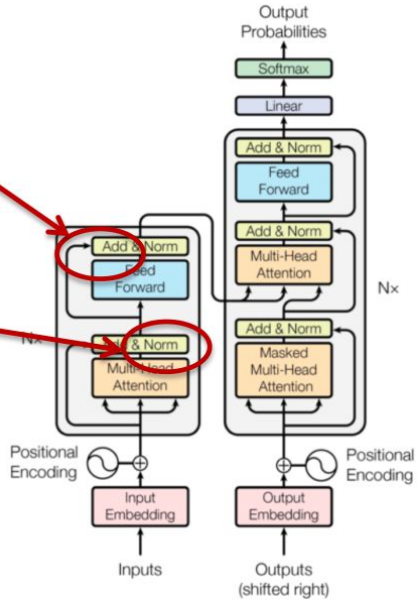


Residual Connections and Z-Scoring



- Importance of skip connections/residual blocks :
  - Generic to NNs :
    - Good for backpropagation.
    - Allows to skip a few steps of reasoning in the k-hop attention mechanism if necessary.
  - Specific to Transformers :
    - Carry positional information to next layers.

- Normalization layer :
  - Simple z-scoring.
  - Optimization trick to flatten the loss landscape and speed-up SGD.



Z-score normalization is a strategy of normalizing data that avoids this outlier issue. The formula for Z-score normalization is below:

$$\frac{value - \mu}{\sigma}$$

Many machine learning algorithms attempt to find trends in the data by comparing features of data points. However, there is an issue when the features are on drastically different scales.

Normalization

From NTM to Transformer architecture - Neural Translator Machine - LSTM

MODEL ARCHITECTURE	Details
LSTM	- process one word from sentence at each time-step - slow but works with variable size sentences since LSTM supports many to many sequences
LSTM Encoder-Decoder	- informational bottleneck = the idea (thought vector) is extracted from sentence (lossy compression = sentence summarization) => more natural translator Disadvantage: - a variable length sentence is encoded in a fixed sized vector (thought vector), so can't handle long sentences
Bi-LSTM Encoder-Decoder	- encoder and decoder use 2 LSTM layers for both directions - Bidirectional = gathers word context from both directions Advantage: - has better context info from left & right for translating current word
Bi-LSTM Encoder-Decoder + Soft Attention (NTM)	- solves the pb with the fixed sized vector using attention - It is a shallow architecture, still problems to use distant relations. - **ELMo** embeddings are learned with this architecture - **ELMo** learns embedding from relations between current token and those previous ones or following it (both directions) Issue: ELMo train two *separate* models that each take the left and right context into account but do not train a model that uses both at the same time.

From NTM to Transformer architecture – Transformer (XL), GPT-2

Transformer Attention Is All You Need, Character-Level Language Modeling with Deeper Self-Attention	- encoder-decoder attention-based architecture - multi head self-attention + FCN + skip (residual) links - masked Attention in Decoder - dual training task (i.e. masked language model and next sentence prediction)
GPT-2 **(**from OpenAI: read me)	- only decoder, auto-regressive training – next word prediction, very good for language generation (fake news generator) *Our model, called GPT-2 (a successor to GPT), was trained simply to predict the next word in 40GB of Internet text. Due to our concerns about malicious applications of the technology, we are not releasing the trained model*
Non-Autoregressive Transformer	- worse accuracy but 8x inference speed due to parallel decoder, no masked attention needed
Evolved Transformer	Enhancing Transformer with Neural Architecture Search
Sparse Transformer (from OpenAI - MuseNET)	- an algorithmic improvement of the *attention* mechanism to extract patterns from sequences 30x longer than possible previously -2 level sparse connectivity attention for modelling long distance interdependencies
Transformer XL Tr. XL - Attentive Language Models Beyond a Fixed-Length Context , Tr-XL - Combining Transformers and RNNs Into a State-of-the-art Language Model	Transformer + Segment level recurrence mechanism and Relative Positional Encoding - add recurrence in order to use context from previous sentence - learns dependency that is 80% longer than recurrent neural networks (RNNs) and 450% longer than vanilla Transformers and is up to 1,800+ times faster than vanilla Transformers during evaluation

BERT **B**i-**D**irectional **E**ncoder **R**epresentations from **T**ransformers (from Google)	- stacks multiple transformer encoders-decoders on top of each other - powerful deep architecture - useful for language understanding tasks but less for lang. generation - dual training task (i.e. masked language model and next sentence prediction) - large-scale TPU training **Advantages:** - no slow sequential LSTM, the input is the entire sentence, - powerful multi-head self-attention - multi–language model available, - unsupervised pre-trained on Wikipedia, - allows fine-tuning for specific tasks **Disadvantage:** - input size limited to 512 tokens (2 sentences or a paragraph) - very large model - hard to put in production, - slow at inference time also due to auto-regressive decoder The [MASK] token used in training does not appear during fine-tuning BERT generates predictions independently (masked tokens) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
BERT MultiQA	MultiQA: An Empirical Investigation of Generalization and Transfer in Reading Comprehension
BertQA-Attention-on-Steroids (Code)	A more focused context to query and query to context attention (C2Q attention)

From NTM to Transformer architecture - ERNIE

ERNIE Enhanced Language Representation with Informative Entities	- Transformer + Knowledge graph - can take full advantage of lexical, syntactic, and knowledge information simultaneously
XLM (from Facebook) (code) Cross-lingual Language Model Pretraining	- Transformer + structured memory layer (key product kNN memory) - Causal + Masked + Translation Language Model - 2X inference speed (Large Memory Layers with Product Keys)
MASS Masked Sequence to Sequence Pre-training for Language Generation	unlike **XLM** pre-trains both the encoder and decoder jointly to predict a missing sentence fragment
Snorkel MeTaL (from Stanford)	Paper (Weak Supervision for Multi-Task Learning)
MT-DNN (from Microsoft) - as ensemble - 2nd in GLUE, - super-human (human is 3rd place)	- multi-task BERT (BERT + multi-task learning + knowledge distillation) Multi-Task Deep Neural Networks for Natural Language Understanding (Code) Microsoft makes Google's BERT NLP model better
ERNIE 2.0 (Enhanced Representation through kNowledge IntEgration) (from **Baidu**) ERNIE 2.0 model almost comprehensively outperforms BERT and XLNet on English tasks .	Baidu's ERNIE 2.0 Beats BERT and XLNet on NLP Benchmarks ERNIE 2.0 is built as a continual pretraining framework to continuously gain enhancement on knowledge integration through multi-task learning, enabling it to more fully learn various lexical, syntactic and semantic information through massive data. We construct several tasks to capture different aspects of information in the training corpora: ● **Word-aware Tasks**: to handle the lexical information ● **Structure-aware Tasks**: to capture the syntactic information ● **Semantic-aware Tasks**: in charge of semantic signals The tasks include named entity prediction, discourse relation recognition, sentence order prediction are leveraged in order to enable the models to learn language representations

XLNET

Generalized Autoregressive Pretraining for Language Understanding

- super-human, see GLUE leaderboard

Transformer XL + TSSA (Two-stream self-attention) + bidirectional data input
- can we train a model to incorporate bidirectional context while avoiding the [MASK] token and parallel independent predictions?
- improved pre-training: learns current token embedding from previous seen tokens but for all permutations of sentence)
- To avoid leaking the information of the position to be predicted, use Two-Stream Self-Attention (TSSA)
Understanding XLNet, Paper Dissected: "XLNet" Explained

RoBERTa: A Robustly Optimized BERT Pretraining Approach - from Facebook

Blog

Paper

Github

BERT was significantly under-trained, and can match or exceed the performance of every model published after it. These results highlight the importance of previously overlooked design choices, and raise questions about the source of recently reported improvements. RoBERTa is trained with dynamic masking, FULL-SENTENCES without NSP (next sentence prediction) loss, large mini-batches and a larger byte-level BPE.
- use a novel dataset, CC-NEWS

RoBERTa uses the BERT LARGE configuration (355 million parameters) with an altered pre-training pipeline. Yinhan Liu and her colleagues made the following changes:

- Increased training data size from 16Gb to 160Gb by including three additional datasets.
- Boosted batch size from 256 sequences to 8,000 sequences per batch.
- Raised the number of pretraining steps from 31,000 to 500,000.
- Removed the next sentence prediction (NSP) loss term from the training objective and used full-sentence sequences as input instead of segment pairs.
- Fine-tuned for two of the nine tasks in the GLUE natural language understanding benchmark as well as for SQuAD (question answering) and RACE (reading comprehension).

ALBERT

ALBERT: A Lite BERT For Self-Supervised Learning of Language Representations

- current state-of-the-art
- super-human, see GLUE leaderboard

Google's ALBERT Is a Leaner BERT; Achieves SOTA on 3 NLP Benchmarks

Core innovations:

Factorized embedding parameterization - (For BERT - WordPiece embedding size E is tied with the hidden layer size H, i.e., E ≡ H) Researchers isolated the size of the hidden layers from the size of vocabulary embeddings by projecting one-hot vectors into a lower dimensional embedding space and then to the hidden space, which made it easier to increase the hidden layer size without significantly increasing the parameter size of the vocabulary embeddings.

Cross-layer parameter sharing - Researchers chose to share all parameters across layers to prevent the parameters from growing along with the depth of the network. As a result, the large ALBERT model has about 18x fewer parameters compared to BERT-large.

Inter-sentence coherence loss (**try to predict the order of two consecutive segments of text**) - In the BERT paper, Google proposed a next-sentence prediction technique to improve the model's performance in downstream tasks, but subsequent studies found this to be unreliable. Researchers used a sentence-order prediction (SOP) loss to model inter-sentence coherence in ALBERT, which enabled the new model to perform more robustly in multi-sentence encoding tasks.

Dataset: For pretraining baseline models, researchers used the BOOKCORPUS and English Wikipedia, which together contain around 16GB of uncompressed text.

Experiment results: The ALBERT model significantly outperformed BERT on the language benchmark tests SQuAD1.1, SQuAD2.0, MNLI SST-2, and RACE.

DistillBERT Smaller, faster, cheaper, lighter: Introducing DistilBERT, a distilled version of BERT	**Knowledge Distillation - Transferring generalization capabilities** *Overall, our distilled model, DistilBERT, has about half the total number of parameters of BERT base and retains 95% of BERT's performances on the language understanding benchmark GLUE.* *Here we are fine-tuning by distilling a question answering model into a language model previously pre-trained with knowledge distillation! In this case, we were able to reach interesting performances given the size of the network: 86.2 F1 and 78.1 EM, ie. within 3 points of the full model!*
TinyBERT TinyBERT: Distilling BERT for Natural Language Understanding	performs transformer distillation at both the pre-training and task-specific learning stages
Google T5 (Text-to-Text Transfer Transformer) (Code) **Same model with no changes is used for different tasks**	By combining the insights from our exploration with scale and our new "Colossal Clean Crawled Corpus" (about 750 GB), we achieve state-of-the-art results on many benchmarks covering summarization, question answering, text classification, and more. To facilitate future work on transfer learning for NLP, we release our dataset, pre-trained models, and code. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Crucially, our text-to-text framework allows us to directly apply the same model, objective, training procedure, and decoding process to every task we consider.

[IASI AI]

Thank You!

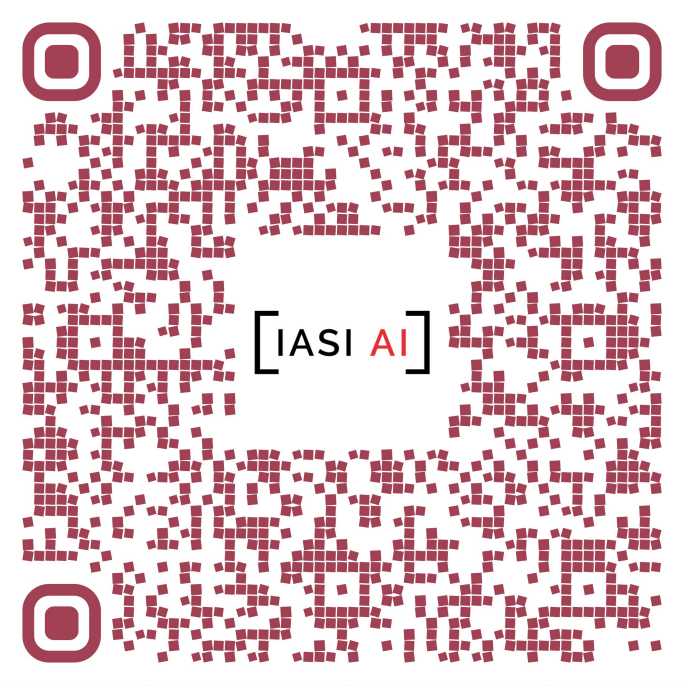
iasi.ai

fb.me/AI.Iasi/

meetup.com/IASI-AI/

[IASI AI]

We ❤ Feedback



[IASI AI]

Community partners

The input sequence of tokens is mapped to a sequence of embeddings, which is then passed into the encoder.
The encoder = a stack of “blocks”, a block = two subcomponents: a **self-attention layer** + a **small feed-forward network**.
Layer normalization is applied to the input of each subcomponent
A **residual skip connection** adds each sub-component’s input to its output.
Dropout is applied within the feed-forward network, on the skip connection, on the attention weights, and at the input and output of the entire stack.

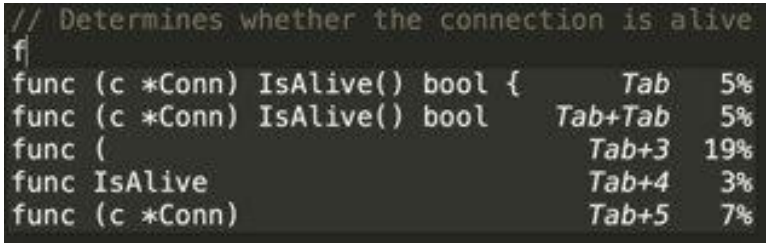
The decoder is similar in structure to the encoder except that it includes **a standard attention mechanism** after each self-attention layer that attends to the output of the encoder.
The self-attention mechanism in the decoder also uses a form of **autoregressive or causal self-attention, which only allows the model to attend to past outputs**.

The output of the final decoder block is fed into a dense layer with a softmax output, whose weights are shared with the input embedding matrix.
All attention mechanisms in the Transformer are split up into independent “heads” whose outputs are concatenated before being further processed.

Since self-attention is order-independent (i.e. it is an operation on sets), it is common to **provide an explicit position signal** to the Transformer. While the original Transformer used **a sinusoidal position signal or learned position embeddings**, it has recently become more common to **use relative position embeddings**. Instead of using a fixed embedding for each position, relative position embeddings produce a different learned embedding according to the offset between the “key” and “query” being compared in the self-attention mechanism.”

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Deep TabNine: A Powerful AI Code Autocompleter For Developers (Youtube)
- a GPT-2 (transformer) model trained on around 2 million files from GitHub



A Google Brain Program Is Learning How to Program (Paper: Neural Networks for Modeling Source Code Edits)
Codota - AI Pair Programmer (Intellij plugin)
What is AutoPandas?
Kite launches Line-of-Code Completions, goes cloudless, and secures $17 million in funding (Pycharm plugin)
Neural Code Search (NCS) from Facebook
- NCS takes natural language queries and returns relevant code snippets retrieved directly from a large codebase
MIT - Toward artificial intelligence that learns to write code (Learning to Infer Program Sketches)
Microsoft wants to apply AI ‘to the entire application developer lifecycle
Director of AI at Tesla talks about Software 2.0:
Advantages: Computationally homogeneous, Simple to bake into silicon, Constant running time,
Constant memory use, It is highly portable, It is very agile, Modules can meld into an optimal whole, It is better than you.

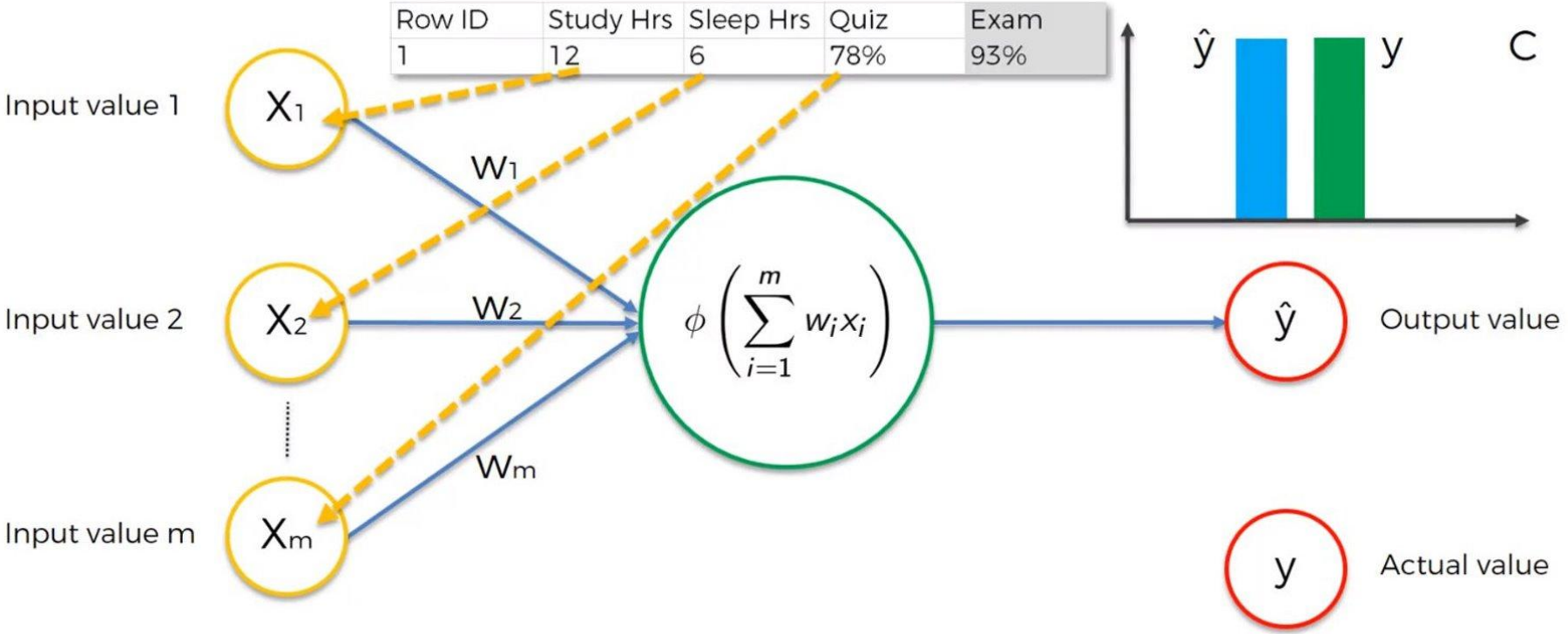
[How uncertainty could help a machine hold a more eloquent conversation]

AI startup Gamalon developed a clever new way for chatbots and virtual assistants to converse with us. "...the system can deal with uncertainty by making its best guess about what someone means. It also provides a conversational memory: you could ask “What about tomorrow?” after previously asking what the weather is like today. "

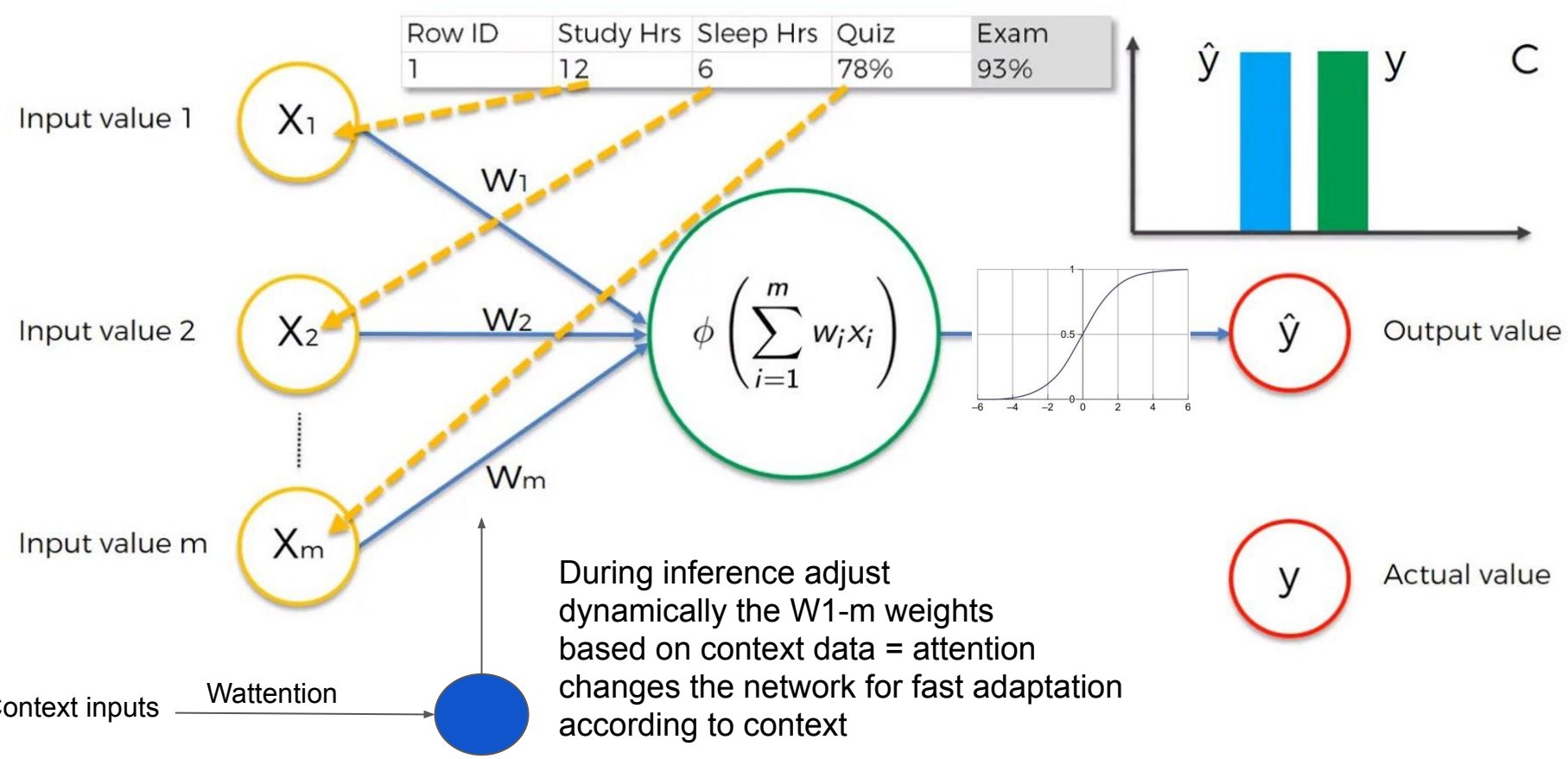
[Genpact Awarded U.S. Patent for its Natural Language Understanding Technology Framework]

"Our NLU technology gets around this by applying more contextual understanding with a linguistics-based approach, in addition to a statistical approach"

What the neuron is doing - adjust the importance of inputs and sum them up



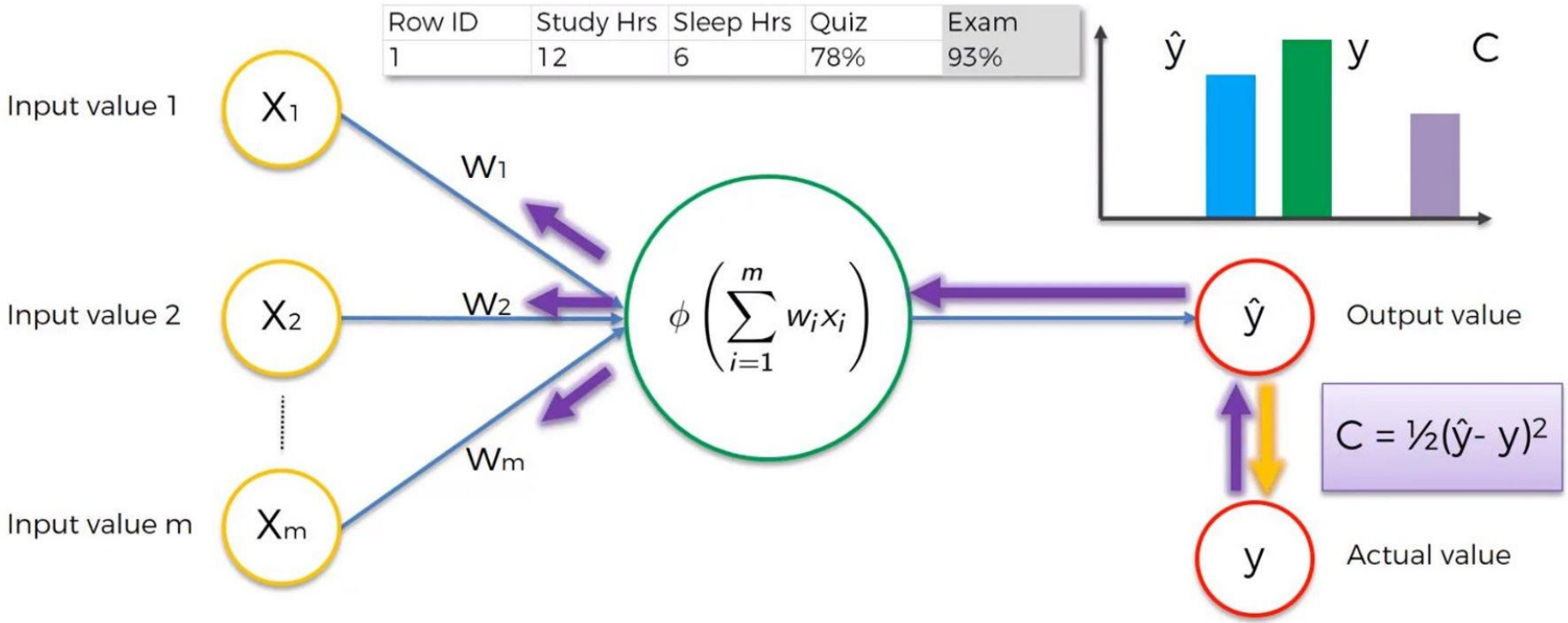
What the neuron is doing - add nonlinearity for better expressivity, add attention for fast weights



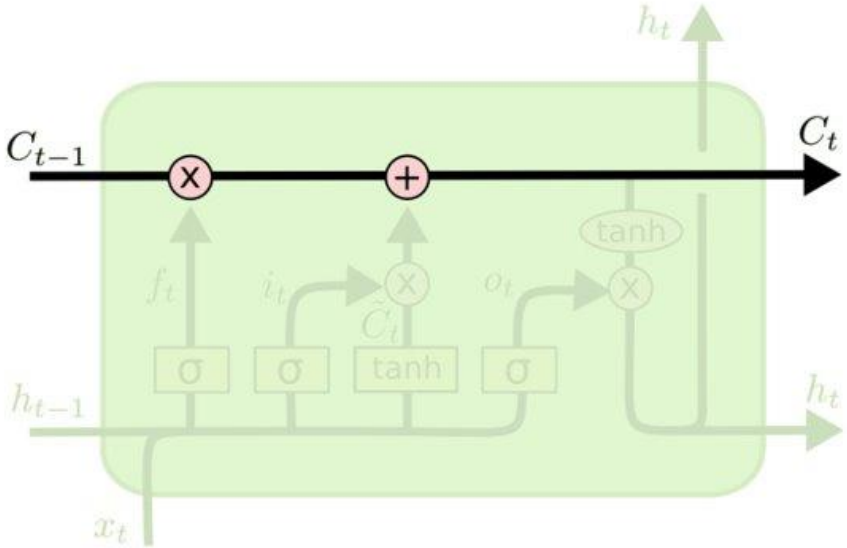
During inference adjust dynamically the W1-m weights based on context data = attention changes the network for fast adaptation according to context

Exam type - limited time but easy quiz vs long time but hard quiz

Error is back-propagated to find and adjust guilty synapses (parameters)



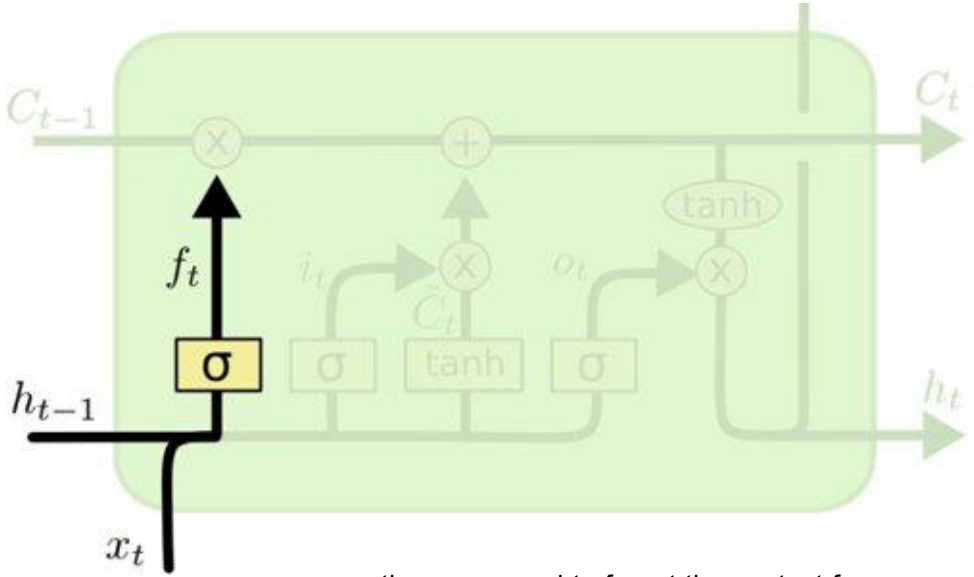
Cell state:



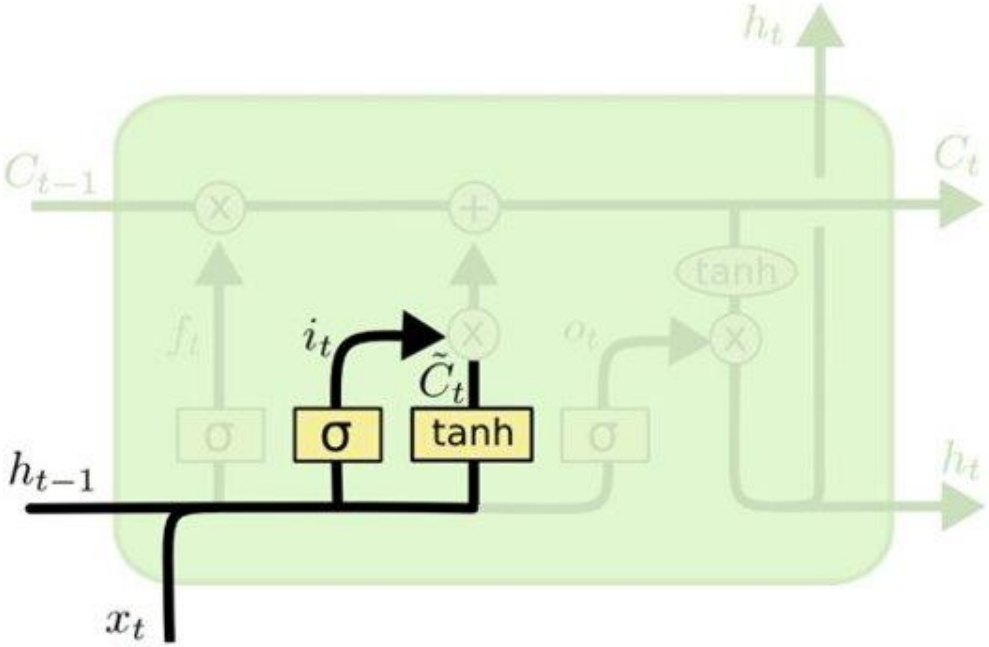
The internal state of mind: The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged.

Forget gate:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

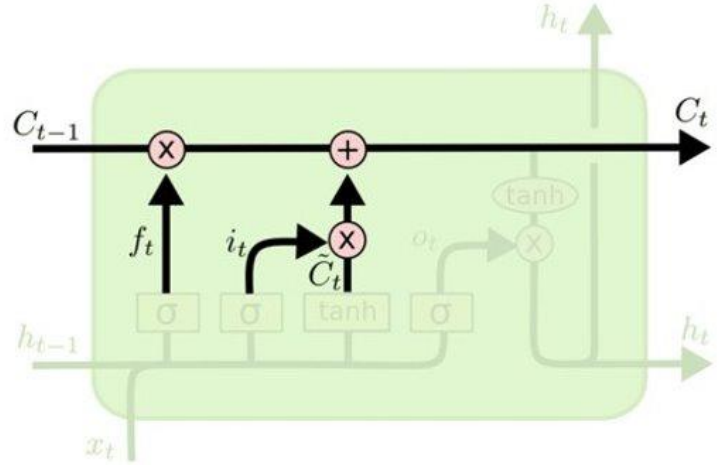


- sometimes we need to forget the context from the past
- we decide what to forget and how much based on current input and previous output



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

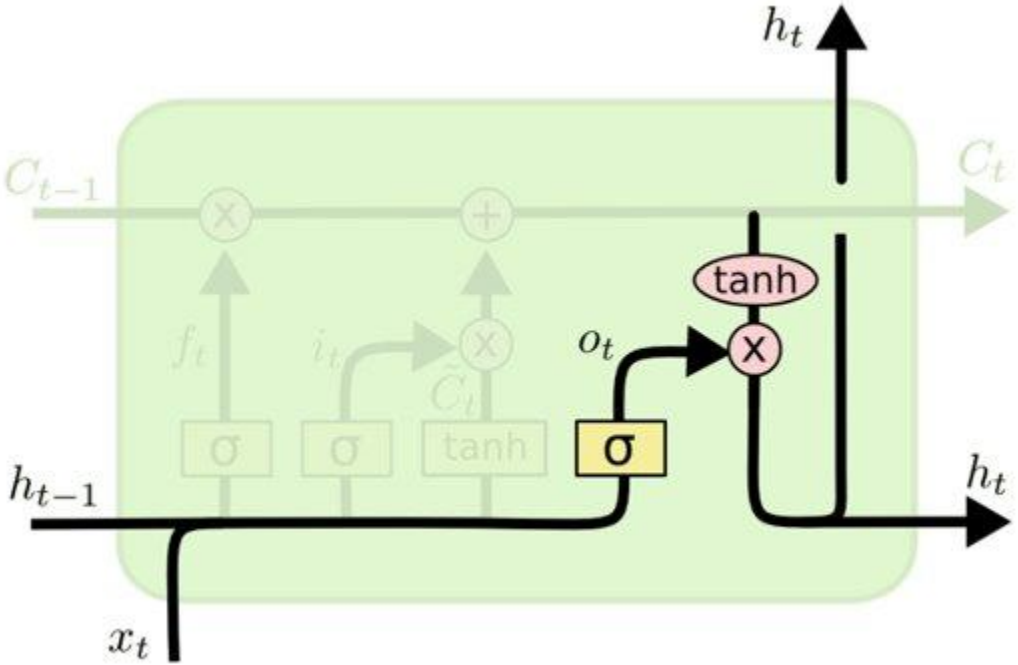


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

No more Vanishing gradient!
When we backpropagate and take the derivative of Ct with respect to Ct−1, **this added term just disappears**!

Why tanh?
To overcome the vanishing gradient problem, we need a function whose second derivative can sustain for a long range before going to zero. *tanh* is a suitable function with the above property.

- next output / action = the internal state of mind scaled down and censored by relevant context



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

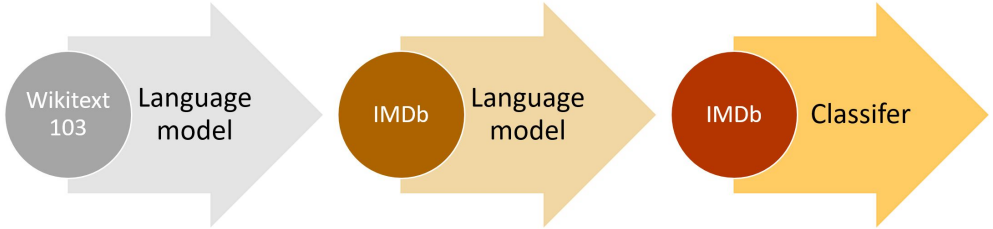
$$h_t = o_t * \tanh\left(C_t\right)$$

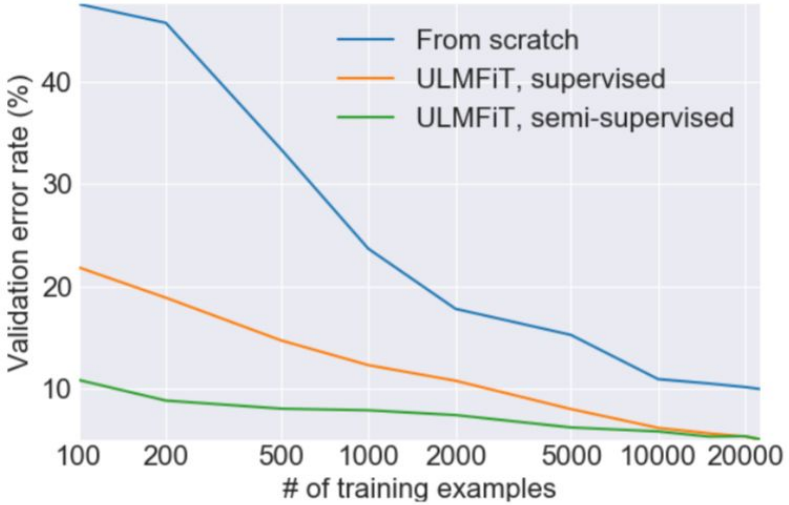
LSTM Disadvantages:

- we have a long sequential path from older past cells to the current one with additive and forget branches attached to it - hard to handle distant relations
- slow to train and not hardware friendly
- limited memory
- shallow architecture (no deep layers)

Universal Language Model Fine-tuning for Text Classification (ULMFiT)

- pretraining + fine-tuning
- uses AWD-LSTM (LSTM + special dropout for LSTM)



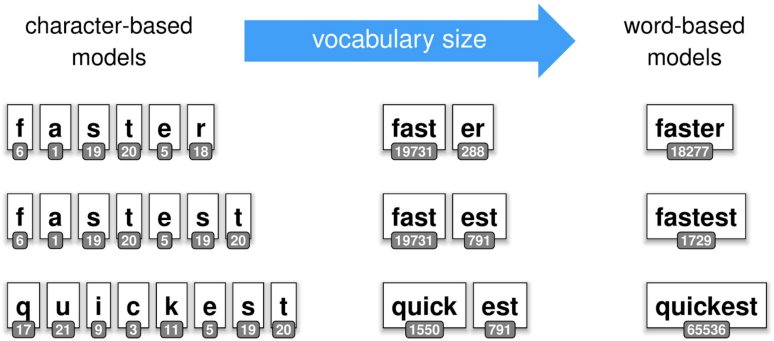


Understanding building blocks of ULMFIT ,
Efficient multi-lingual language model fine-tuning ,
Introducing state of the art text classification with universal language models

We show that we can fine-tune efficient monolingual language models that are competitive with multilingual BERT, in many languages, on a few hundred examples.

- pretraining + fine-tuning
- uses QRNN:
  - **Quasi-Recurrent Neural Network** (LSTM + convolutions) = parallelizable LSTM
  - **16x faster computation speed but also has better performance**
  - QRNN combines **best of two worlds: parallel nature of convolutions and time dependencies of LSTMs**
- trained on 100 labeled documents in the target language, outperforms multi-lingual BERT

Uses Subword tokenization
(better for exotic langs, reduces vocab size):



Understanding building blocks of ULMFIT ,Efficient multi-lingual language model fine-tuning , Introducing state of the art text classification with universal language models

One-Hot Encoding as NN inputs for categorical variables (can be colors or...words in a vocabulary)

One word in a dictionary = like a categorical variable.
A categorical variable it is usually converted to one-hot encoding to be passed as input to a NN. (to avoid NN to cheat while learning and use the index order)

One-Hot Encoding drawbacks:

- For high-cardinality variables- those with many unique categories - **the dimensionality of the transformed vector becomes unmanageable**.
- The mapping is completely uninformed: **“similar” categories are not placed closer to each other** in embedding space. (Big Issue: Backpropagation does not work on discrete variable)

```
# One Hot Encoding Categoricals

books = ["War and Peace", "Anna Karenina",
         "The Hitchhiker's Guide to the Galaxy"]

books_encoded = [[1, 0, 0],
                 [0, 1, 0],
                 [0, 0, 1]]

Similarity (dot product) between First and Second = 0
Similarity (dot product) between Second and Third = 0
Similarity (dot product) between First and Third = 0
```

Neural Network Embeddings Explained. Neural Network Embeddings: from inception to simple,
How to One Hot Encode Sequence Data in Python

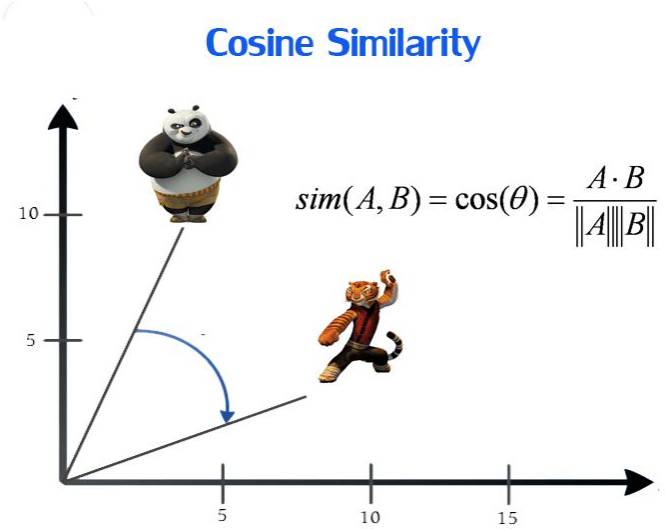
Embeddings - meaningfully encode words in a continuous vector space

Solution : **Embed each word into a continuous vector space in which words can be easily compared for similarity. They are based on the *distributional hypothesis* stating that a word's meaning can be inferred from the contexts it appears in.**

We can train a mini neural network on NLP supervised task(s) to ***learn embeddings***. *(Ex:* predict missing word, next word, sentiment). The embeddings vectors will be learned as network parameters = a matrix of size (Dict size * Emb. Dim).

Example of embeddings: **Word2vec**, **Skip-gram**, **Glove**, **ELMo**.

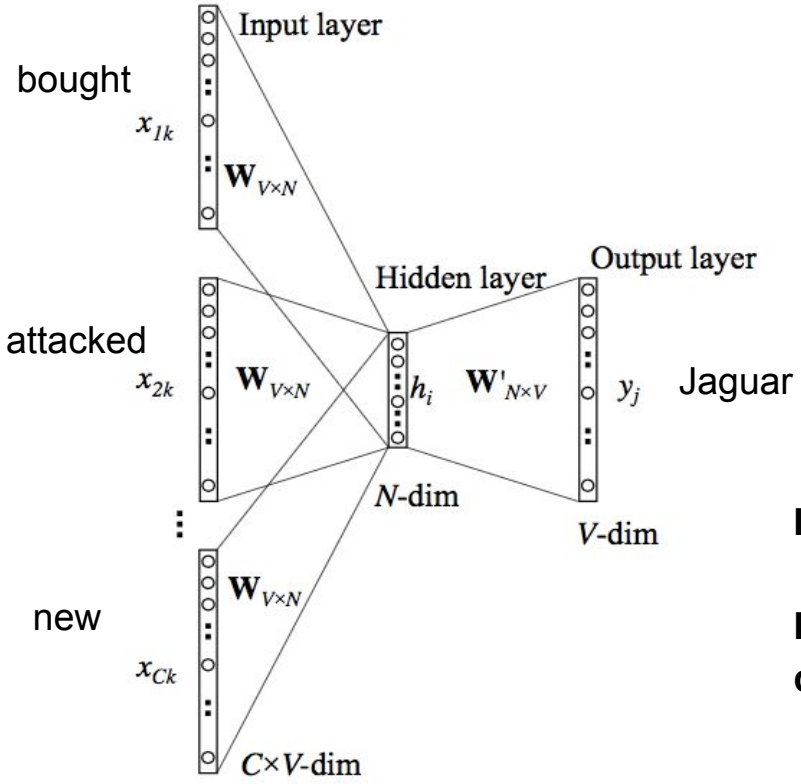
```
# Idealized Representation of Embedding

books = ["War and Peace", "Anna Karenina",
         "The Hitchhiker's Guide to the Galaxy"]

books_encoded_ideal = [[0.53,  0.85],
                       [0.60,  0.80],
                       [-0.78, -0.62]]

Similarity (dot product) between First and Second = 0.99
Similarity (dot product) between Second and Third = -0.94
Similarity (dot product) between First and Third = -0.97
```



Build your own Skip-gram Embeddings and use them in a Neural Network, Learning Word Embedding, The Illustrated Word2vec, The Current Best of Universal Word Embeddings and Sentence Embeddings, Extracting knowledge from knowledge graphs using Facebook Pytorch BigGraph

Word2Vec Embeddings - extract hidden layer from a NN trained to predict a word from the context

Common Bag Of Words (CBOW) method - use a simple NN with 1 hidden layer:



Note: Does not use non-linear activation

Word2Vec Big Disadvantage:
They learn an embedding for the word **type** and not the word **token** in the **context:**

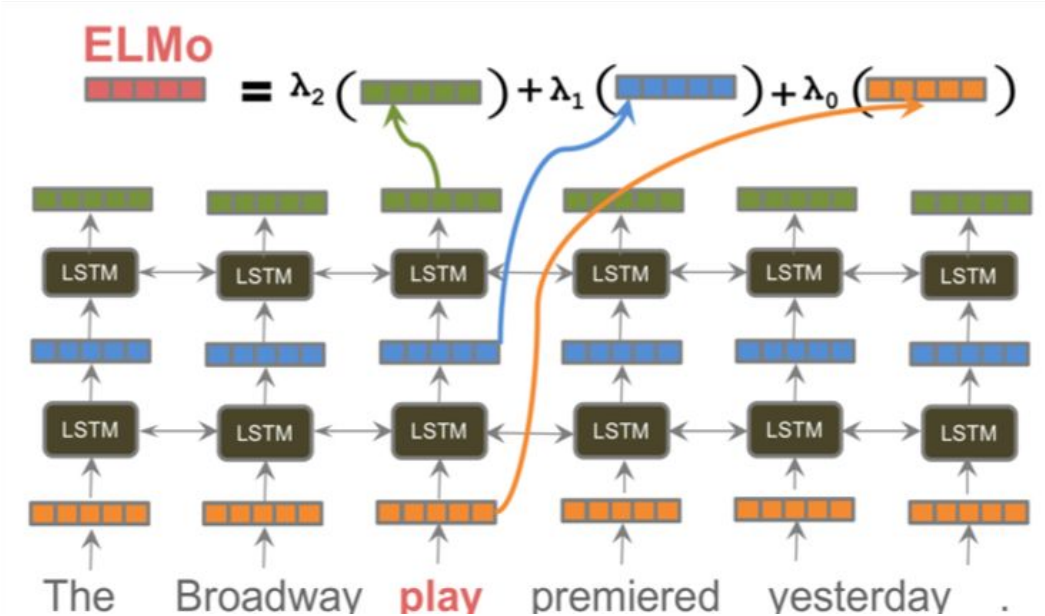
"jaguar" will have the same embedding in both
"I just bought a new Jaguar" and
"A Jaguar has attacked them"

Interesting arithmetics: King – Man + Woman = Queen

Embeddings = a simple form of transfer learning for quickly learning more complex tasks

Visualise embeddings: Embedding Projector
Introduction to Word Embedding and Word2Vec

ELMo [code] pre-trains a bidirectional LSTM-based character-level language model and extracts contextual word vectors as learned combinations of hidden states (see Figure 4). For downstream tasks, these word embeddings are used as inputs without any changes (so, they serve like features).



$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}.$$

Motivation to combine different layer representations:

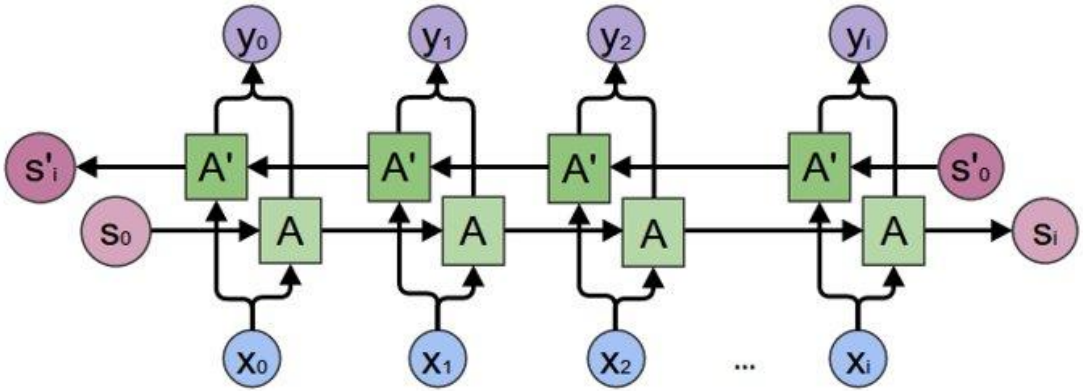
Lower level layers might contain more information useful for low-level NLP tasks (POS tagging, syntactic parsing…) and higher layers useful for high level NLP tasks (sentiment analysis, question answering…)

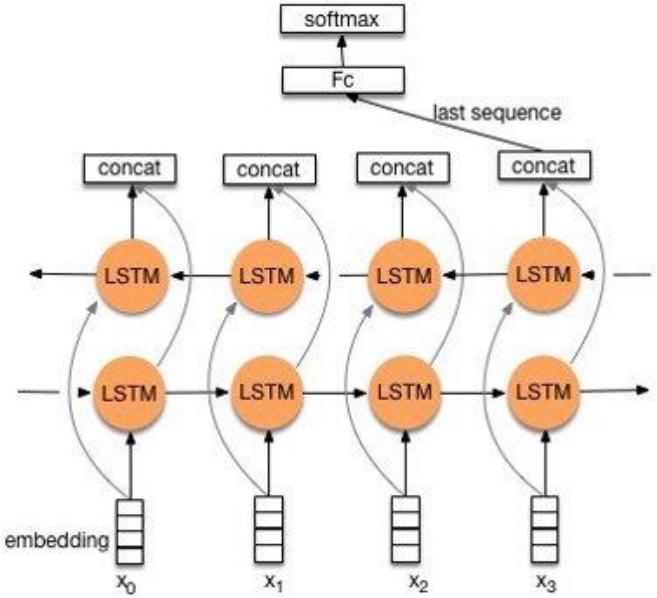
The Rise of Language Models (Talk)

Universal Embeddings: embeddings that are pre-trained on a large corpus and can be plugged in a variety of downstream task models (sentimental analysis, classification, translation…) Knowledge embeddings

BiLSTM (Bidirectional LSTM)

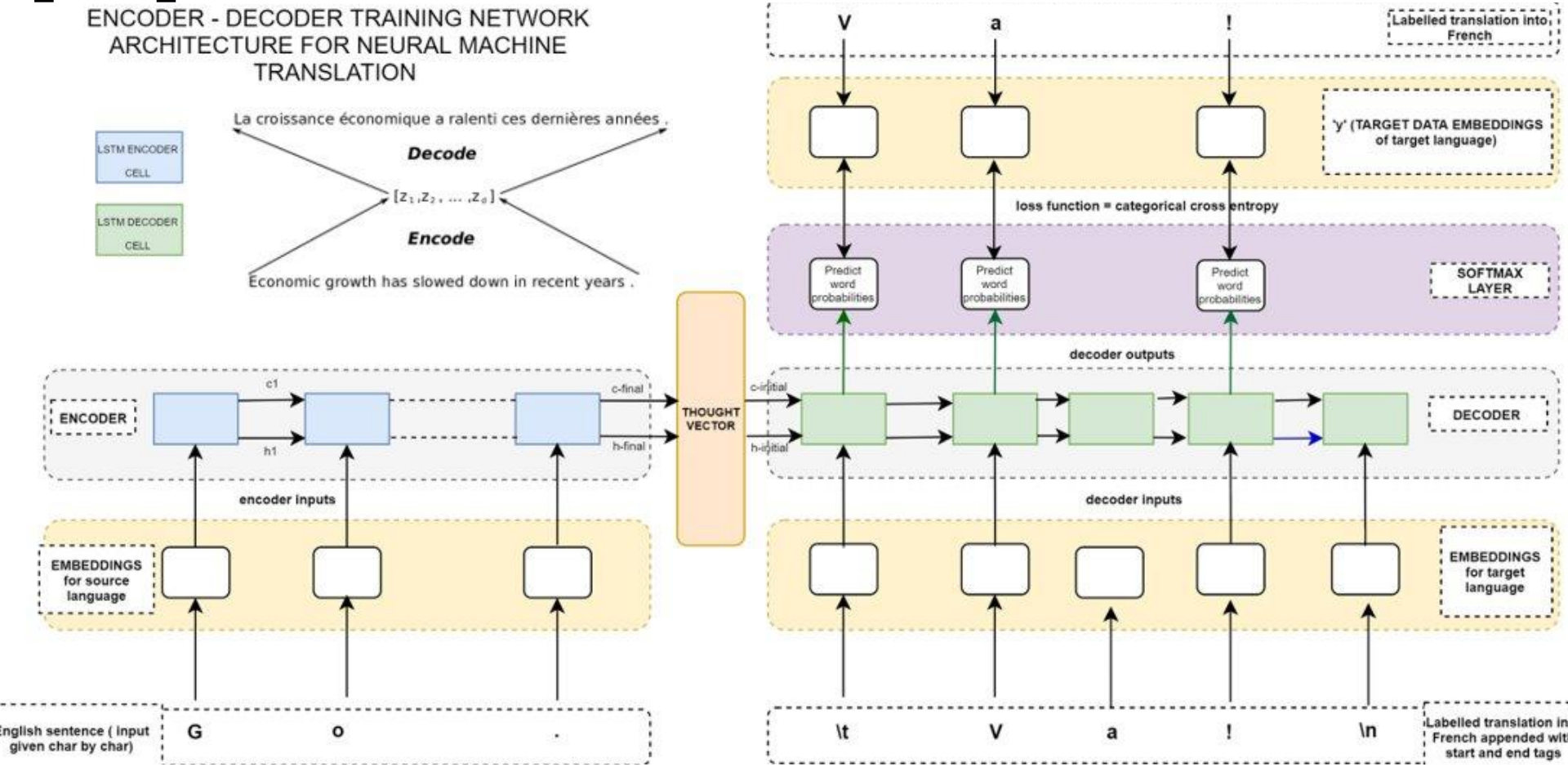
A special kind of RNN is the *Bi-directional RNN* (BiRNN). While standard RNNs remember the historical context by ‘remembering’ past data, BiRNNs also traverse in the reverse direction, to understand context from the future:



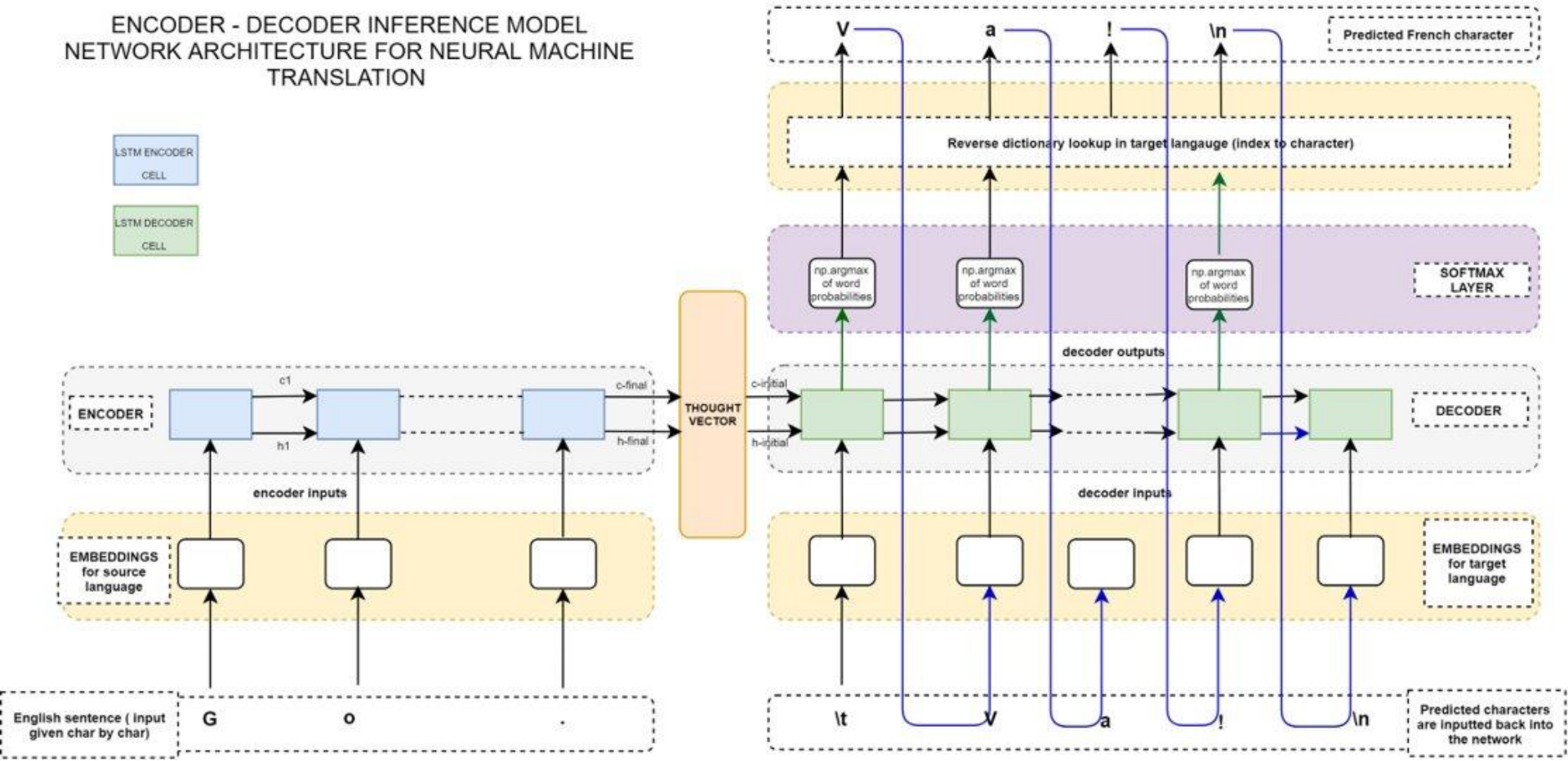


It is important to note that while RNNs can theoretically remember any length of history, they are usually much better at incorporating short-term context rather than long-term information (>20–30 steps apart).

Neural Translator Machine – Training

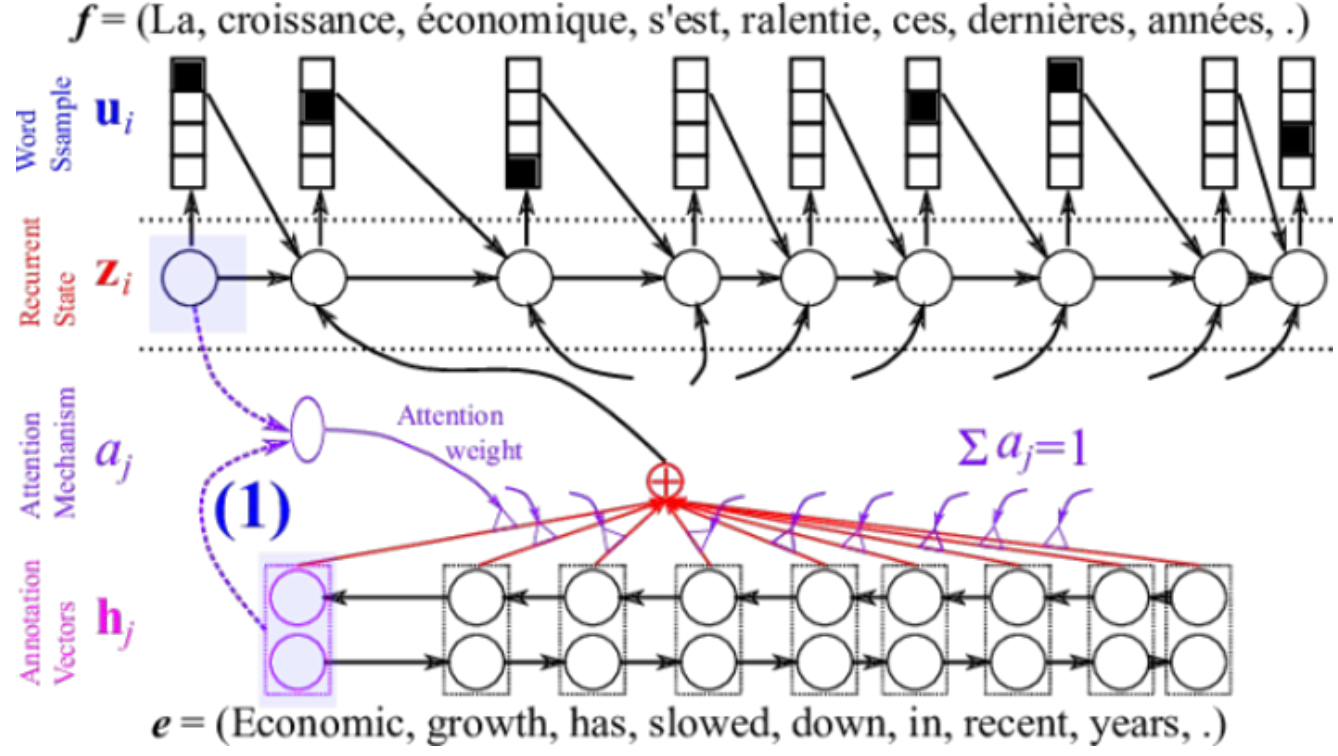


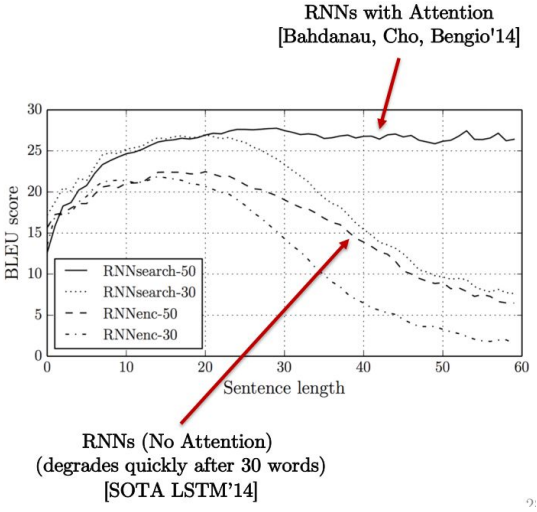
Neural Machine Translation—Using seq2seq with Keras


ENCODER - DECODER INFERENCE MODEL NETWORK ARCHITECTURE FOR NEURAL MACHINE TRANSLATION
LSTM ENCODER CELL
LSTM DECODER CELL
ENCODER
c1
h1
c-final
h-final
THOUGHT VECTOR
c-initial
h-initial
encoder inputs
EMBEDDINGS for source language
English sentence (input given char by char)
G
o
.
V
a
!
\n
Predicted French character
Reverse dictionary lookup in target langauge (index to character)
np.argmax of word probabilities
np.argmax of word probabilities
np.argmax of word probabilities
SOFTMAX LAYER
decoder outputs
DECODER
decoder inputs
EMBEDDINGS for target language
\t
V
a
!
\n
Predicted characters are inputted back into the network

Bi-LSTM + (Additive) Soft Attention Mechanism

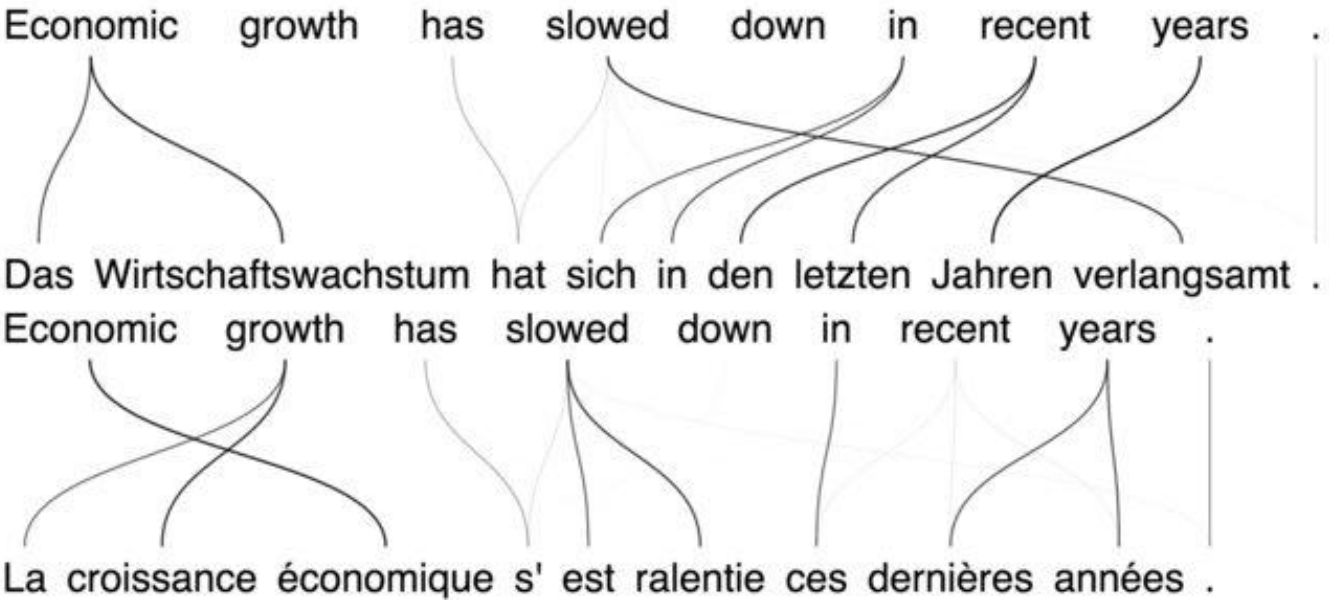
- It's hard for the RNN to remember *everything* that happened over the sequence in this single vector
- Bidirectional recurrent neural networks for encoding a source sentence.
- Attention Mechanism takes into consideration what has been translated and one of the source words.
- Attention Mechanism returns a single scalar corresponding to a relevance score of the j-th source word.





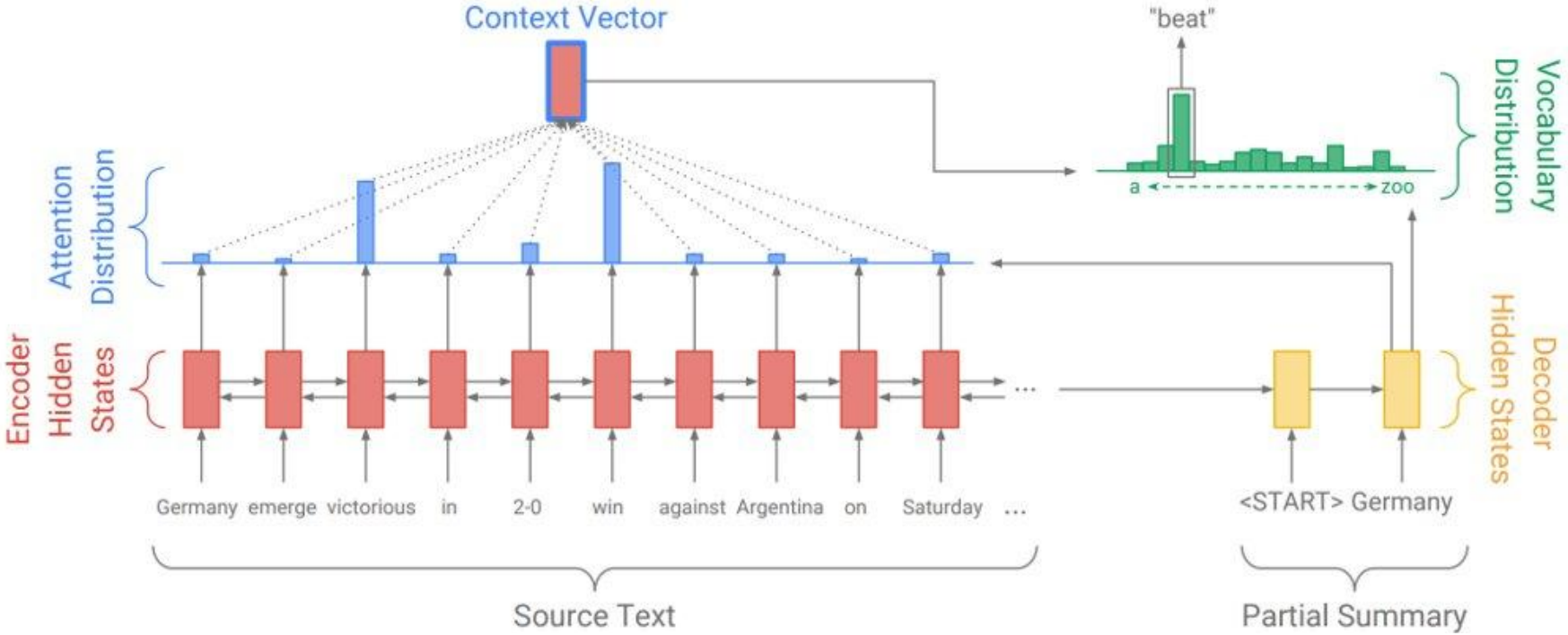
Intro to NMT with GPUs

Attention in Neural Networks is modeled after the way humans *focus on a particular subset of their sensory input, and tune-out the rest*. Attention is also first big step towards reasoning.



Attention? Attention!,How Does Attention Work in Encoder-Decoder Recurrent Neural Networks

[IASI AI] Abstractive text summarization (LSTM + attention) by using new words from vocabulary



Encoder-Decoder architecture, **decoder computes context attention over source words based on current summary, current decoder state and context vector is used to choose next word from vocabulary**

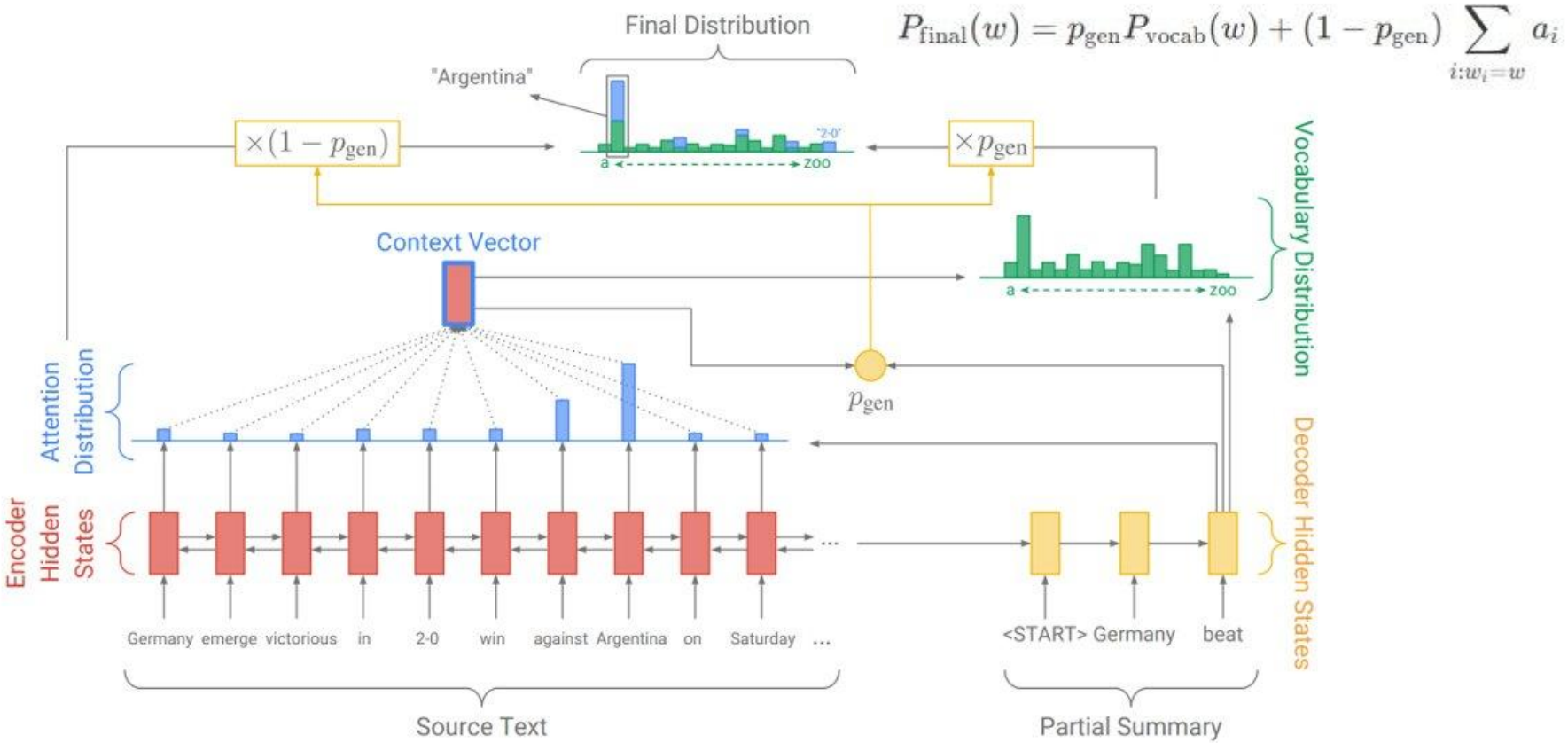
Issues:

Problem 1: The summaries sometimes **reproduce factual details inaccurately** (e.g. *Germany beat Argentina **3-2***)

Problem 2: The summaries sometimes **repeat themselves** (e.g. *Germany beat Germany beat Germany beat…*)

Taming Recurrent Neural Networks for Better Summarization

- ***best of both worlds*, combining both extraction (pointing) and abstraction (generating)**



two stages: modeling and inference, - scales to real-world settings

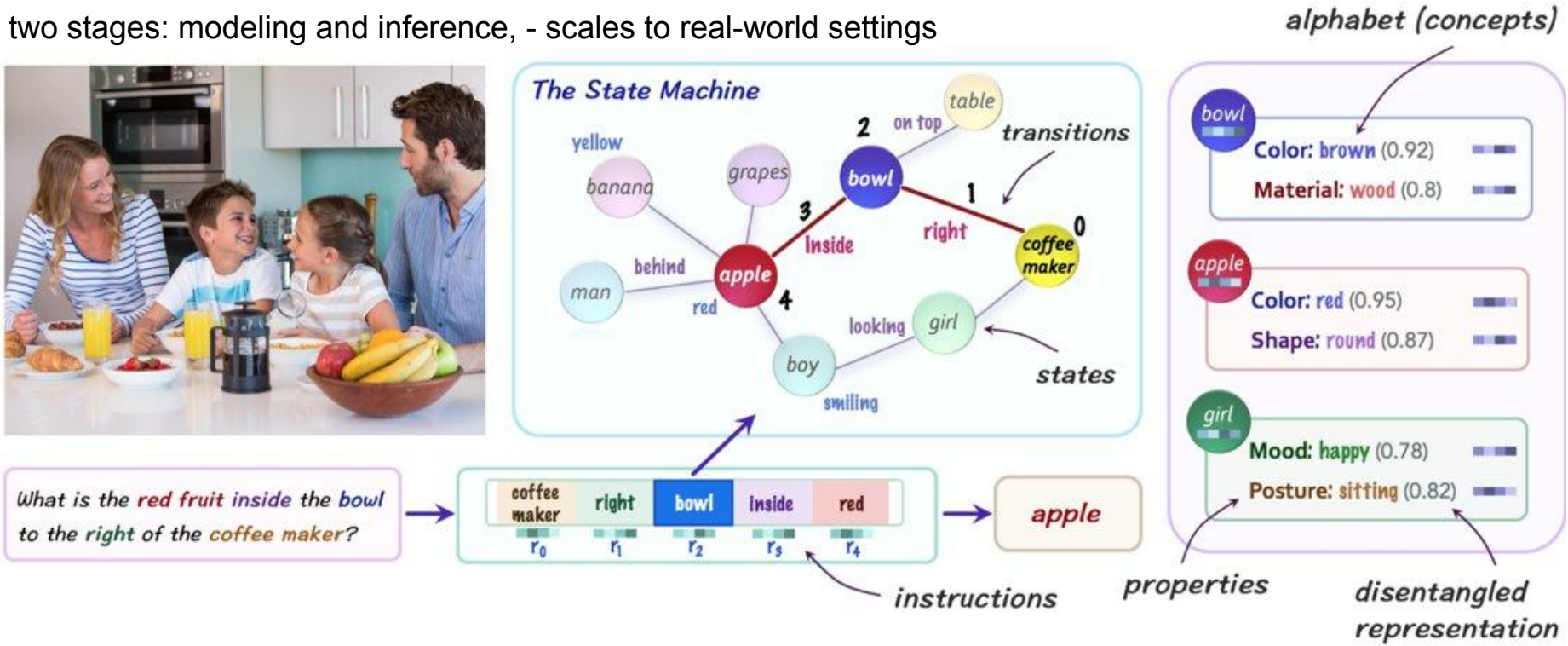


Figure 1: The Neural State Machine is a graph network that simulates the computation of an automaton. For the task of VQA, the model constructs a probabilistic scene graph to capture the semantics of a given image, which it then treats as a state machine, traversing its states as guided by the question to perform sequential reasoning.

Paper , Building Neural Network Models That Can Reason

We construct a probabilistic scene graph that consists of:

1) A set of object nodes S from the image each accompanied by a bounding box, a mask, dense visual features, and a collection of discrete probability distribution for each object's semantic properties (such as its color, material, shape, etc.)
2) A set of relation edges between the objects, each associated with a probability distribution of its semantic type (e.g. on top of, eating) and corresponding to a valid transition between the machine's states.

Note :
a) we convert visual and linguistic inputs (GLOVE embeddings) in terms of our vocabulary (concept embeddings),
b) next convert query to reasoning instructions with LSTM,
c) use instruction to redistribute our attention over the states (the objects) (use state transition function neural module)

A visualization of object masks from the inferred scene graphs, which form the basis for our model.

Notes: We use **Mask R-CNN** object detector. Use **ResNet-101** and **FPN** for features and region proposals respectively.

Use distance closeness between objects and **graph attention network** to predict relation types

1) What is the giraffe looking at? ***person*** ✓
2) Is the fence in front of the giraffe made of metal? ***no*** ✓
3) Is the woman's shirt blue or yellow? ***blue*** ✓
4) On which side of the image is the person? ***right*** ✓
5) Is there a child behind the giraffe? ***no*** ✗

1) What is the fruit to the right of the salad? ***strawberries*** ✓
2) Is the fork to the right of the salad? ***no*** ✓
3) Is the plate white and square? ***no*** ✓
4) Is the cup behind the round plate? ***yes*** ✓
5) What is the plate made of? ***paper*** ✗

1) Are there either scarves or hats that are not pink? ***no*** ✓
2) Do the bear's dress and the person's shirt have the same color? ***yes*** ✓
3) Is the bear sitting or standing? ***sitting*** ✓
4) What is the green object that the bear is sitting on? ***book*** ✓
5) Is the bear wearing white shoes? ***yes*** ✗

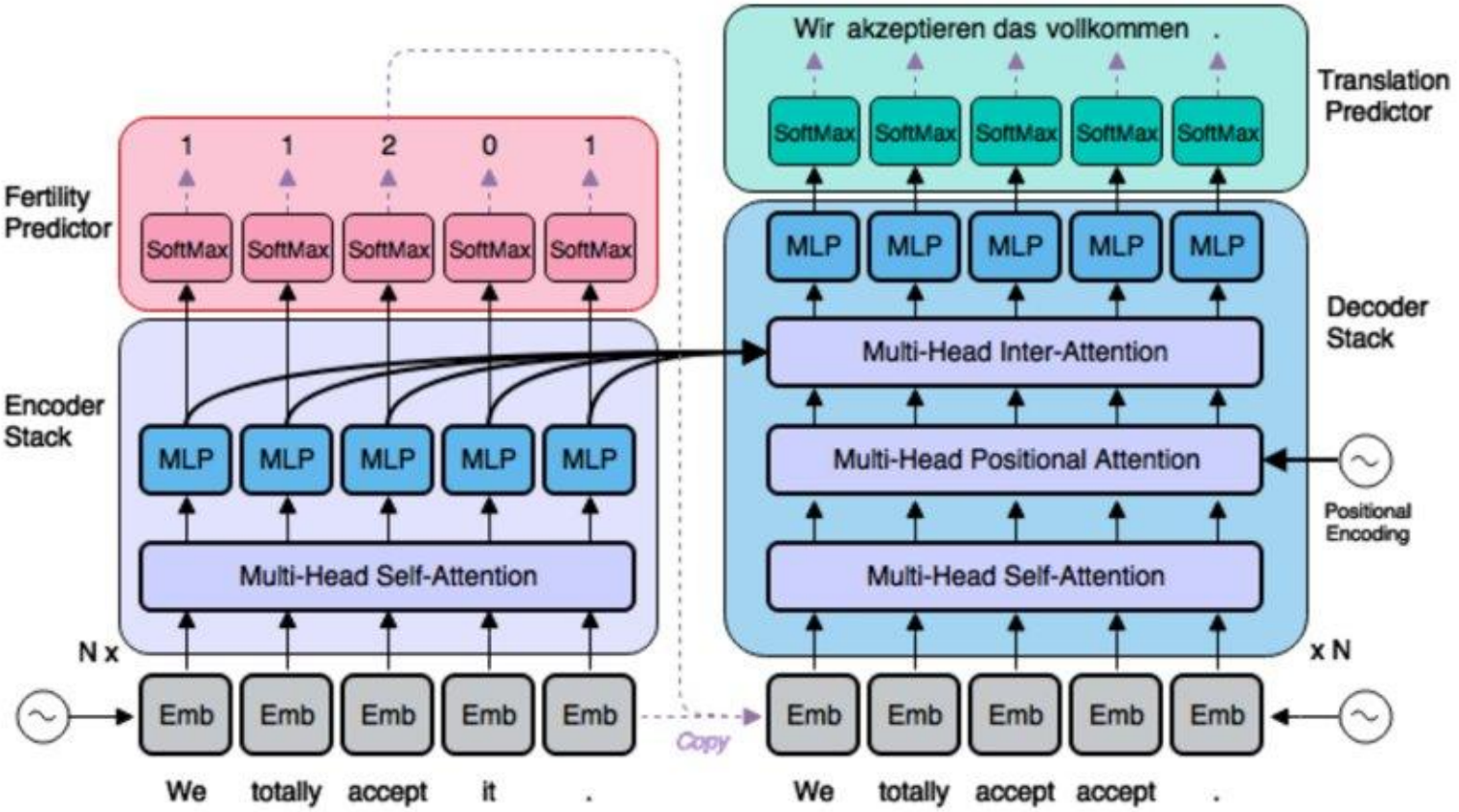
1) Are there either a chair or a clock in the image? ***no*** ✓
2) Are there any flowers behind the bed on the left of the room? ***yes*** ✓
3) What color is the appliance on the right? ***black*** ✓
4) Is the carpet brown or blue? ***brown*** ✓
5) Is the TV turned on? ***yes*** ✗

Figure 2: Question examples along with answers predicted by the NSM. The questions involve diverse reasoning skills such as multi-step inference, relational and spatial reasoning, logic and comparisons.

Model	Binary	Open	Consistency	Validity	Plausibility	Distribution	Accuracy
Human [39]	91.20	87.40	98.40	98.90	97.20	-	89.30
Global Prior [39]	42.94	16.62	51.69	88.86	74.81	93.08	28.90
Local Prior [39]	47.90	16.66	54.04	84.33	84.31	13.98	31.24
Language [39]	61.90	22.69	68.68	96.39	**87.30**	17.93	41.07
Vision [39]	36.05	1.74	62.40	35.78	34.84	19.99	17.82
Lang+Vis [39]	63.26	31.80	74.57	96.02	84.25	7.46	46.55
BottomUp [5]	66.64	34.83	78.71	96.18	84.57	5.98	49.74
MAC [38]	71.23	38.91	81.59	96.16	84.48	5.34	54.06
SK T-Brain*	77.42	43.10	90.78	96.26	85.27	7.54	59.19
PVR*	77.69	43.01	90.35	**96.45**	84.53	5.80	59.27
GRN	77.53	43.35	88.63	96.18	84.71	6.06	59.37
Dream	77.84	43.72	91.71	96.38	85.48	8.40	59.72
LXRT	77.76	44.97	92.84	96.30	85.19	8.31	60.34
NSM	**78.94**	**49.25**	**93.25**	**96.41**	84.28	**3.71**	**63.17**

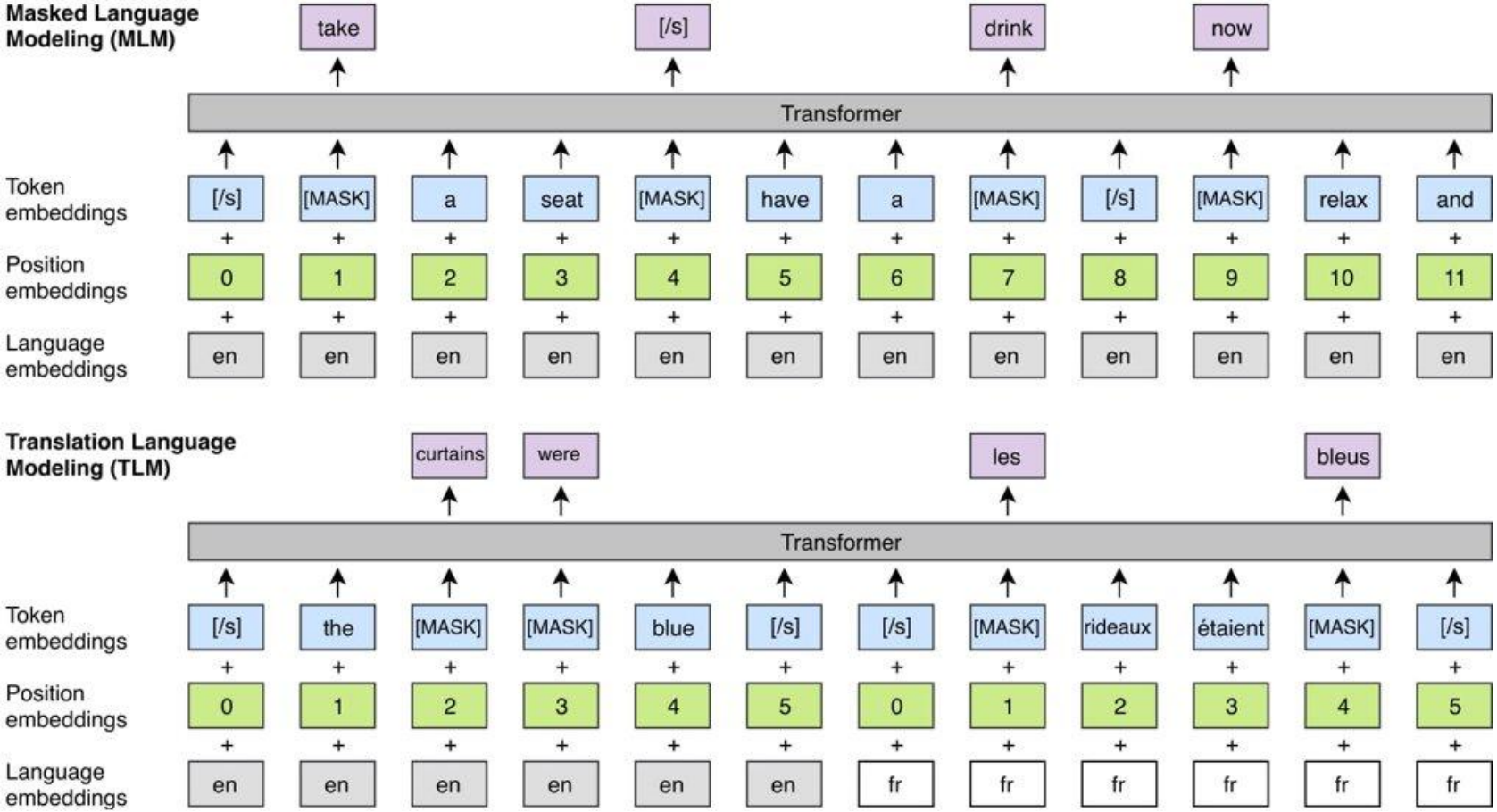
Next step: generate the scene graph unsupervised:
Generative Hierarchical Models for Parts, Objects, and Scenes

Non-Autoregressive Decoding - Parallel Decoder , 8x speed but a bit worse accuracy



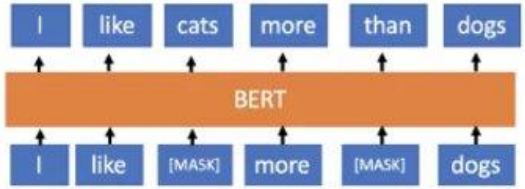
Advances in NLP in 2017

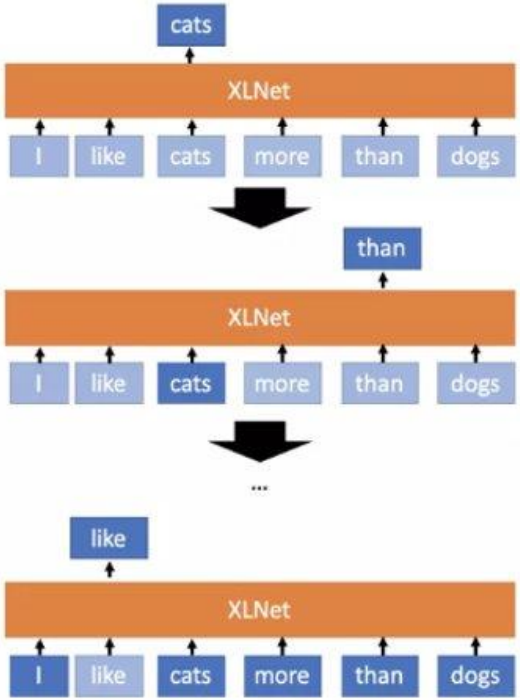
XLM - Cross-lingual Language Model Pretraining



XLM—Enhancing BERT for Cross-lingual Language Model

XLNET- Generalized Autoregressive Pretraining for Lang. Understanding



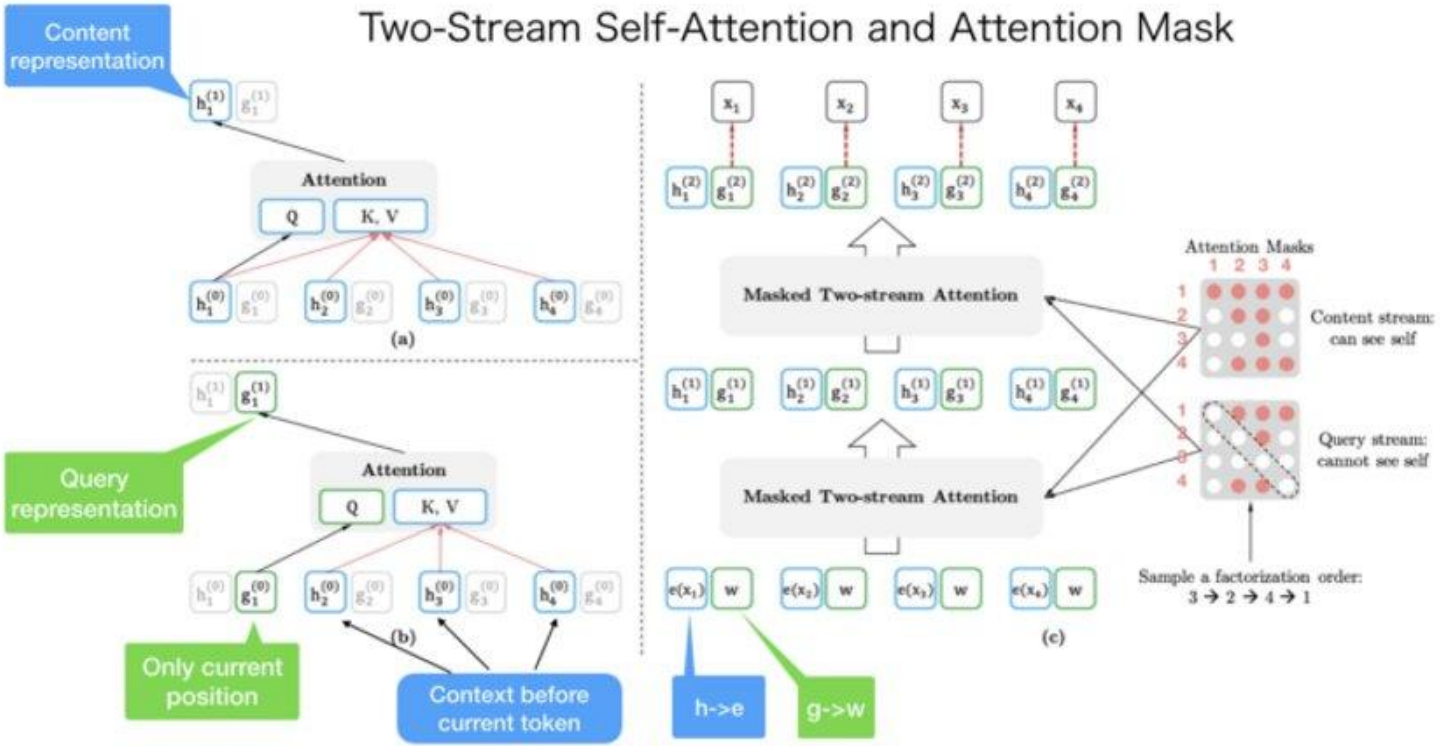


Permutation language modeling is more challenging compared to traditional language modeling which apparently causes the model to converge slowly. To address this problem, the authors chose to predict the last tokens in the permutation instead of predicting the entire sentence from scratch

The conceptual difference between BERT and XLNet. Transparent words are masked out so the model cannot rely on them. XLNet learns to predict the words in an arbitrary order but in an autoregressive, sequential manner (not necessarily left-to-right). BERT predicts all masked words simultaneously.

Paper Dissected: “XLNet: Generalized Autoregressive Pretraining for Language Understanding” Explained

XLNET- Two-stream self-attention



Permutation is handled by attention, We want to mask content of token to predict but not its position

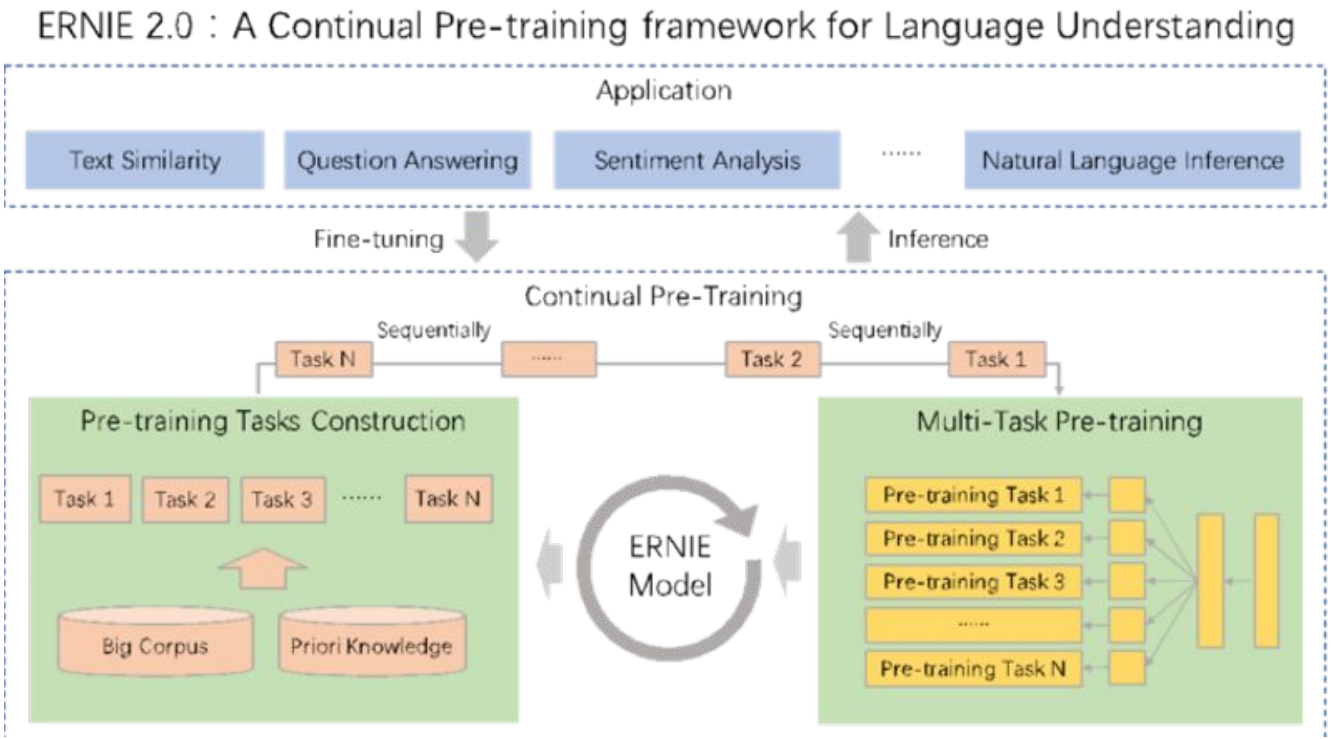
Two-Stream Self-Attention = Content stream attention + Query stream attention

vs BERT:

No longer needed MASK token which is problematic coz does not appear at inference. XLNET uses separate pos info instead.

No longer a problem if there are dependencies between 2 masked words.

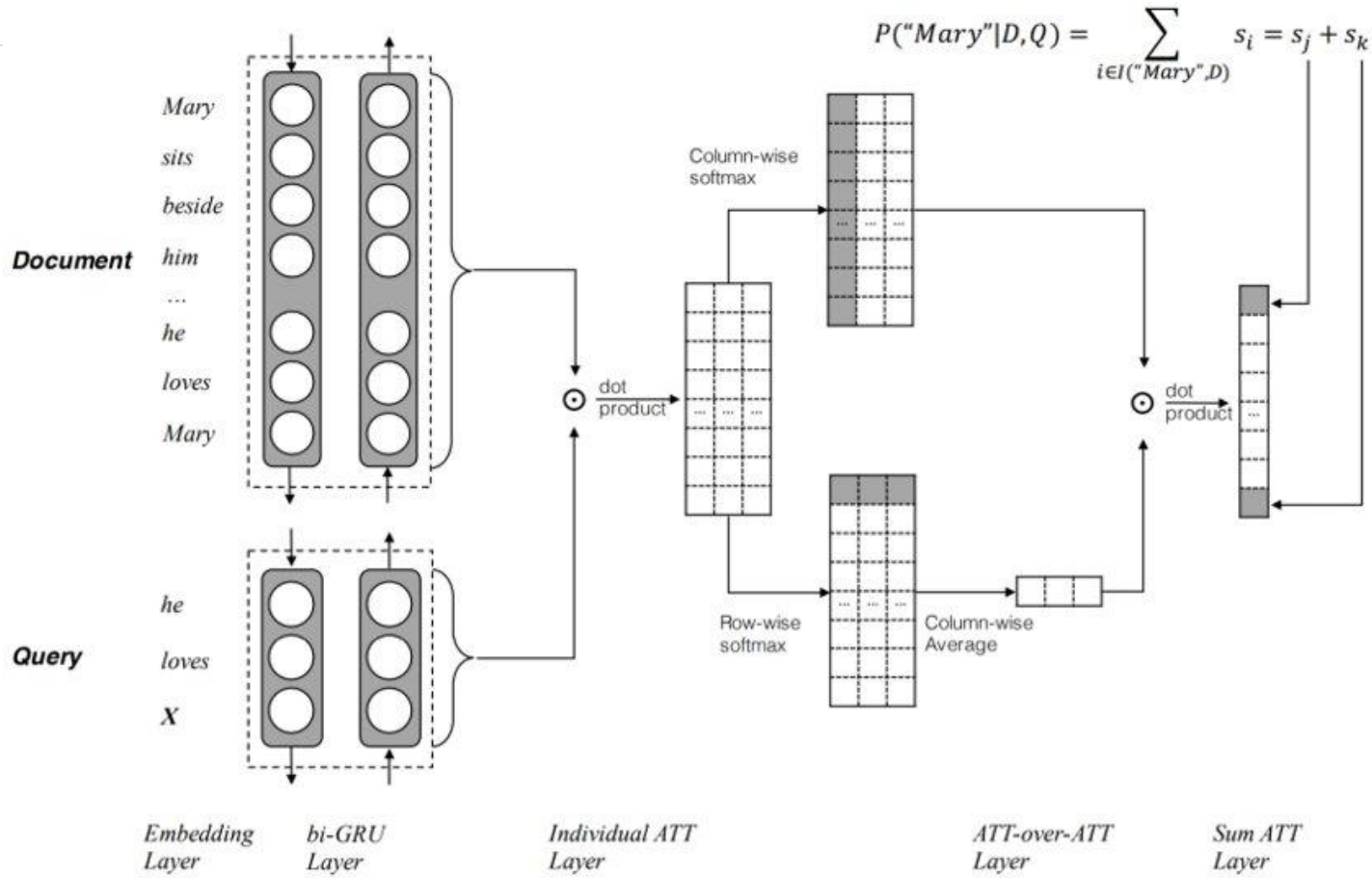
ERNIE 2.0 is a continual pre-training framework. Continual learning aims to train the model with several tasks in sequence so that it remembers the previously learned tasks when learning the new ones.



...besides co-occurrence of words, the researchers at Baidu believe that there are other valuable lexical, syntactic and semantic information in training corpora. For example, named entities, such as names, locations and organisations, could contain conceptual information.

Fine-Tune ERNIE 2.0 for Text Classification

Attention-over-Attention Reader (AoA) for Q&A NLP task



Attention-over-Attention Neural Networks for Reading Comprehension

Previous works regarding attention:
- attention used to extract relevant information from the context by summarizing it into a fixed-size vector
- attention at the current time step - a function of the attended vector at the previous time step
- usually uni-directional, wherein the query attends on the context paragraph

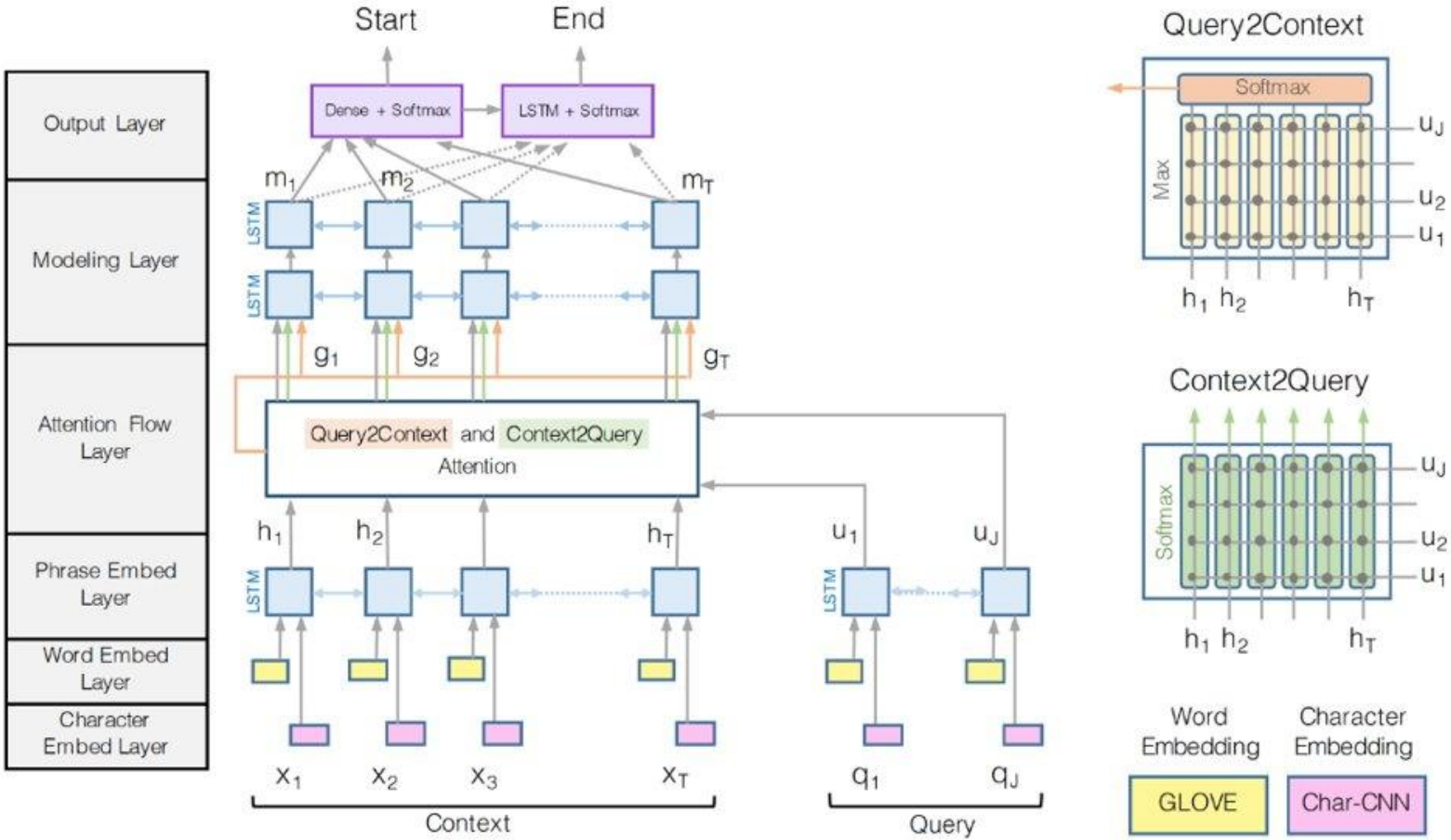
BiDAF new contribution:
Bi-directional Attention Flow (BiDAF) network is a multi-stage hierarchical process that represents context at different levels of granularity and uses a bi-directional attention flow mechanism to achieve a query-aware context representation without early summarization.

BIDAF includes **character-level**, **word-level**, and **contextual embeddings**, and uses **bi-directional attention flow** to obtain a **query-aware context representation**

The proposed attention layer:

- **No early summarization** = attention is computed for every time step, and the attended vector at each time step, along with the representations from previous layers, is allowed to flow through to the subsequent modeling layer
- **memory-less attention mechanism** = the attention at each time step is a function of only the query and the context paragraph at the current time step and does not directly depend on the attention at the previous time step
- Use **attention mechanisms in both directions**, **query-to-context** and **context-to-query**, which provide complimentary information to each other

Bi-Directional Attention Flow (BiDAF) for Q&A



Bi-Directional Attention Flow for Machine Comprehension

R-NET: Machine Reading Comprehension with Self-Matching Networks

At an intuitive level, R-Net performs reading comprehension in a way that is pretty similar to how a you or I would: by 'reading' (*applying RNN*) the text multiple times (3, to be exact), and 'fine-tuning' (*using Attention*) the vectorial representations of the terms better and better in each iteration.

a) use a BiRNN over standard word embeddings to get more meaningful representation of words in the surrounding context
Ex: "*May happen*" & "*the fourth of May*"

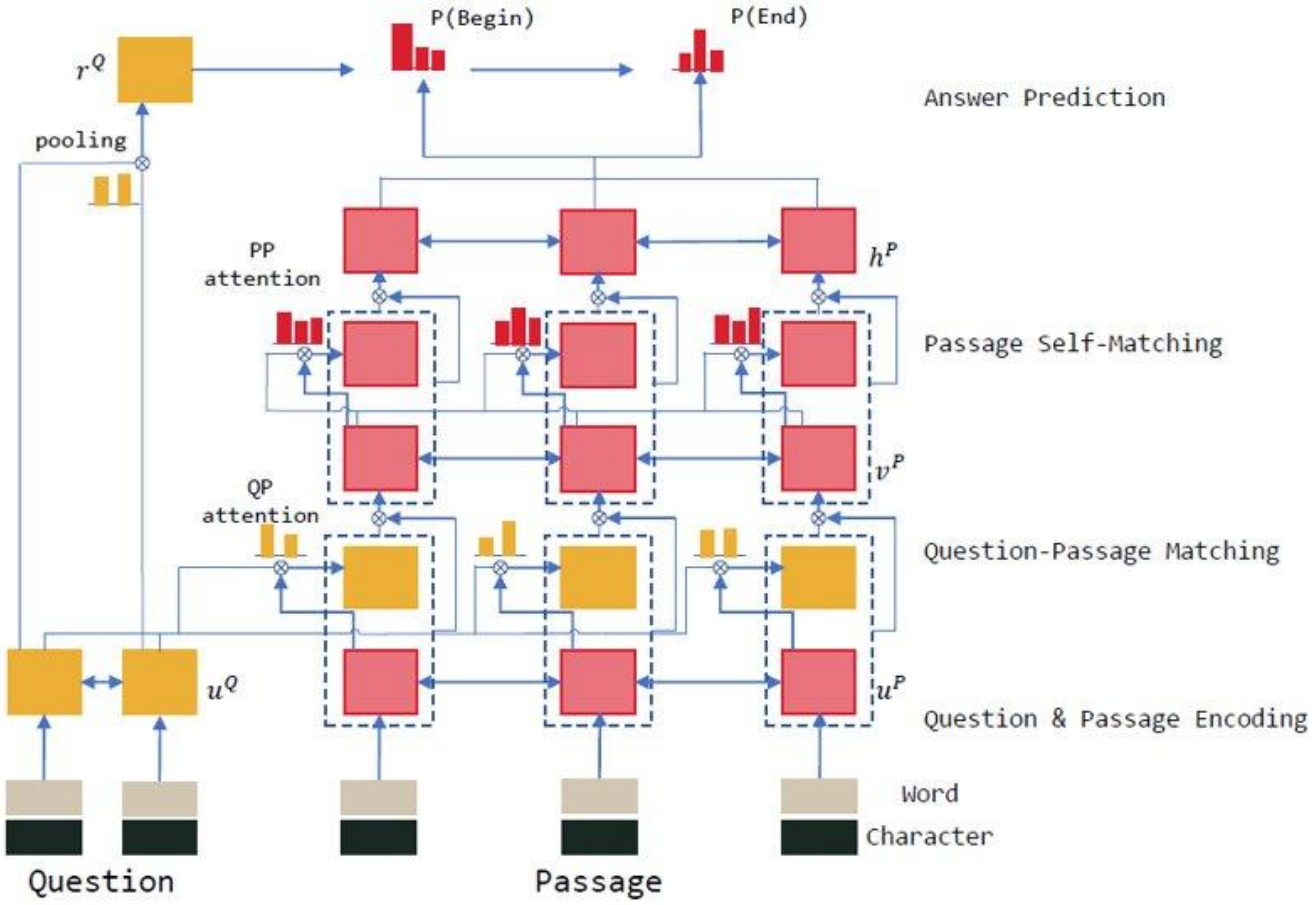
b) Second reading: Question-based analysis with ("***Gated Attention-based RNNs***")
In the second pass, the network tunes word-representations from the passage in the context of the question itself.
Ex: the network adjusts the vector for "*making*" to get it closer to "*abilities*" in a semantic sense:
*"…had a talent for **making** home craft tools, mechanical appliances, and the ability to memorize Serbian epic poems. Đuka had never received a formal education…"*
*"What were Tesla's mother's special **abilities**?"*

*c)*Third reading: Self-aware, complete passage understanding
('**Self-Matched Attention**' *to compare far-off terms in the same passage*)
get a bird's eye view of the entire passage, to pinpoint those sections that actually help in answering the question

*Tesla's mother, Đuka Tesla (née Mandić), whose father was also an Orthodox priest,:10 had a talent for making home craft tools, mechanical appliances, and the **ability** to memorize Serbian epic poems. Đuka had never received a formal education. Nikola credited his eidetic memory and creative **abilities** to his mother's genetics and influence.*
d) Final step: Marking the answer (Pointer Networks)

Microsoft and Alibaba AI programs beat humans in Stanford reading comprehension test for 1st time

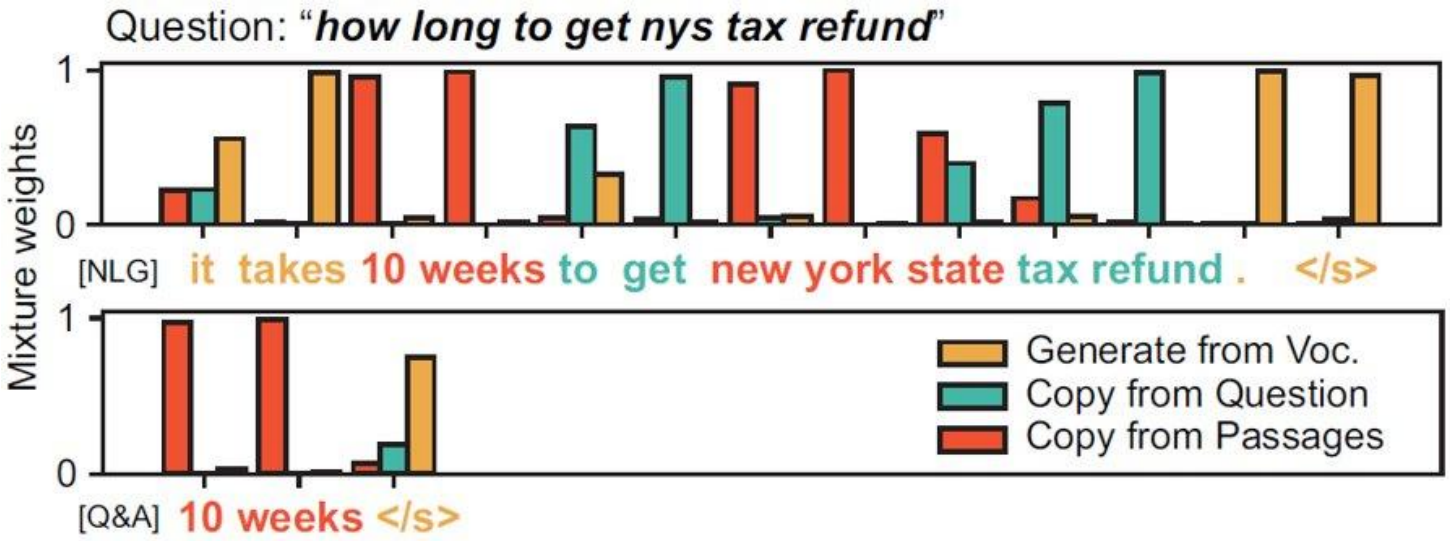
R-NET: Machine Reading Comprehension with Self-Matching Networks



Open-domain question answering with DeepPavlov (Demo), Understanding R-Net: Microsoft's 'superhuman' reading AI

First, unlike most studies on RC that have focused on extracting an answer span from the provided passages, our model instead focuses on generating a summary from the question and multiple passages.

Second, whereas previous studies built a specific model for each answer style because of the difficulty of acquiring one general model, our approach learns multi-style answers within a model to improve the NLG capability for all styles involved.

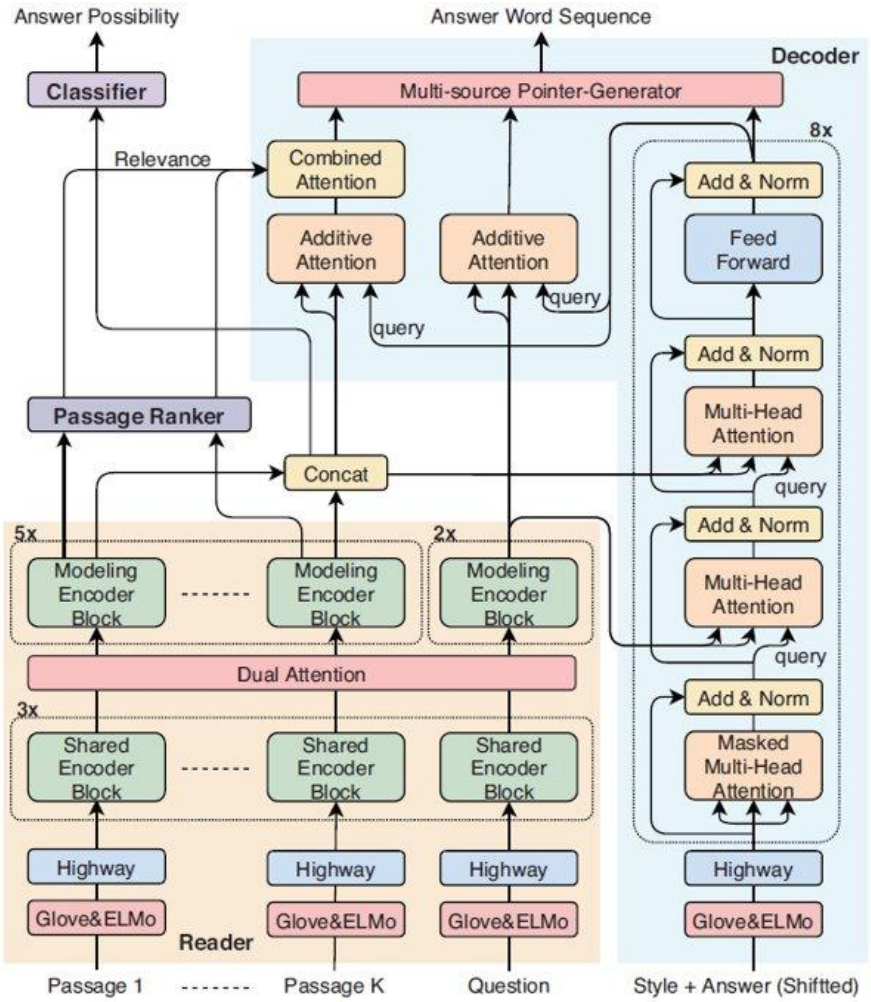


Multi-style Generative Reading Comprehension, NarrativeQA leaderboard

Masque model architecture

Highlights:

- multi-style learning
- jointly trained with the passage ranker and answer possibility classifier
- Transformer-based pointer-generator



Multi-style Generative Reading Comprehension

Masque model architecture – multi-source pointer-generator

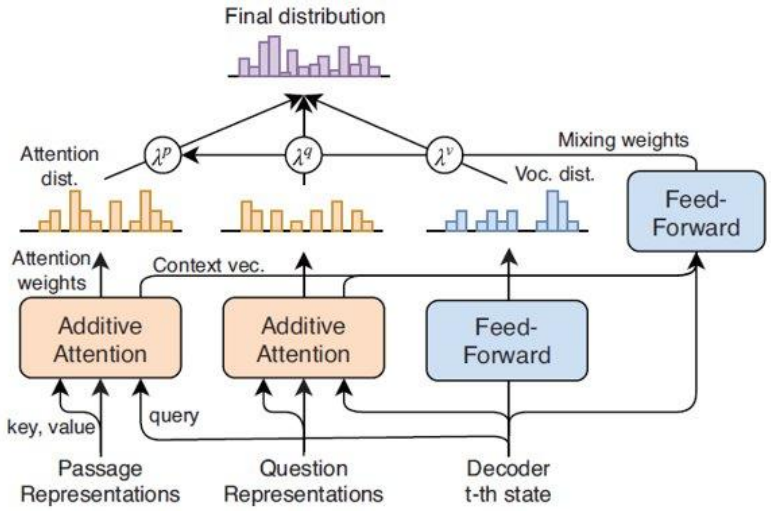


Figure 3: Multi-source pointer-generator mechanism. For each decoding step t, mixture weights $\lambda^v, \lambda^q, \lambda^p$ for the probability of generating words from the vocabulary and copying words from the question and the passages are calculated. The three distributions are weighted and summed to obtain the final distribution.

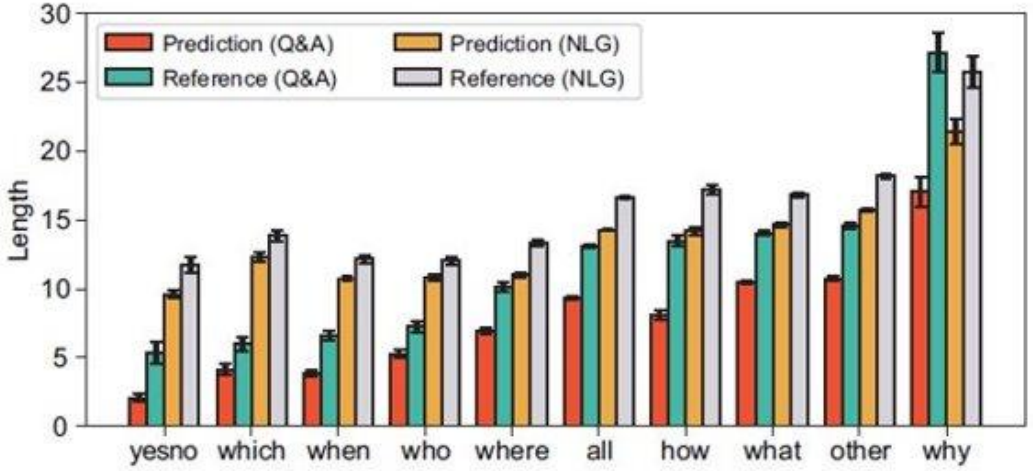


Figure 5: Lengths of answers generated by Masque broken down by the answer style and query type on the NLG dev. set. The error bars indicate standard errors.

Model	NLG		Q&A	
	R-L	B-1	R-L	B-1
BiDAF[a]	16.91	9.30	23.96	10.64
Deep Cascade QA[b]	35.14	37.35	52.01	**54.64**
S-Net+CES2S[c]	45.04	40.62	44.96	46.36
BERT+Multi-PGNet[d]	47.37	45.09	48.14	52.03
Selector+CCG[e]	47.39	45.26	50.63	52.03
VNET[f]	48.37	46.75	51.63	54.37
Masque (NLG; single)	49.19	49.63	48.42	48.68
Masque (NLG; ensemble)	**49.61**	**50.13**	48.92	48.75
Masque (Q&A; single)	25.66	36.62	50.93	42.37
Masque (Q&A; ensemble)	28.53	39.87	**52.20**	43.77
Human Performance	63.21	53.03	53.87	48.50

Table 2: Performance of our and competing models on the MS MARCO V2 leaderboard (4 March 2019). [a]Seo et al. (2017); [b]Yan et al. (2019); [c]Shao (unpublished), a variant of Tan et al. (2018); [d]Li (unpublished), a model using Devlin et al. (2018) and See et al. (2017); [e]Qian (unpublished); [f]Wu et al. (2018). Whether the competing models are ensemble models or not is unreported.

Model	B-1	B-4	M	R-L
BiDAF[a]	33.72	15.53	15.38	36.30
DECAPROP[b]	42.00	23.42	23.42	40.07
MHPGM+NOIC[c]	43.63	21.07	19.03	44.16
ConZNet[d]	42.76	22.49	19.24	46.67
RMR+A2D[e]	50.4	26.5	N/A	53.3
Masque (NQA)	**54.11**	**30.43**	**26.13**	**59.87**
w/o multi-style learning	48.70	20.98	21.95	54.74
Masque (NLG)	39.14	18.11	24.62	50.09
Masque (NQA; valid.)[f]	52.78	28.72	25.38	58.94

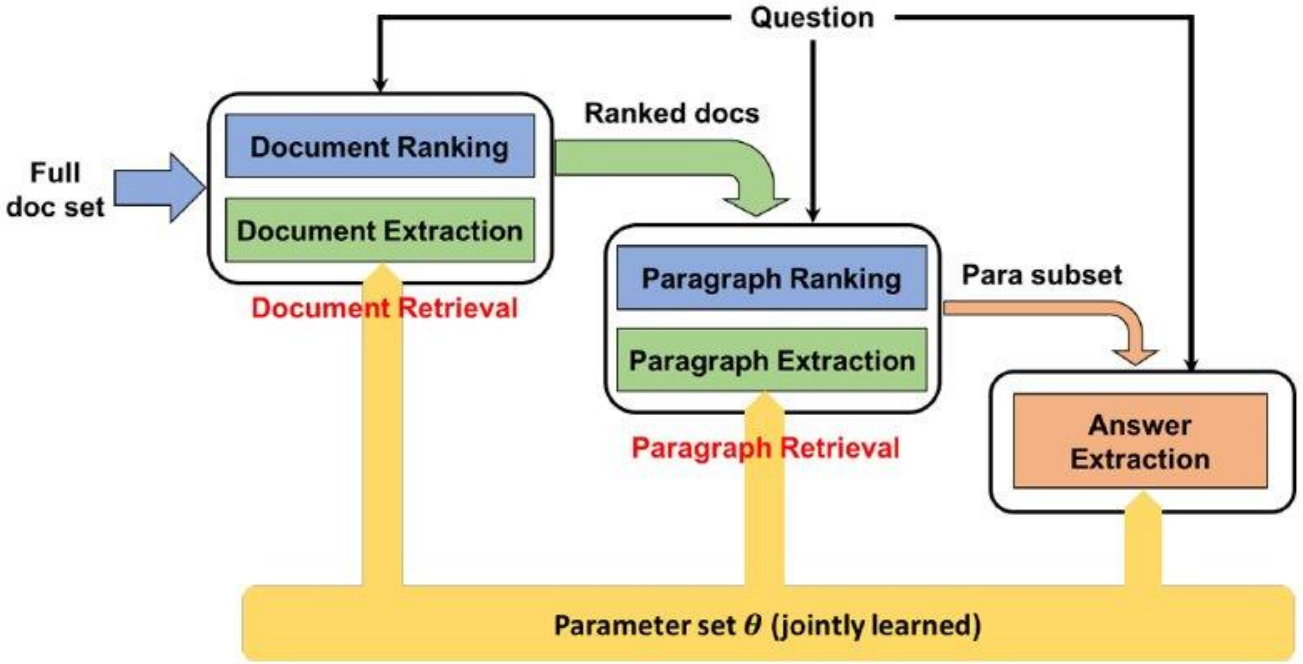
Table 5: Performance of our and competing models on the NarrativeQA test set. [a]Seo et al. (2017); [b]Tay et al. (2018); [c]Bauer et al. (2018); [d]Indurthi et al. (2018); [e]Hu et al. (2018). [f]Results on the NarrativeQA validation set.

(e) Question: does gameplay programmer need math skill
Relevant Passage: A good computer programmer is more of a problem solver and logical thinker than a math buff. And besides, the industry is peppered with many successful computer programmers who do not really know much about mathematics.
Reference Answer (Q&A): No
Prediction (Q&A): yes ✗
Reference Answers (NLG): No, a gameplay programmer doesn't need math skill. / No, gameplay programmer do not need an math skill.
Prediction (NLG): no , gameplay programmer does not need math skill . ✓

(c) Question: average height nba player
Relevant Passage: The average height of an NBA player is around 6 feet 7 inches. The tallest NBA player ever was Gheorghe Muresan, who was 7 feet 7 inches tall. In contrast, the shortest NBA player ever was Tyrone Muggsy Bogues, who was 5 feet 3 inches tall.
Reference Answer (Q&A): Around 6 feet 7 inches
Prediction (Q&A): 6 feet 7 inches ✓
Reference Answers (NLG): The average height of NBA players is around 6 feet, 7 inches. / The height of NBA player is around 6 feet 7 inches.
Prediction (NLG): the average height of an national basketball association player is 6 feet 7 inches . ✓

A deep cascade model for Q&A - MRC model (in production at Alibaba)

- The overall framework of our deep cascade model, which consists of the document retrieval, paragraph retrieval and answer extraction modules.
- balance between the effectiveness and efficiency



ALIBABA AI BEATS HUMANS IN READING-COMPREHENSION TEST,
A Deep Cascade Model for Multi-Document Reading Comprehension

[IASI AI] Machine reading comprehension over long narratives (from Facebook 2019)

We propose a **curriculum learning** (CL) based **Pointer-Generator framework** for reading/sampling over large documents...

The usage of the Pointer-Generator softens the requirement of having the answer within the context...

We train our models to be robust regardless of whatever is retrieved = learning to guess, largely motivated by how humans are able to extrapolate/guess even when given access to a small fragment of a film/story

Additionally, we propose a new **Introspective Alignment Layer** (IAL), which reasons over decomposed alignments using **block-based self-attention**. … (additional local reasoning over these enhanced alignment vectors)

We evaluate our proposed method on the NarrativeQA reading comprehension benchmark, achieving state-of-the-art performance, 10 times improvement in terms of BLEU-4 score over BiDAF
NarrativeQA = the reader must answer questions about stories by reading entire books or movie scripts.

Curriculum learning: based on two concepts of difficulty :
- the answer exists in the context (answerability),
-the second, depends on the size of retrieved documents (coherence and understandability).

Facebook aug 2019 - Simple and Effective Curriculum Pointer-Generator Networks for Reading Comprehension over Long Narratives

Other Transformer + BERT related papers

BERT for Joint Intent Classification and Slot Filling,
Baidu's ERNIE Tops Google's BERT in Chinese NLP Tasks,
BertQA - Attention on Steroids
XLM—Enhancing BERT for Cross-lingual Language Model
New Google Brain Optimizer Reduces BERT Pre-Training Time From Days to Minutes
Massive Multi-Task Learning with Snorkel MeTaL: Bringing More Supervision to Bear
GluonNLP 0.6: Closing the Gap in Reproducible Research with BERT
The Evolved Transformer - Enhancing Transformer with Neural Architecture Search
Transformer-XL – Combining Transformers and RNNs Into a State-of-the-art Language Model
Advances in NLP in 2017 (Parallel Decoder for Neural Machine Translation)
Fast Transformer Decoding: One Write-Head is All You Need

Other Transformer + BERT related tutorials

Deconstructing BERT: Distilling 6 Patterns from 100 Million Parameters
BERT Technology introduced in 3-minutes
Multi-label Text Classification using BERT – The Mighty Transformer
Attention is all you need's review
Dissecting BERT Part 1: Understanding the Transformer
The Annotated Transformer
How to code The Transformer in PyTorch
When Multi-Task Learning meet with BERT
How BERT leverage attention mechanism and transformer to learn word contextual relations
How the Embedding Layers in BERT Were Implemented
NLP: Contextualized word embeddings from BERT
BERT in Keras with Tensorflow hub
The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)
How Transformers Work
Google BERT—Pre Training and Fine Tuning for NLP Tasks
Evolution of Representations in the Transformer
How BERT leverage attention mechanism and transformer to learn word contextual relations

Other Transformer + BERT related tutorials

The Illustrated Transformer
Dissecting BERT Part 1: Understanding the Transformer
Transformer Architecture: Attention Is All You Need
Paper Dissected: “Attention is All You Need” Explained
A Gentle Introduction to Calculating the BLEU Score for Text in Python
What's New in XLNet?
CMU & Google XLNet Tops BERT; Achieves SOTA Results on 18 NLP Tasks
Understand how the XLNet outperforms BERT in Language Modelling
Understanding XLNet
NLP Gets A Surprise Addition As XLNet Outperforms BERT
NAACL ’19 Notes: Practical Insights for Natural Language Processing Applications
Painless Fine-Tuning of BERT in Pytorch
NLP — BERT & Transformer
NLP Breakfast 2: The Rise of Language Models

Understanding building blocks of ULMFIT

Efficient multi-lingual language model fine-tuning

Introducing state of the art text classification with universal language models

STRUCTURED NEURAL SUMMARIZATION (Microsoft develops flexible AI system that can summarize the news) (Gated Graph Neural Networks + BiLSTM:)

Other Commonsense Reasoning

- Visual Commonsense Reasoning (VCR)
- Compositional Attention Networks for Machine Reasoning
- From Deep Learning of Disentangled Representations to Higher-level Cognition

[IASI AI]

Thank You!

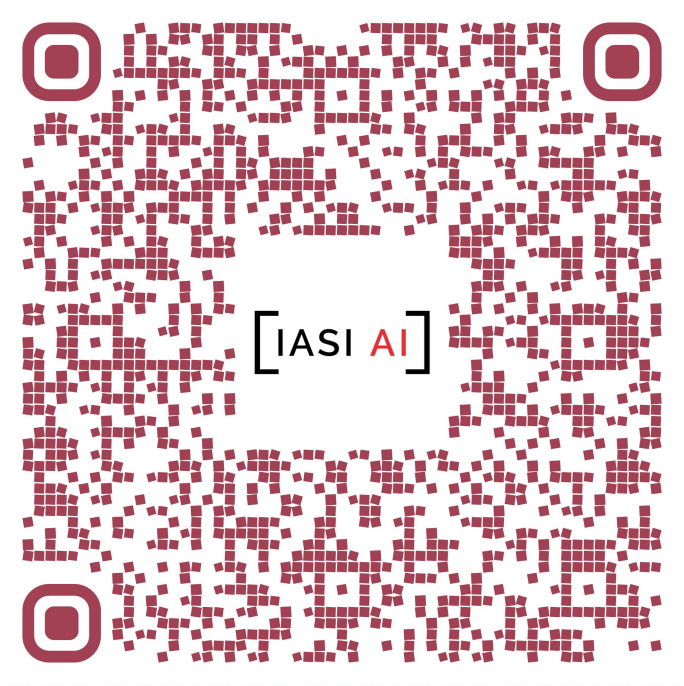
iasi.ai

fb.me/AI.Iasi/

meetup.com/IASI-AI/

[IASI AI]

We ❤ Feedback



[IASI AI]

Community partners