Gridspace

GRIDSPACE IAP LECTURE 6
LARGE LANGUAGE MODELS & INSTRUCTGPT

January 20, 2023

TODAY'S ROADMAP

- Modern NLP
- Language Models
- Sequence Models
- Attention
- Transformers
- BERT & GPT
- InstructGPT

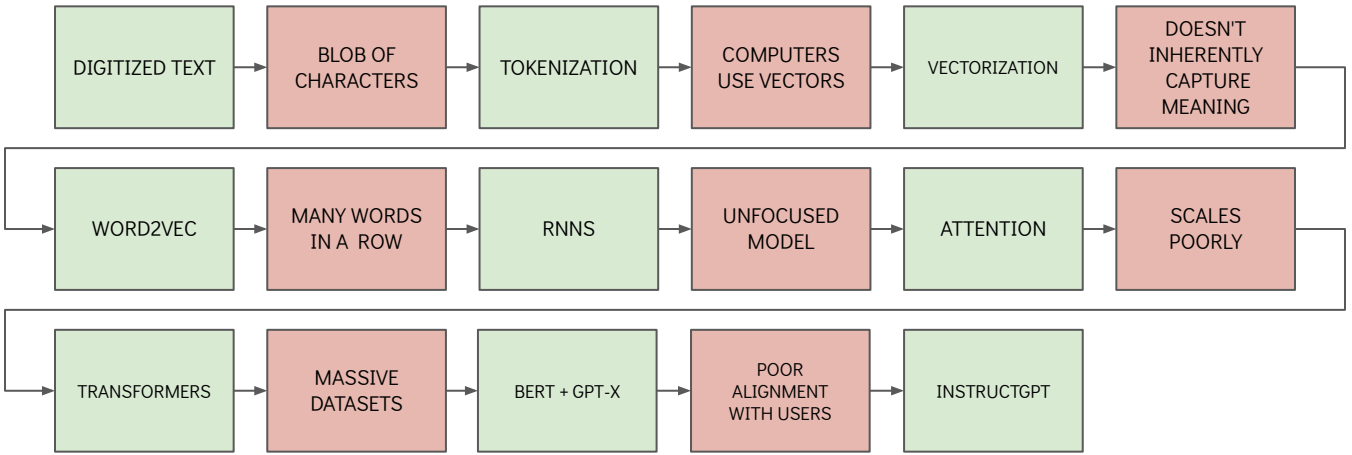
MODERN NLP

HOW DID WE GET HERE?

Longer sequences of words

Larger datasets

Optimized for modern hardware



LANGUAGE MODELS

once upon a _____

four score ____

twas ____

COMPLETE THE PROMPT:

What are you doing this weekend?

COMPLETE THE PROMPT:

Why can't dogs drive cars?

N-grams

Unigram - "i"

Bigram - "i am"

Trigram - "i am waiting"

4-gram - "i am waiting for"

Statistical Language Models

P("i")

P("am"|"i")

P("waiting"|"i am")

P("for"|"i am waiting")

Markov Models

The probability of the next state is dependent only upon the current state.

Probabilities of all new states sums to one.

Approximating with maximum likelihood estimation (MLE)

P("i") = Count("i")/Count("tokens")

Approximating with maximum likelihood estimation (MLE)

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)}$$

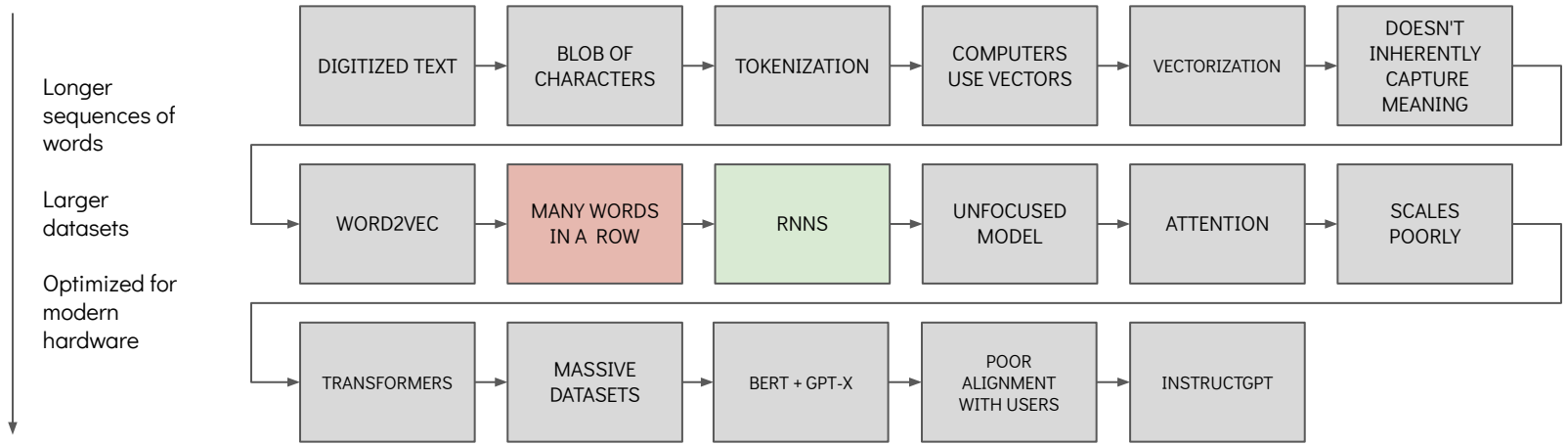
$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

Approximating with maximum likelihood estimation (MLE)

$$P(\text{"i am"}) = \text{Count("i am")}/\text{Count("i")}$$

SEQUENCE MODELS

HOW DID WE GET HERE?



symmetries in data =
opportunities for simplification

simplification = more expressive models per parameter

in a long sequence of words we can shift by a word
a long sequence of words we can shift by a word and
long sequence of words we can shift by a word and retain
sequence of words we can shift by a word and retain the
of words we can shift by a word and retain the same
words we can shift by a word and retain the same problem

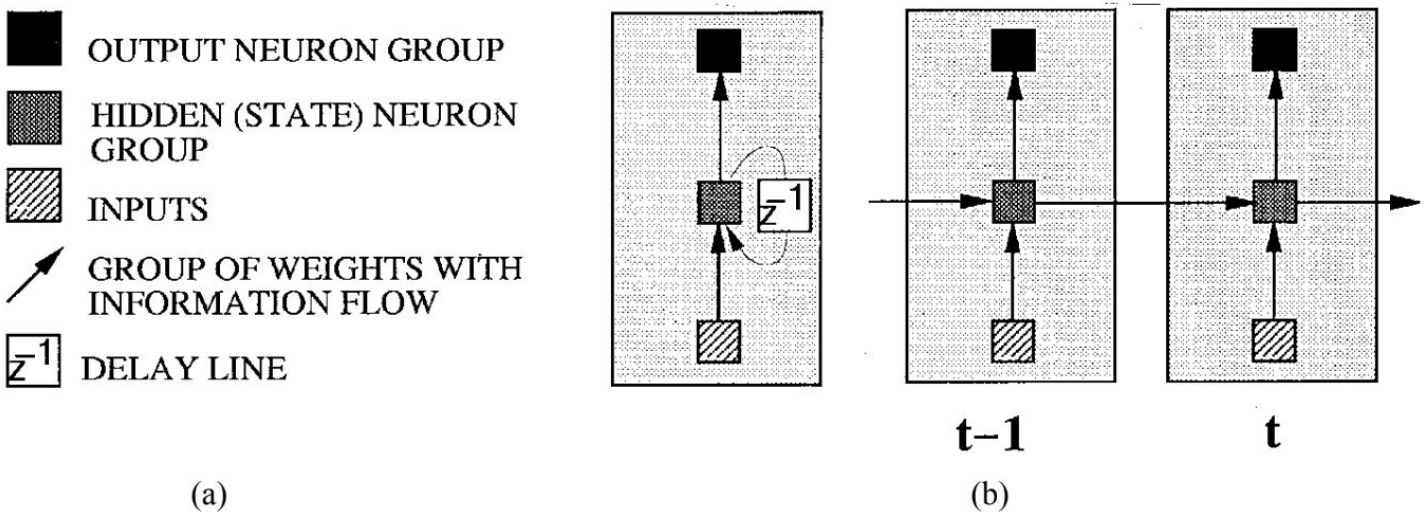
Translation symmetry allows reuse of parameters.

Reuse of parameters allows smaller models that can be reused word-by-word.

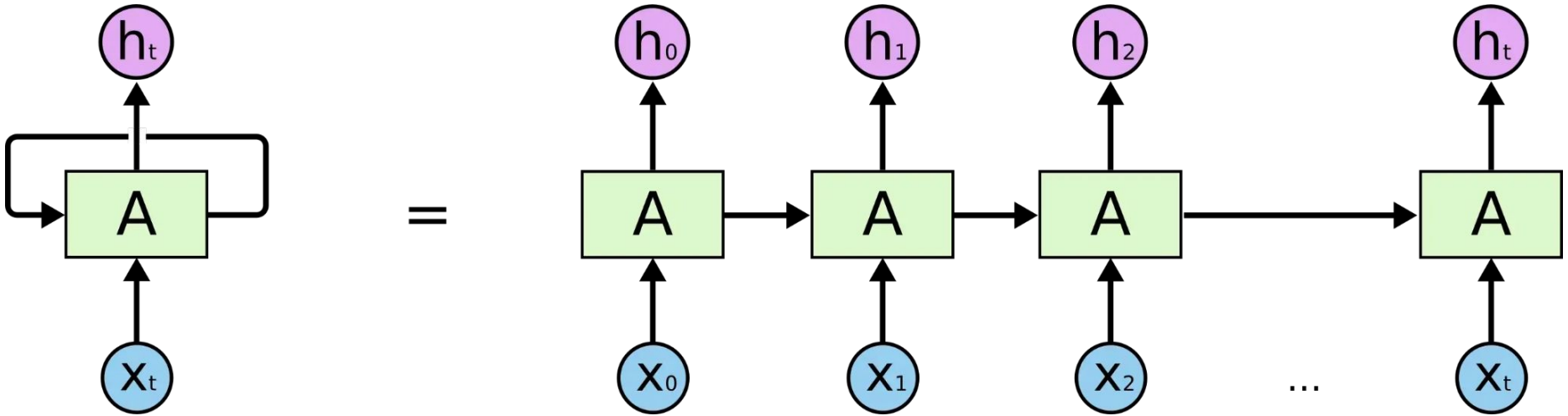
More parameters to scale up our model.

RNNS

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 45, NO. 11, NOVEMBER 1997



RNNS



LSTMS

LONG SHORT-TERM MEMORY

Neural Computation 9(8):1735–1780, 1997

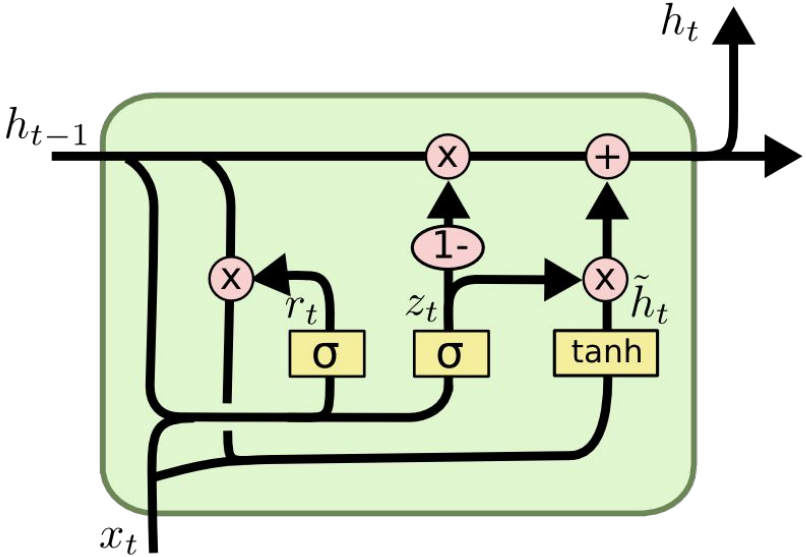
Sepp Hochreiter
Fakultät für Informatik
Technische Universität München
80290 München, Germany
hochreit@informatik.tu-muenchen.de
http://www7.informatik.tu-muenchen.de/~hochreit

Jürgen Schmidhuber
IDSIA
Corso Elvezia 36
6900 Lugano, Switzerland
juergen@idsia.ch
http://www.idsia.ch/~juergen

Abstract

Learning to store information over extended time intervals via recurrent backpropagation takes a very long time, mostly due to insufficient, decaying error back flow. We briefly review Hochreiter's 1991 analysis of this problem, then address it by introducing a novel, efficient, gradient-based method called "Long Short-Term Memory" (LSTM). Truncating the gradient where this does not do harm, LSTM can learn to bridge minimal time lags in excess of 1000 discrete time steps by enforcing *constant* error flow through "constant error carrousels" within special units. Multiplicative gate units learn to open and close access to the constant error flow. LSTM is local in space and time; its computational complexity per time step and weight is $O(1)$. Our experiments with artificial data involve local, distributed, real-valued, and noisy pattern representations. In comparisons with RTRL, BPTT, Recurrent Cascade-Correlation, Elman nets, and Neural Sequence Chunking, LSTM leads to many more successful runs, and learns much faster. LSTM also solves complex, artificial long time lag tasks that have never been solved by previous recurrent network algorithms.

LSTMS



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

SEQ2SEQ

Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

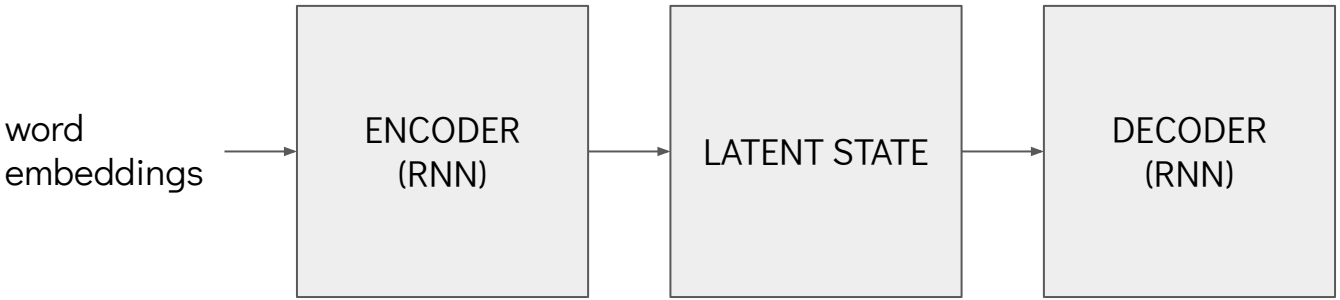
Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

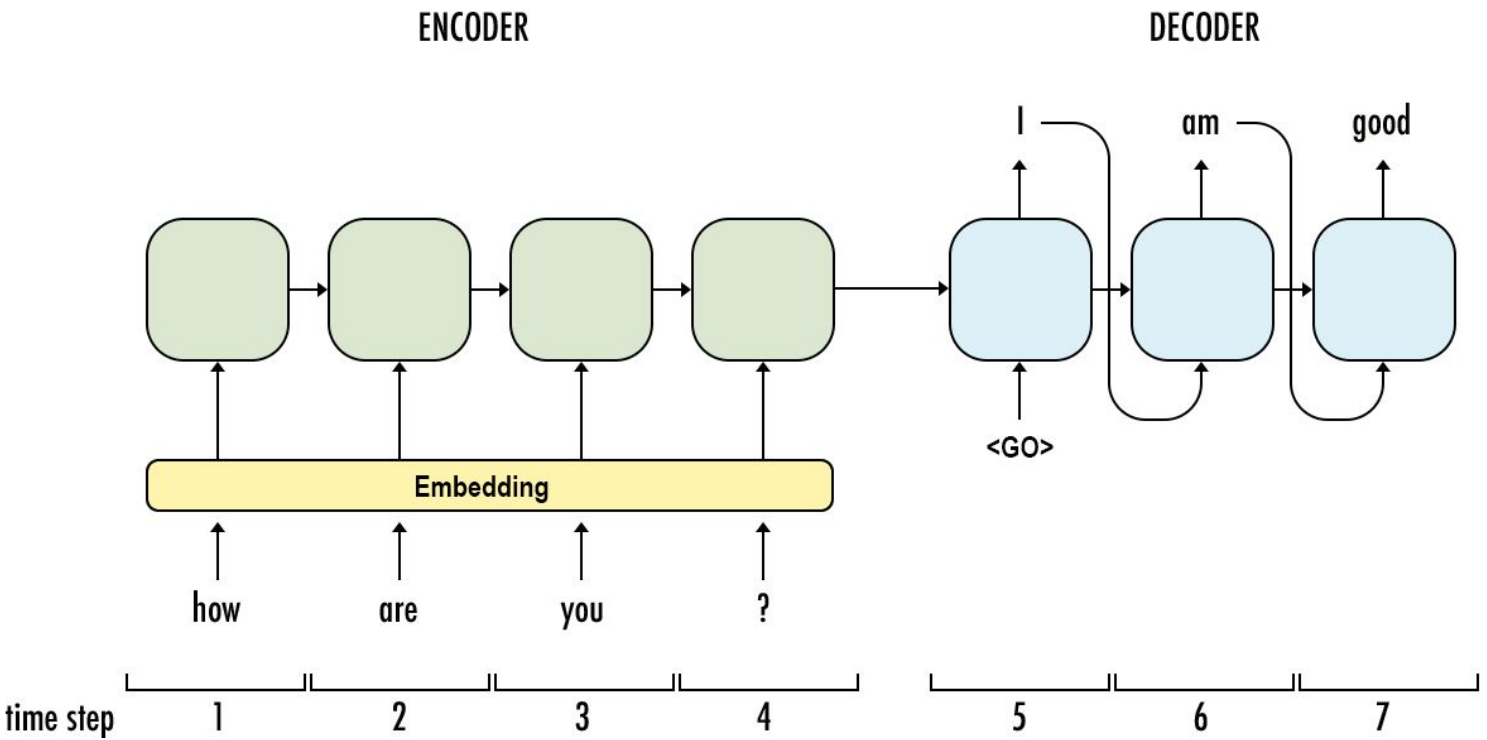
Abstract

Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our main result is that on an English to French translation task from the WMT-14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU score was penalized on out-of-vocabulary words. Additionally, the LSTM did not have difficulty on long sentences. For comparison, a phrase-based SMT system achieves a BLEU score of 33.3 on the same dataset. When we used the LSTM to rerank the 1000 hypotheses produced by the aforementioned SMT system, its BLEU score increases to 36.5, which is close to the previous state of the art. The LSTM also learned sensible phrase and sentence representations that are sensitive to word order and are relatively invariant to the active and the passive voice. Finally, we found that reversing the order of the words in all source sentences (but not target sentences) improved the LSTM's performance markedly, because doing so introduced many short term dependencies between the source and the target sentence which made the optimization problem easier.

SEQ2SEQ

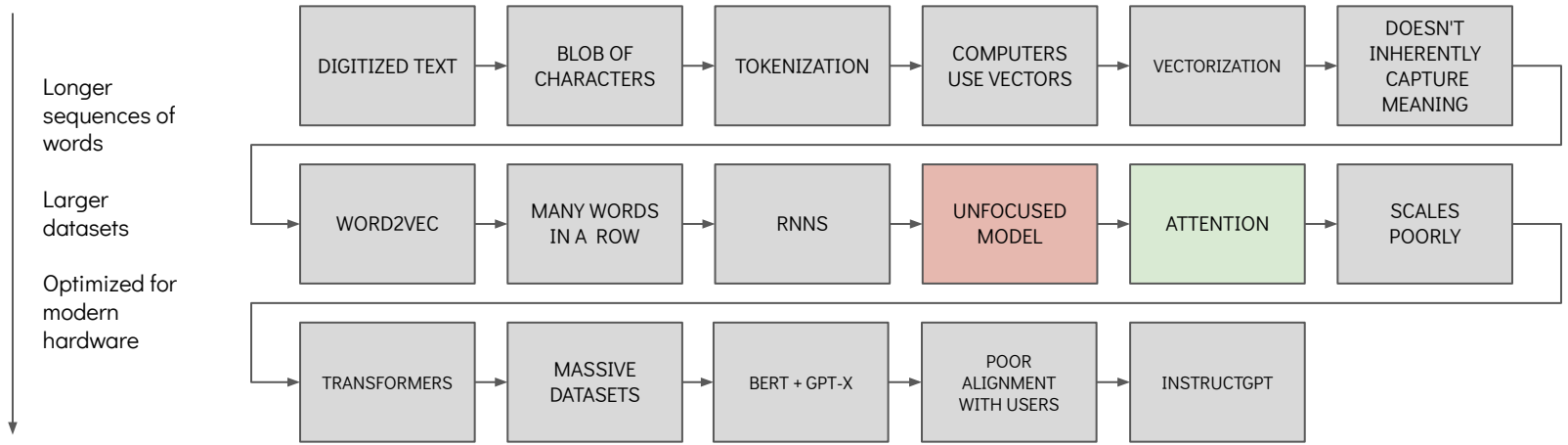


SEQ2SEQ



ATTENTION

HOW DID WE GET HERE?



ATTENTION

Published as a conference paper at ICLR 2015

NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

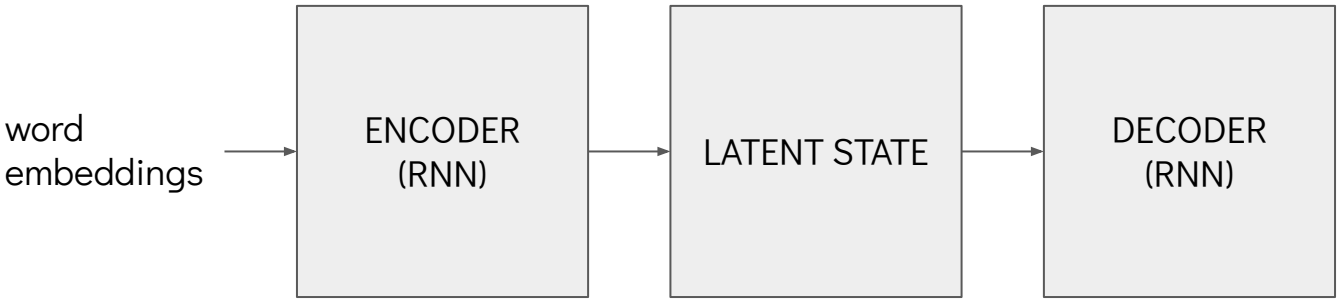
KyungHyun Cho **Yoshua Bengio***
Université de Montréal

ABSTRACT

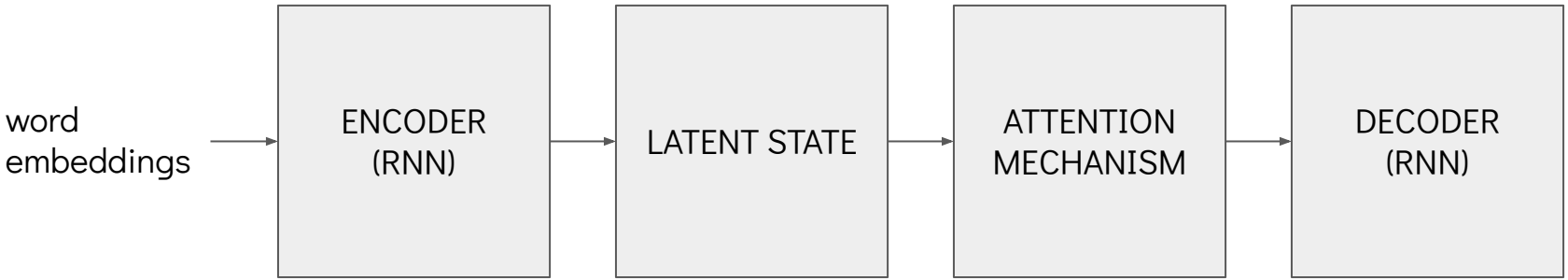
Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder–decoders and encode a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder–decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition.

1 INTRODUCTION

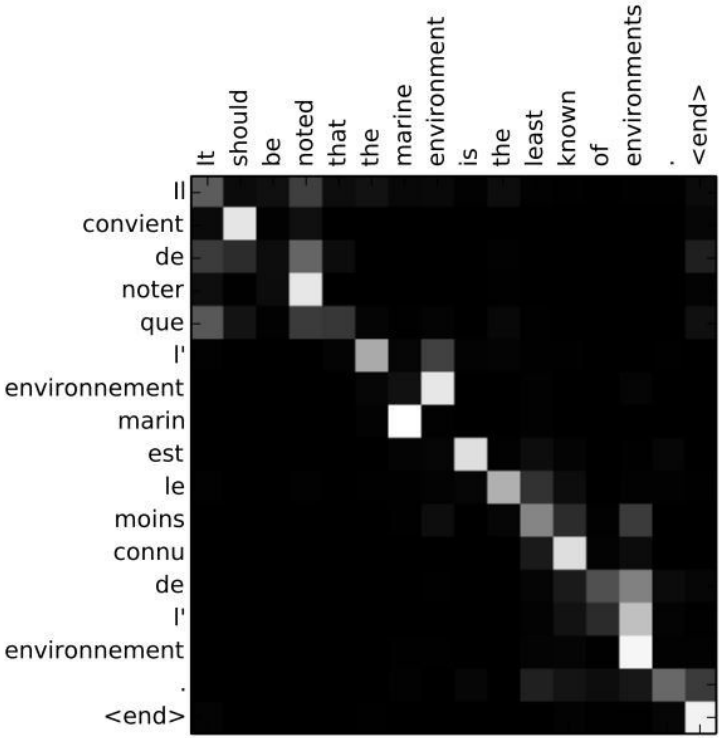
PREVIOUSLY...



ATTENTION



ATTENTION



TRANSFORMERS

TRANSFORMERS (SELF-ATTENTION)

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

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

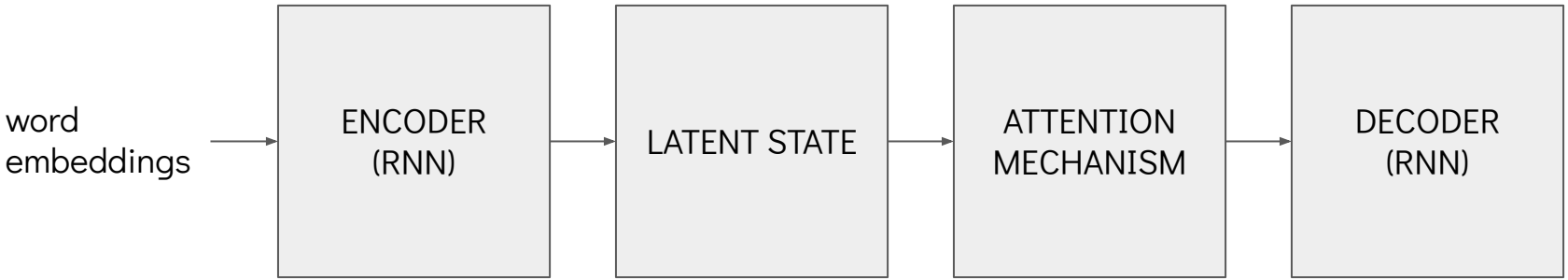
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

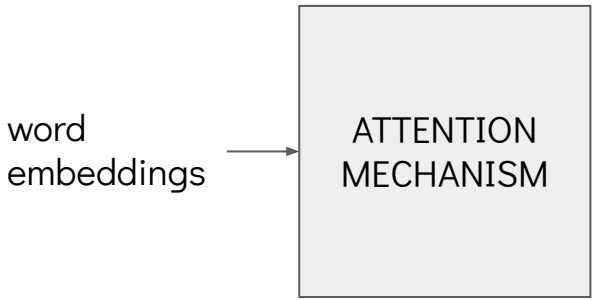
Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

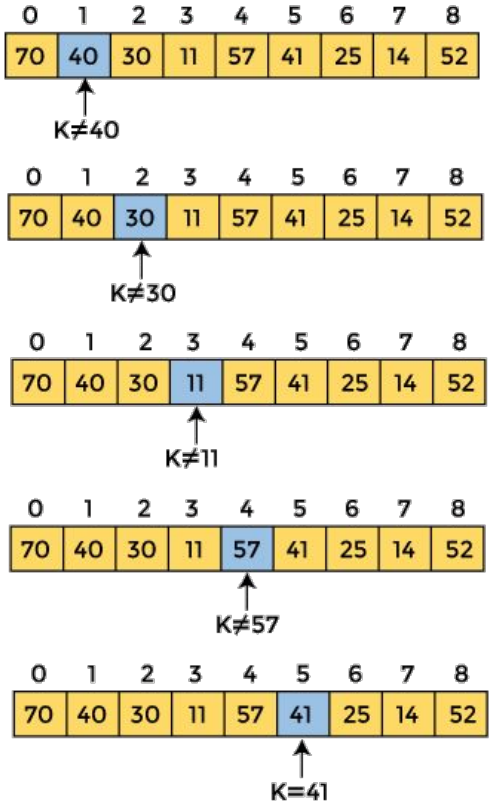
PREVIOUSLY...



TRANSFORMERS

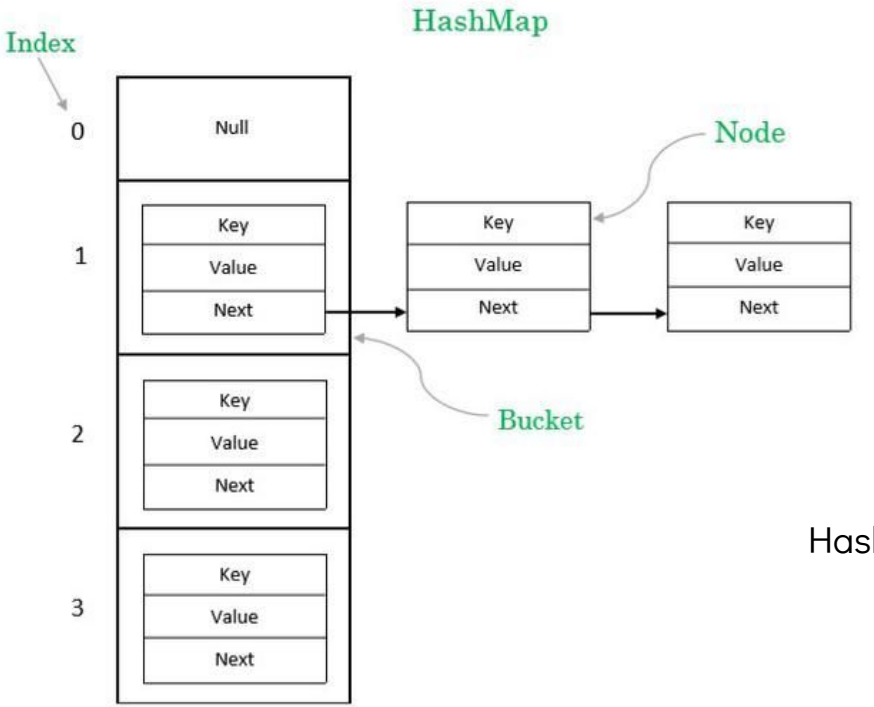


TRANSFORMERS



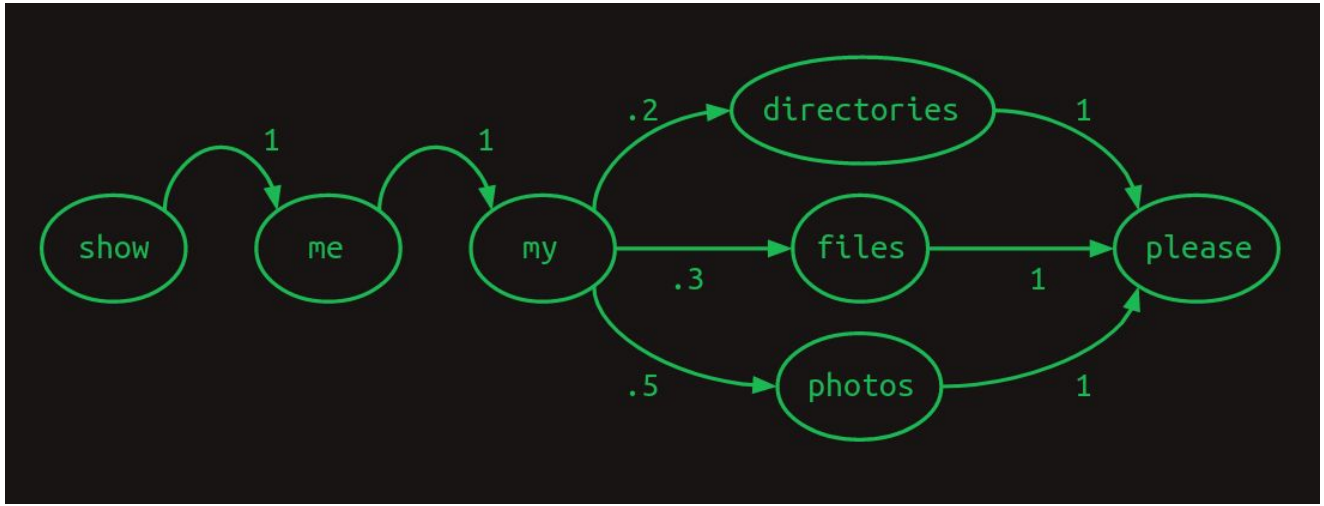
Linear Search O(n)

TRANSFORMERS



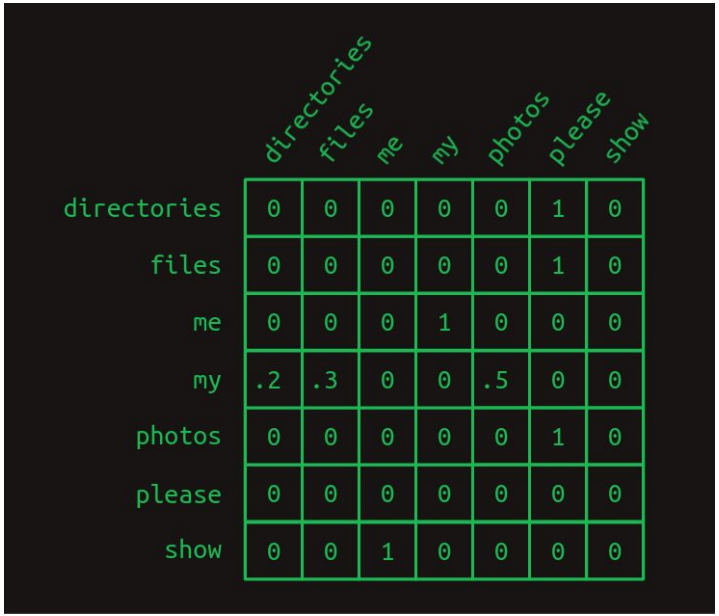
Hashmap O(1)

TRANSFORMERS



Language Modelling
as a lookup operation

TRANSFORMERS



	directories	files	me	my	photos	please	show
directories	0	0	0	0	0	1	0
files	0	0	0	0	0	1	0
me	0	0	0	1	0	0	0
my	.2	.3	0	0	.5	0	0
photos	0	0	0	0	0	1	0
please	0	0	0	0	0	0	0
show	0	0	1	0	0	0	0

Language Modelling
as a lookup operation

TRANSFORMERS

We need the lookup table to be differentiable so we can do calculus on it (and therefore implement backpropagation).

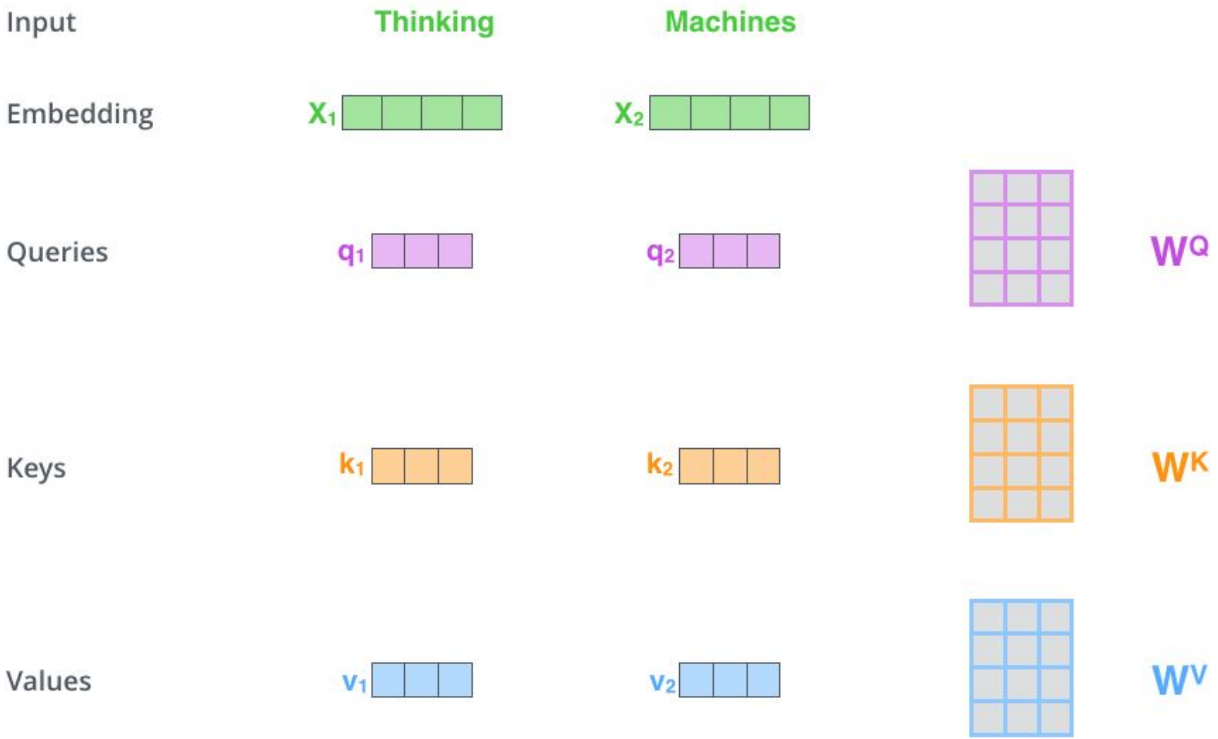
Rebuild key-value lookup as a vector matrix operation (our usual trick).

TRANSFORMERS

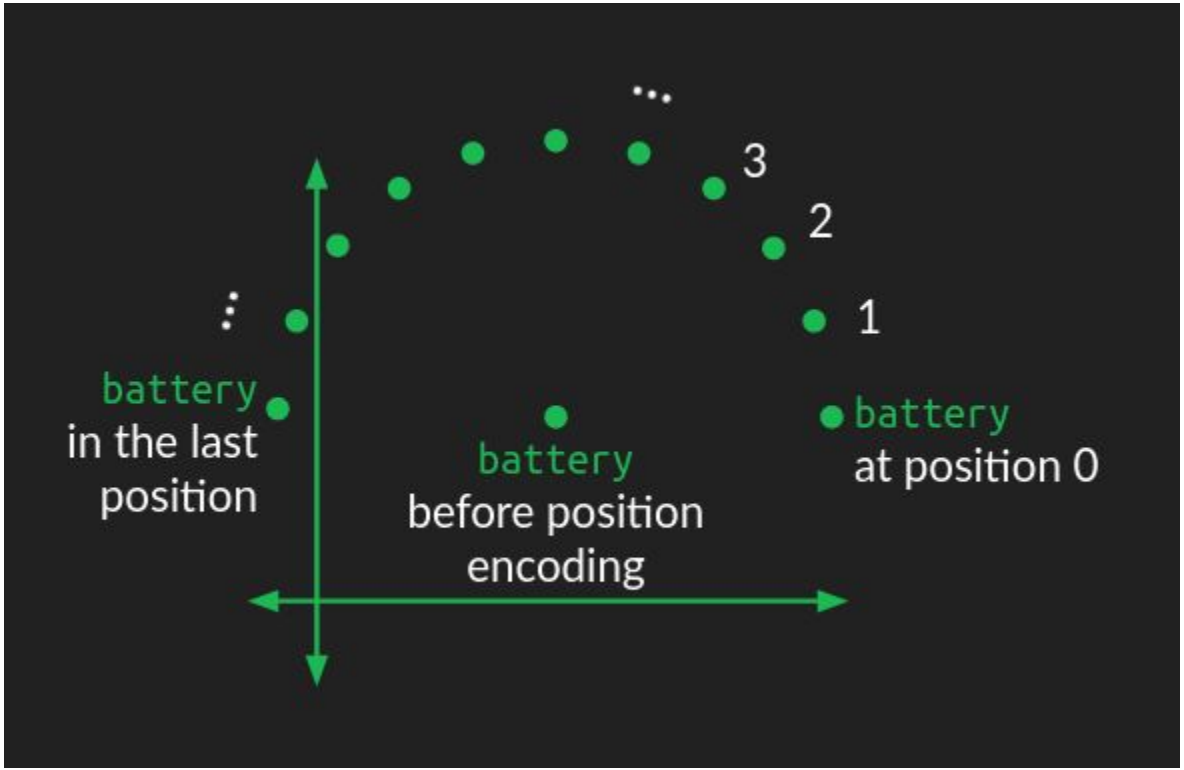
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Three matrices:
Q - The "query"
K - The "key"
V - The "value"
d_k - The dimension of key vectors

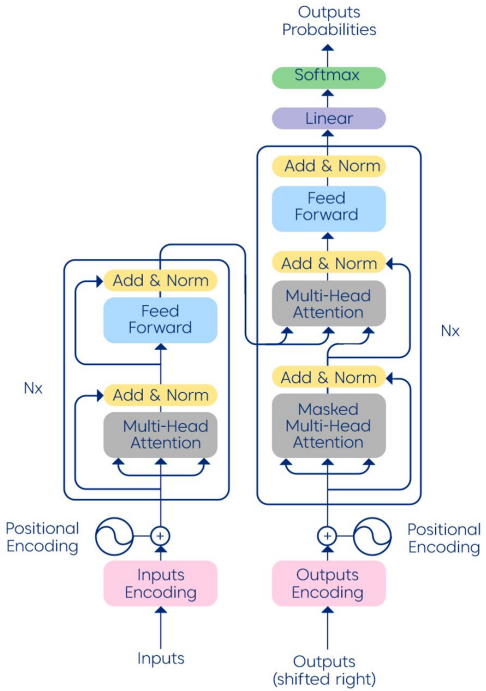
TRANSFORMERS



TRANSFORMERS

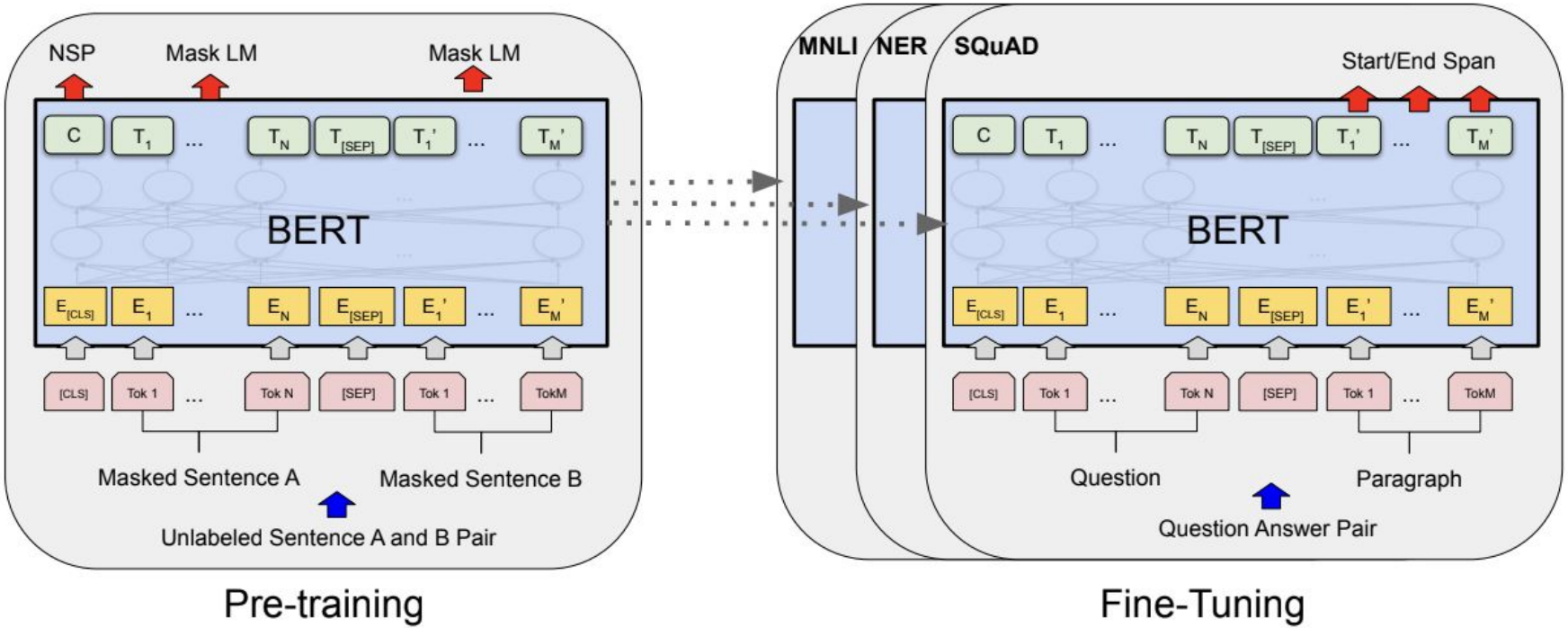


TRANSFORMERS (SELF-ATTENTION)

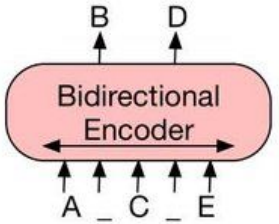


BERT & GPT

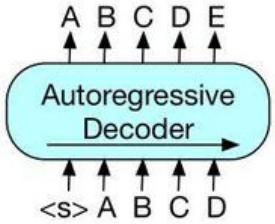
BERT



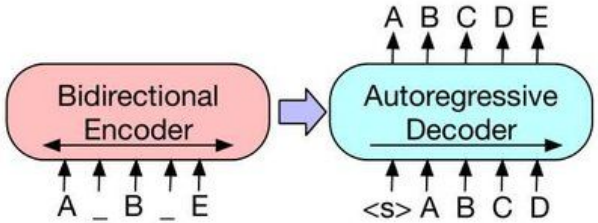
GPT



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.



(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with a mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

INSTRUCTGPT

Training language models to follow instructions with human feedback

Long Ouyang* Jeff Wu* Xu Jiang* Diogo Almeida* Carroll L. Wainwright*

Pamela Mishkin* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray

John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens

Amanda Askell† Peter Welinder Paul Christiano*†

Jan Leike* Ryan Lowe*

OpenAI

Abstract

Making language models bigger does not inherently make them better at following a user's intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not *aligned* with their users. In this paper, we show an avenue for aligning language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts submitted through the OpenAI API, we collect a dataset of labeler demonstrations of the desired model behavior, which we use to fine-tune GPT-3 using supervised learning. We then collect a dataset of rankings of model outputs, which we use to further fine-tune this supervised model using reinforcement learning from human feedback. We call the resulting models *InstructGPT*. In human evaluations on our prompt distribution, outputs from the 1.3B parameter InstructGPT model are preferred to outputs from the 175B GPT-3, despite having 100x fewer parameters. Moreover, InstructGPT models show improvements in truthfulness and reductions in toxic output generation while having minimal performance regressions on public NLP datasets. Even though InstructGPT still makes simple mistakes, our results show that fine-tuning with human feedback is a promising direction for aligning language models with human intent.

1 Introduction

Large language models (LMs) can be "prompted" to perform a range of natural language processing (NLP) tasks, given some examples of the task as input. However, these models often express unintended behaviors such as making up facts, generating biased or toxic text, or simply not following

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION

GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

InstructGPT

```
People went to the moon, and they took pictures of what they saw,
and sent them back to the earth so we could all see them.
```

Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

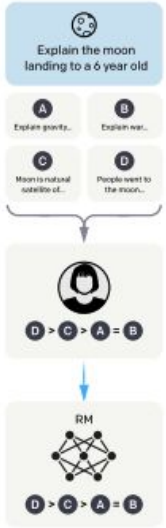
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

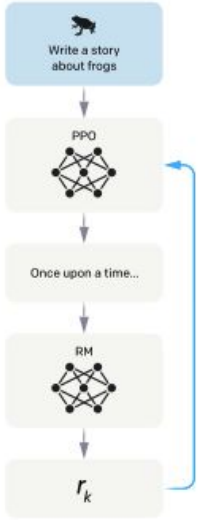
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Dataset

RealToxicity

GPT	0.233
Supervised Fine-Tuning	0.199
InstructGPT	**0.196**

Dataset

TruthfulQA

GPT	0.224
Supervised Fine-Tuning	0.206
InstructGPT	**0.413**

API Dataset

Hallucinations

GPT	0.414
Supervised Fine-Tuning	**0.078**
InstructGPT	0.172

API Dataset

Customer Assistant Appropriate

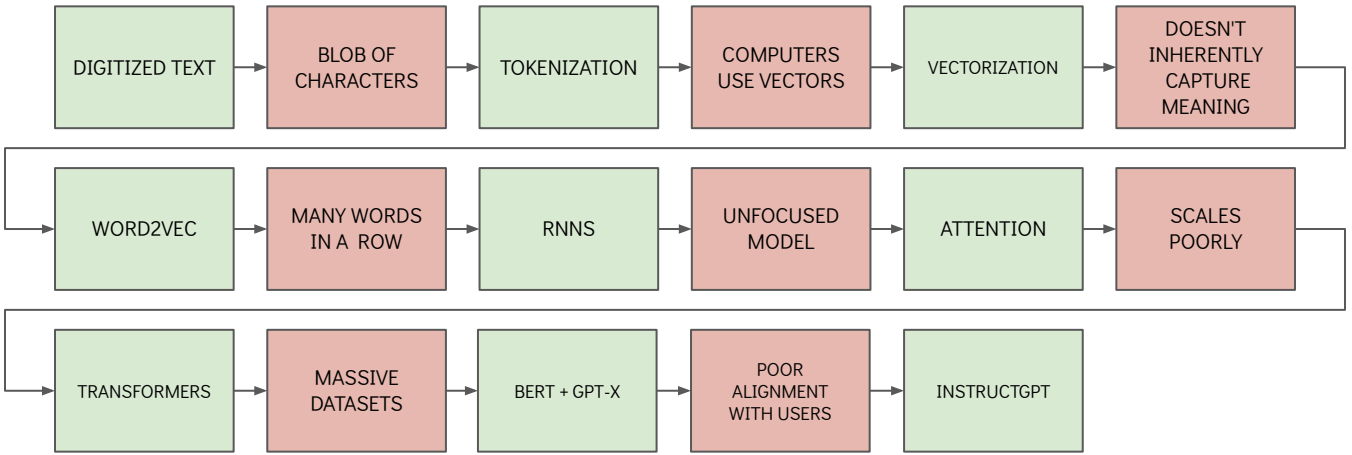
GPT	0.811
Supervised Fine-Tuning	0.880
InstructGPT	**0.902**

HOW DID WE GET HERE?

Longer sequences of words

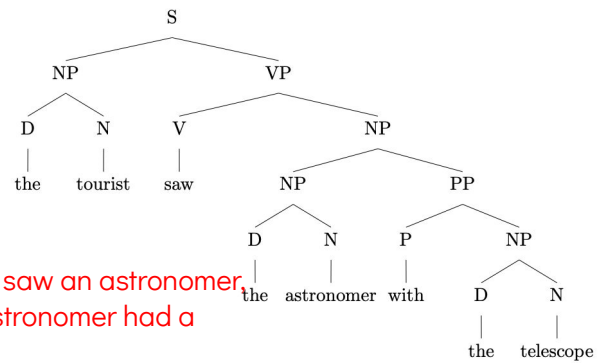
Larger datasets

Optimized for modern hardware

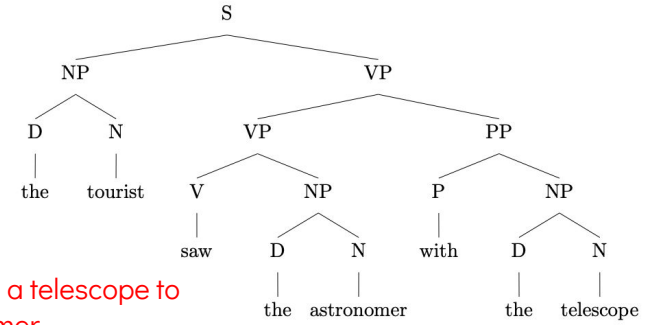


1. Give an explanation of two different meanings the following sentence could have, extra points for writing out the associated syntax trees

"The astronomer that the tourist saw had a telescope"



The tourist saw an astronomer, and that astronomer had a telescope.



The tourist used a telescope to see the astronomer.

2. Come up with an utterance which is grammatical but is semantically meaningful and one that is semantically meaningful but not traditionally grammatical

Discussion Question! (e.g. "Sad yesterday's bike." "I need to get rid of things, I need to have less shoes."

3. Come up with an example of how you might use word embeddings for an NLP task

Discussion Question! (e.g. Look for word embeddings in text that are near to the embedding for words like "Paris" to extract other countries mentioned in a text)

Exercises for Next Time

- Recurrent networks exploit translation symmetry in text (you can shift text and it's still text). What symmetry do convolutional neural networks exploit?

- GPT-2 had a context window of 1024 tokens. ChatGPT / GPT-3.5 has a context window of 8,192 tokens. How might this change the number of parameters in the model, keeping embeddings and number of attention heads / layers the same?

- Why does ChatGPT struggle to perform arithmetic when it can write functional code?