IA Générative

Bénéfices et défis pour le Traitement Automatique des Langues

Claire Gardent

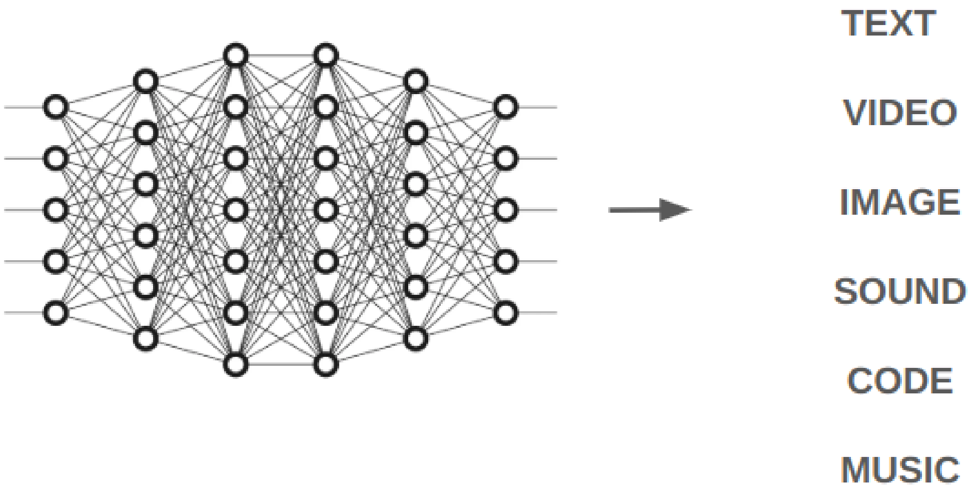
CNRS / LORIA, Nancy

Generative AI

What is Generative AI ?

A branch of AI which ***generate*** new content using Machine Learning techniques:

- Supervised Learning: Requires training data (Input/Output Examples)
- Deep Learning: Uses Neural Networks



TEXT

VIDEO

IMAGE

SOUND

CODE

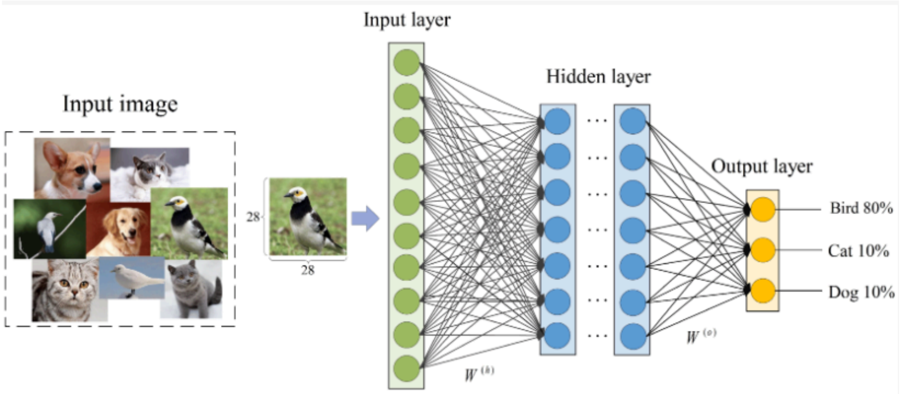
MUSIC

What's a Neural Network ?

Neural Network, Deep Learning

- Neurons are connected to one another in enormous networks
- Each neuron does a simple pattern recognition tasks
- When triggered, the neuron sends a signal to its connections
- The output is determined by which neurons was triggered

What's a Neural Network ?



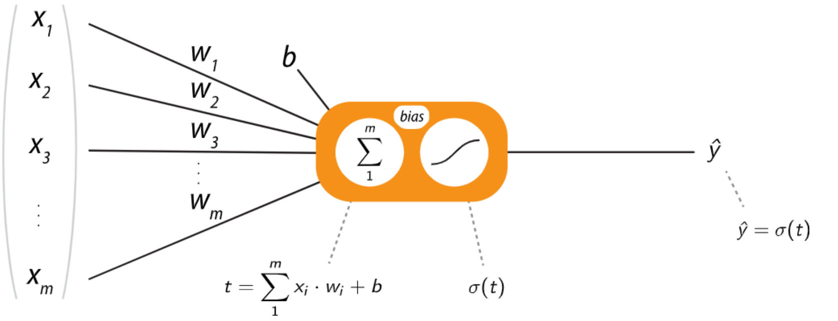
Source: Deng et al. 2022

A neural network has several layers

- A input layer which models the entry. E.g., for an image classifier, the image pixels
- An output layer which models the model prediction
- The output layer is often a probability distribution. E.g., the three output neurones indicates the probability of each target class (Bird, Cat or Dog)
- One or more intermediate layer(s) which models the relation between input and output

How do Neurons compute values ?

$$\hat{y} = \sigma(\Theta^T \cdot X + b)$$



Input X

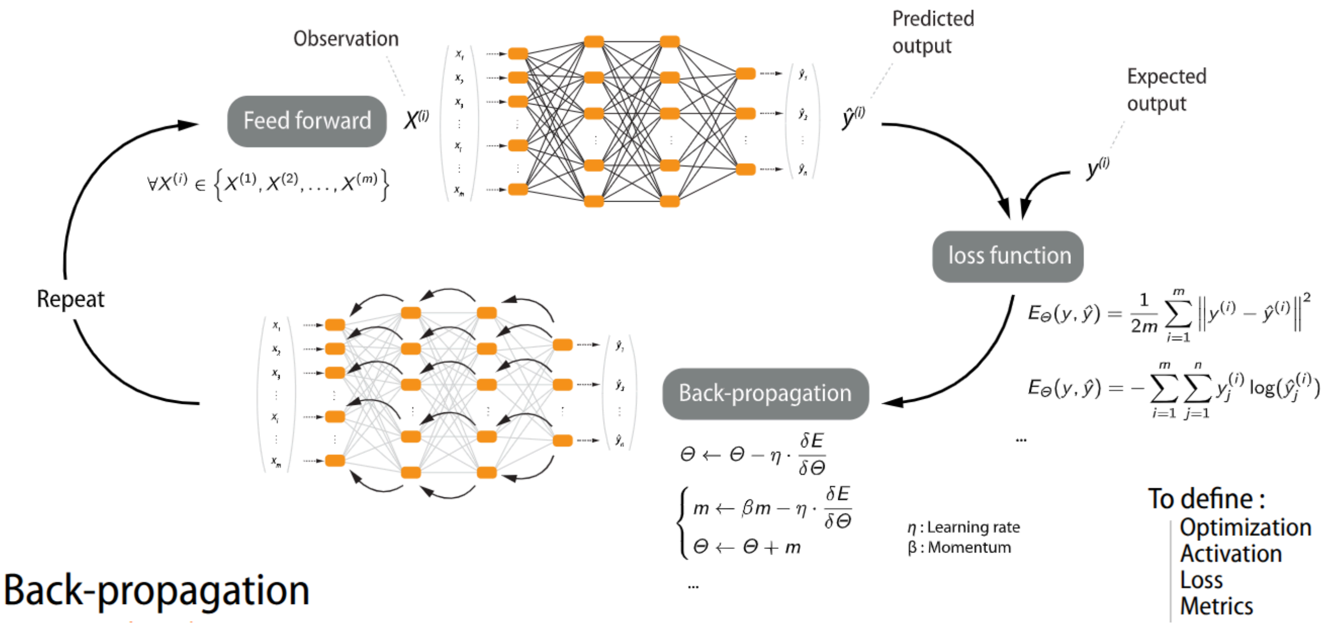
Bias / Weight Θ, b

Activation function $\sigma(t)$

Output $\hat{y}$

- Each neuron applies an ***activation function*** to the ***weighted sum of its inputs*** to return an ***activation value*** .
- This ***activation value*** is passed on as input (signal) to the next layer
- The ***weights*** are learned during training

Training - The Back-Propagation Algorithm



$\forall X^{(i)} \in \left\{X^{(1)}, X^{(2)}, \dots, X^{(m)}\right\}$

$$E_\Theta(y,\hat{y}) = \frac{1}{2m}\sum_{i=1}^{m}\left\|y^{(i)} - \hat{y}^{(i)}\right\|^2$$

$$E_\Theta(y,\hat{y}) = -\sum_{i=1}^{m}\sum_{j=1}^{n} y_j^{(i)} \log(\hat{y}_j^{(i)})$$

...

$$\Theta \leftarrow \Theta - \eta \cdot \frac{\delta E}{\delta \Theta}$$

$$\begin{cases} m \leftarrow \beta m - \eta \cdot \dfrac{\delta E}{\delta \Theta} \\ \Theta \leftarrow \Theta + m \end{cases}$$

...

η : Learning rate
β : Momentum

To define :

- Optimization
- Activation
- Loss
- Metrics

Back-propagation

Src: Fiddle Course on Deep Learning

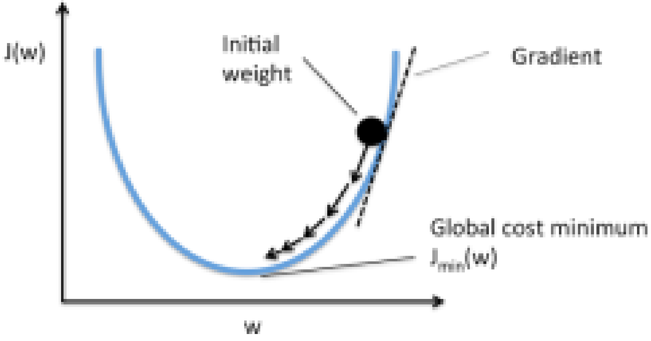
SGD

Stochastic Gradient Descent (SGD)

- updates the weights according to following rule (η = learning rate hyperparameter):

$$w \leftarrow w - \eta \frac{dJ(w)}{dw}$$

- moves each weight in the direction of the derivative (***gradient***)
- E.g., on the picture $dJ(w)$ is positive, hence the update rule decreases the value of w and $J(w)$ decreases.

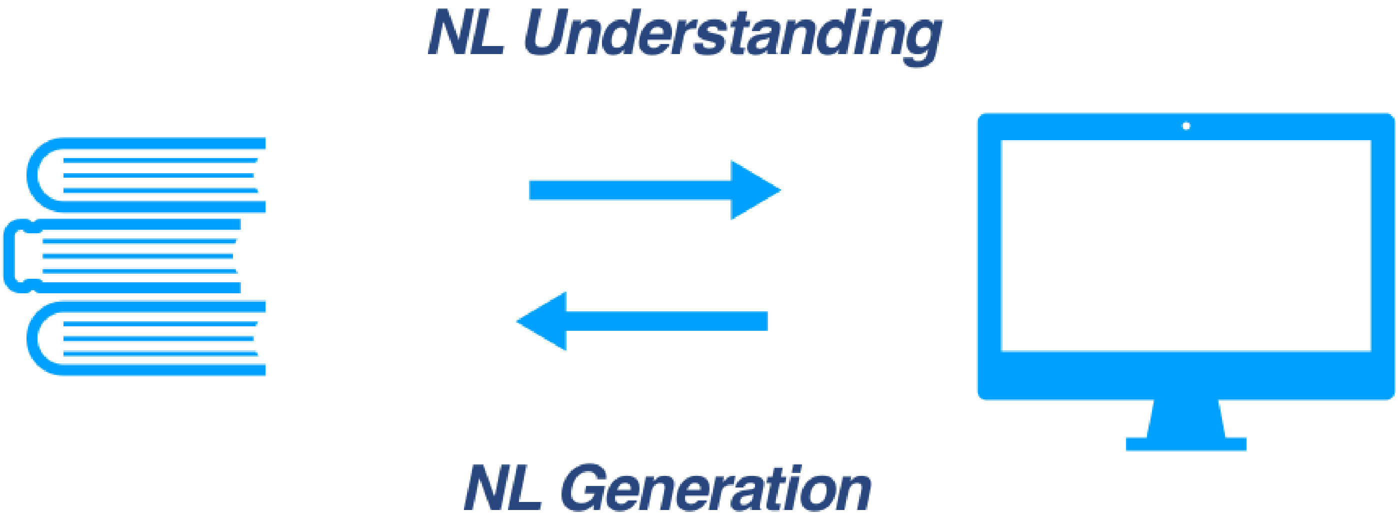


Rumelhart et al. 1986

Generative AI and Natural Language Processing

What is NLP ?

Natural Language Processing (NLP) is a field of artificial intelligence (AI) that focuses on enabling computers to ***understand*** and ***generate*** human language. It can also be used to study natural language i.e., to analyze and understand natural language structure and use.



Example NLP Tasks and Applications

NLU (Understanding)

- Spam Detection (Email filtering)
- Sentiment Analysis (Social media monitoring)
- Texte Classification
- Author Attribution
- Information Extraction
- Search Engines (Google, Bing)

The input is text

NLG (Generation)

- Human-Machine Dialog (Automated Customer Support, Chatbots)
- Translation (DeepL, Google Translate)
- Text Simplification (for non expert, non native speaker, people with reading disability)
- Summarisation
- Image captioning
- Video subtitles
- Creative writing (poems, novels, essays)

The input is varied: text, data, numerical data, images, video, a story title, etc.

Neural NLP Key Milestones

2014 - Encoder-Decoder for Machine Translation

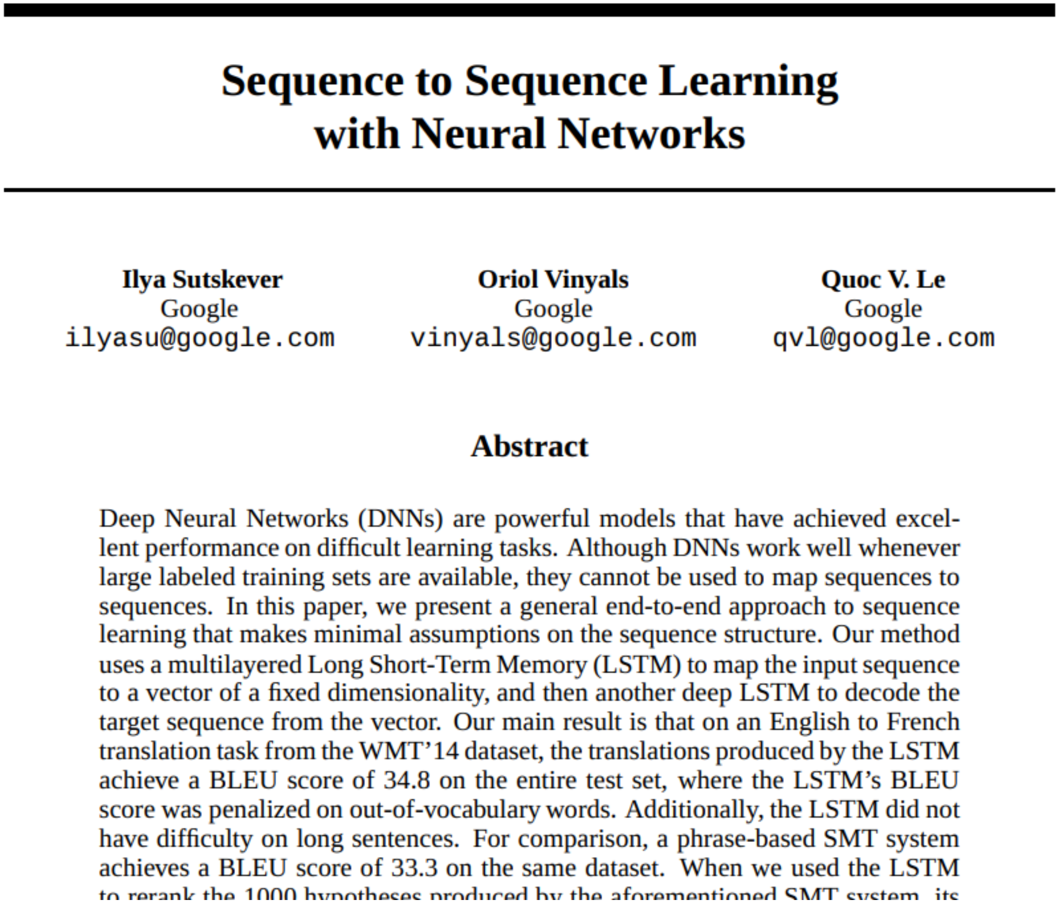
2015 - Cross-Attention

2017 - Transformer

- Pre-training and Fine-Tuning
- Parallelism allows scaling to large models and bigger training sets

2023 - LLMs (ChatGPT, BLOOM, Llama, LeChat, ...)

2014 - The Encoder-Decoder Architecture



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

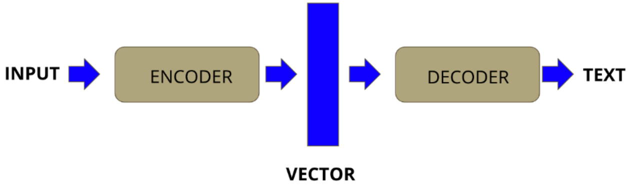
Abstract

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

The Encoder-Decoder Architecture

Encoder

- ***Builds a continuous representation*** of the input, a real valued vector
- Commonly used decoders:
  - Recurrent: RNN, LSTM, GRU
  - Convolutional
  - Graph
  - Transformer

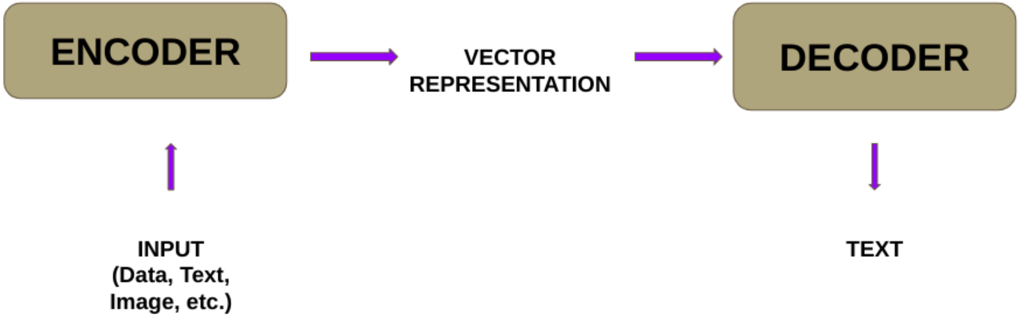


Decoder

- = ***Language Model***
- ***Generates text one word at a time***
- Conditioned on input
- Commonly used encoders:
  - Recurrent: RNN, LSTM, GRU
  - Transformer

The Encoder-Decoder Architecture

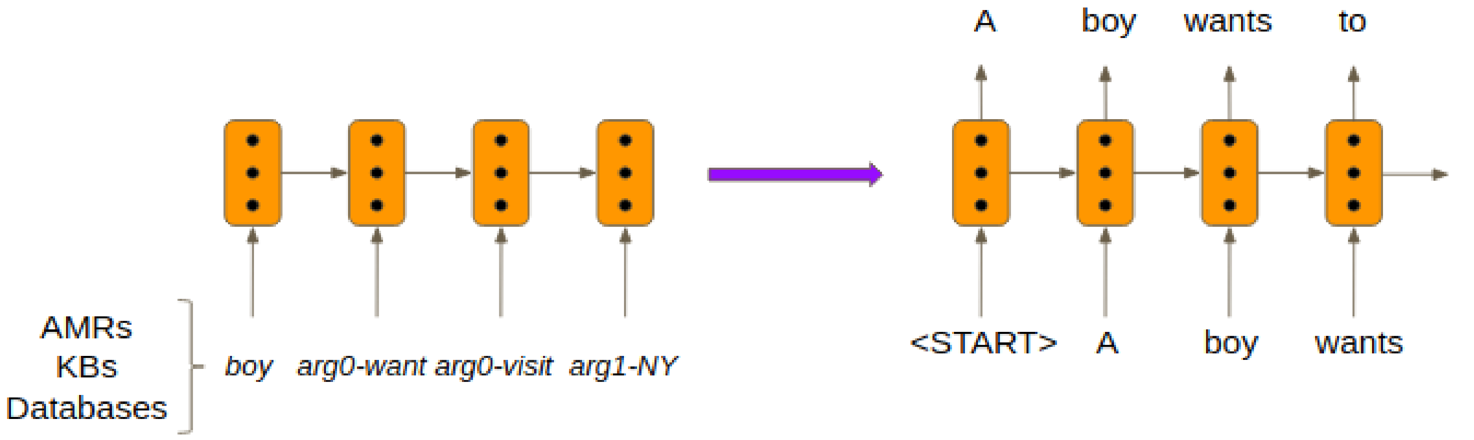
A unifying framework for all text production tasks



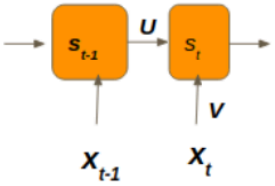
- ***End to end*** : Direct input-output mapping
- ***Unifying Framework for Text Generation*** : All types of input (data, text, meaning representation) are encoded into a numerical representation

Recurrent Encoder

- The input to NLG (text but also data and MRs) is a sequence of tokens
- Data or meaning representations need to be linearised first



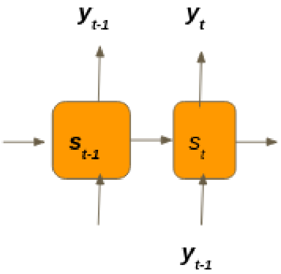
Encoding the Input using an RNN



$$s_t = tanh(U * s_{t-1} + V * x_t)$$

- x_i are vectors representing the input tokens (words, data or MR tokens)
- At each step, the encoder produces a new vector s_t (state) which represents the content of the preceding string of tokens
- The last state represents the meaning of the whole input
- U and V are the parameters learned during training
- tanh is a non linear function

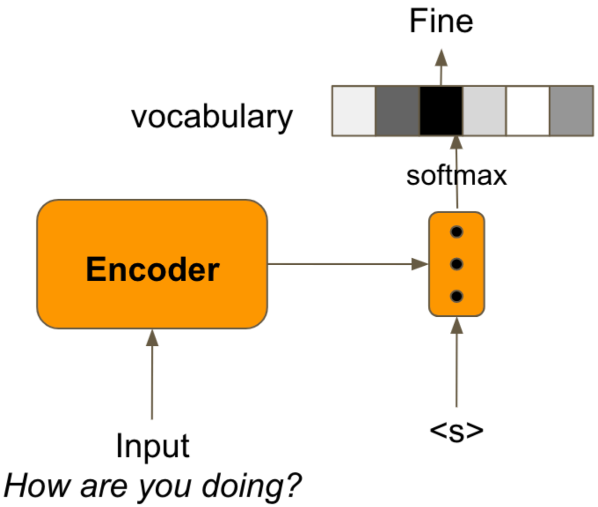
Decoding Words using an RNN



$$\begin{aligned} s_t &= tanh(U * s_{t-1} + V * y_{t-1}) \\ y_t &= softmax(W * s_{t-1}) \end{aligned}$$

- y_t is the word predicted at time t
- s_t is the network state at time t
- Each new state is computed taking into account the previous state s_{t-1} and the last predicted word y_{t-1}.
- The softmax function turns a vector of scores into a probability distribution
- At each time step t, the output/predicted token y_t is sampled from this probability distribution

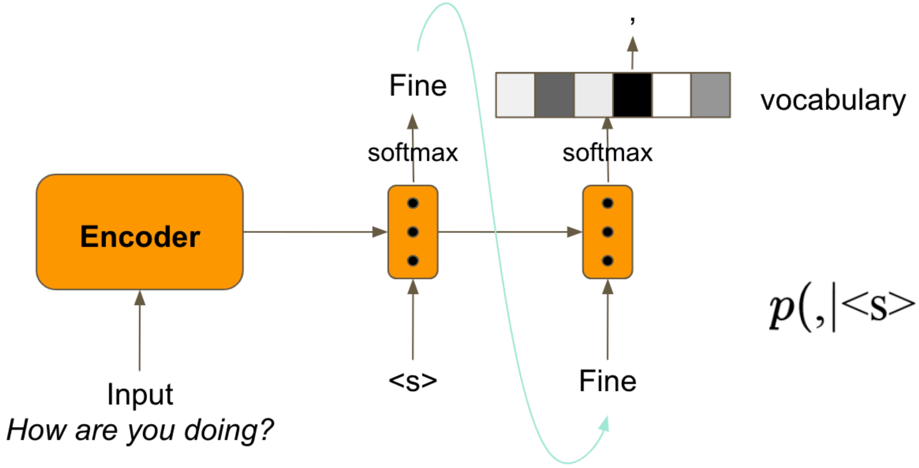
Generating Text using an RNN



$$p(\text{Fine} | \text{<s>}, \text{How are you doing?})$$

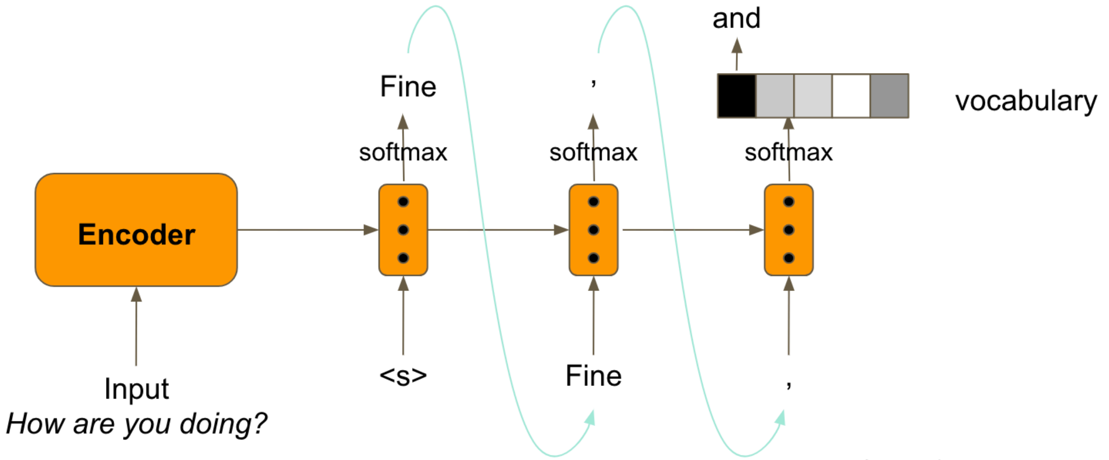
Conditional Generation

Generating Text using an RNN



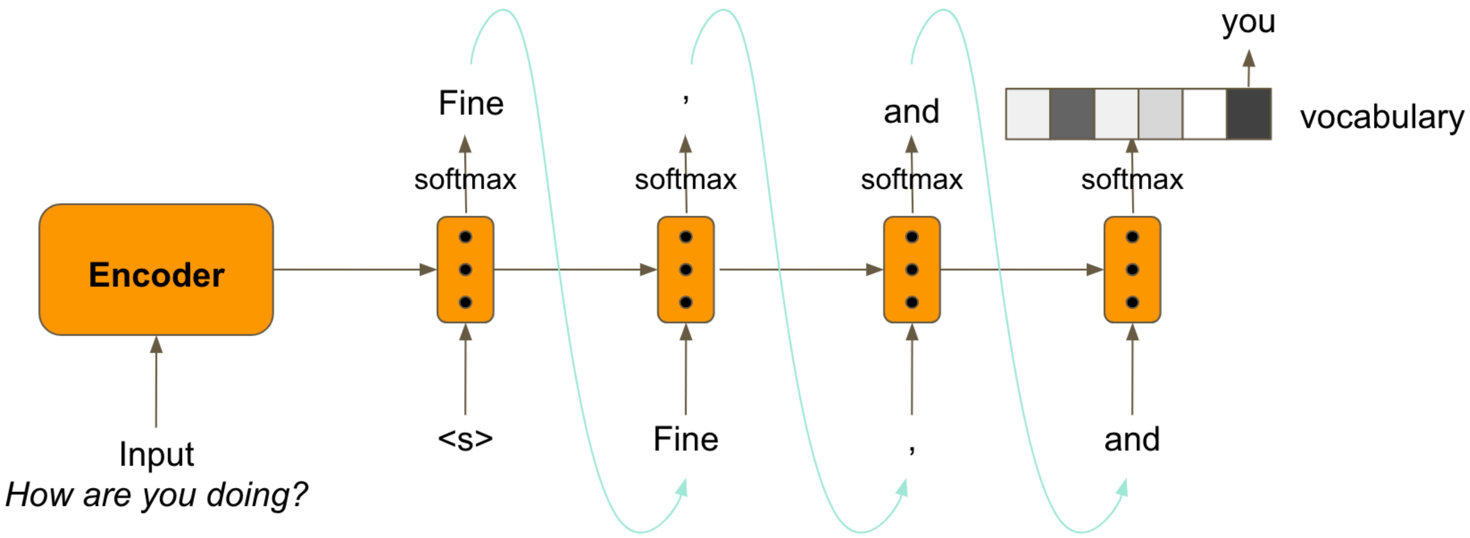
$p(,|\text{<s> Fine, How are you doing?})$

Generating Text using an RNN

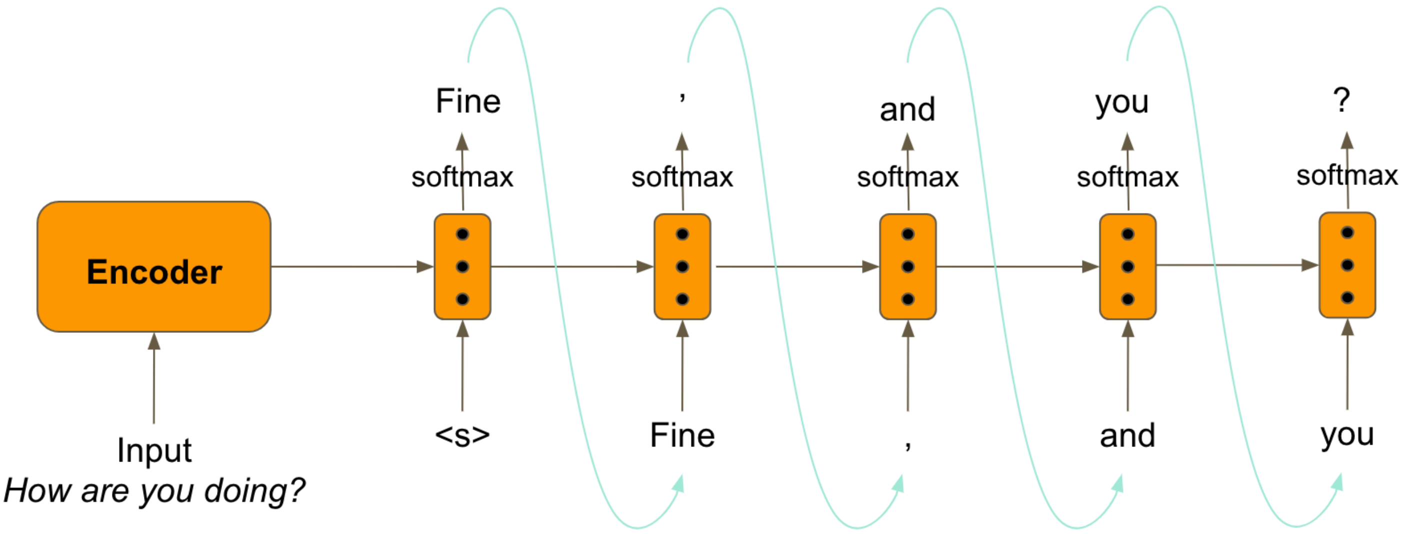


$p(\text{and}|\text{<s> Fine,; How are you doing?})$

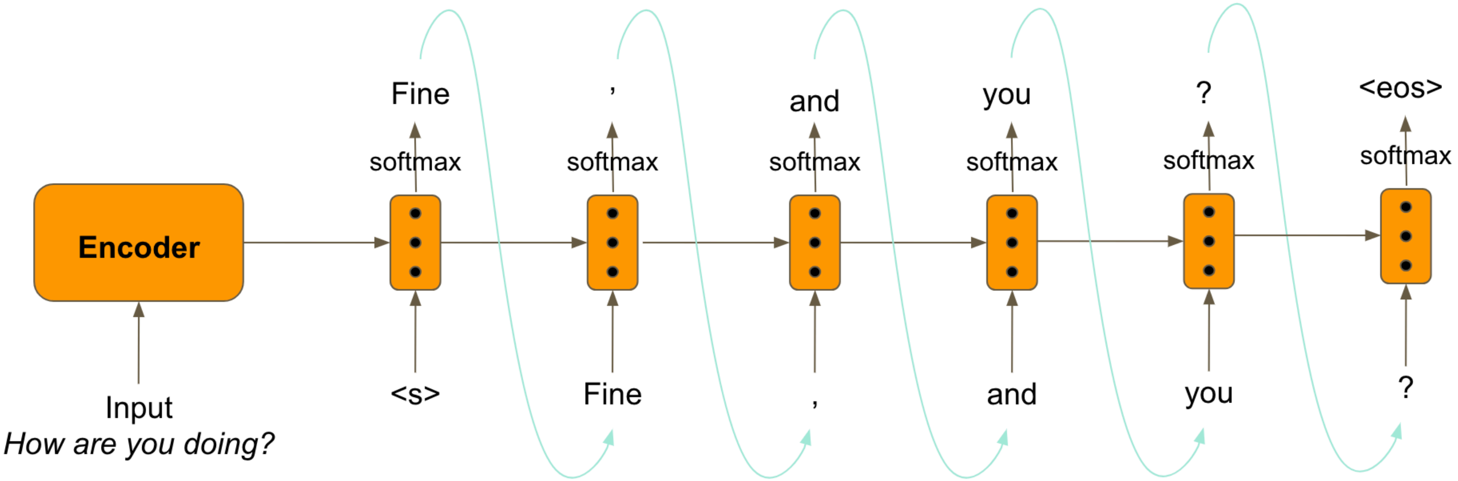
Generating Text using an RNN



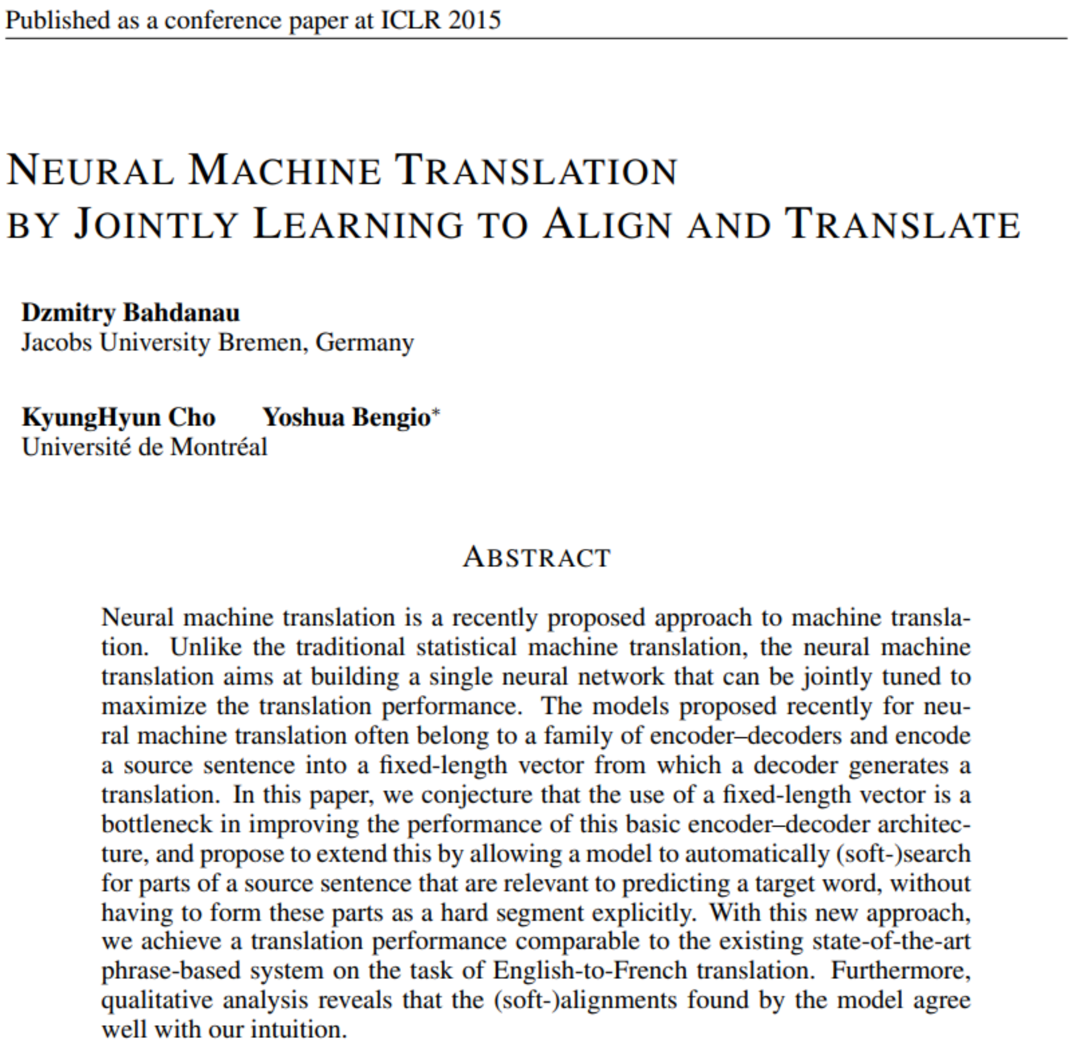
Generating Text using an RNN



Generating Text using an RNN



2015 - Decoding with Attention



Published as a conference paper at ICLR 2015

NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

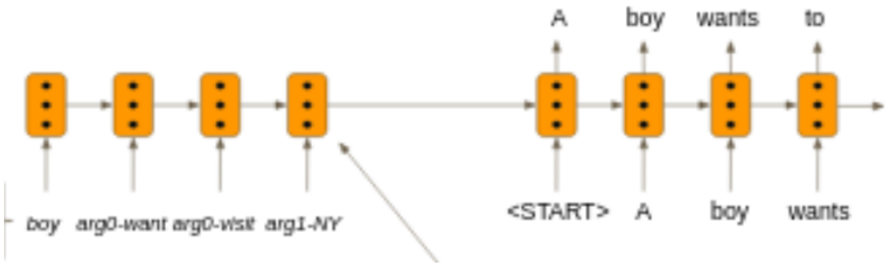
Dzmitry Bahdanau
Jacobs University Bremen, Germany

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.

Standard RNN Decoding



- The input is compressed into a fixed-length vector
- Performance decreases with the length of the input

Bahdanau et al. 2015

Decoding with Attention

$$s_t = f(s_{t-1}, y_{t-1}, c_t)$$

A ***context vector*** c_t is added which

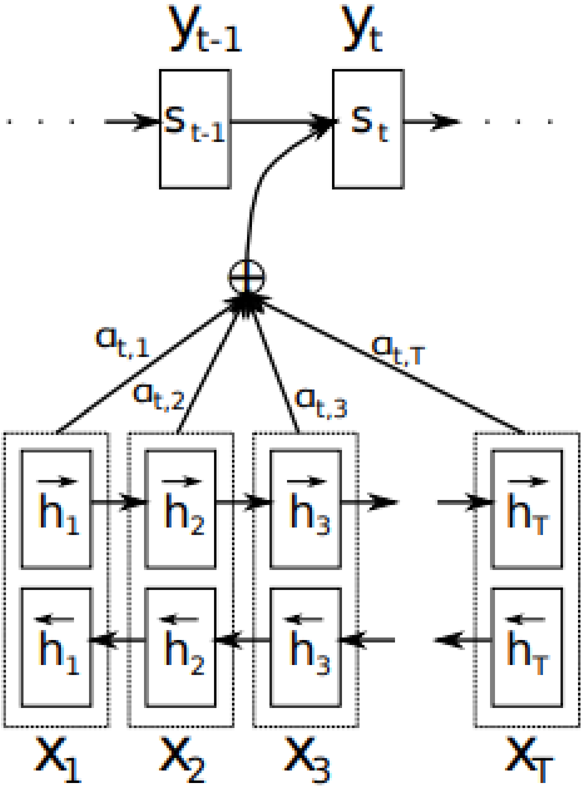
- depends on the previous encoder states and therefore ***changes at each step***
- indicates ***which part of the input is most relevant*** to the decoding step

RNN Cross-Attention

- A score $a_{t,j}$ is computed between each input token encoder state h_j and the previous state s_{t-1}

- The context vector is the weighted sum of the encoder states passed thru a softmax layer

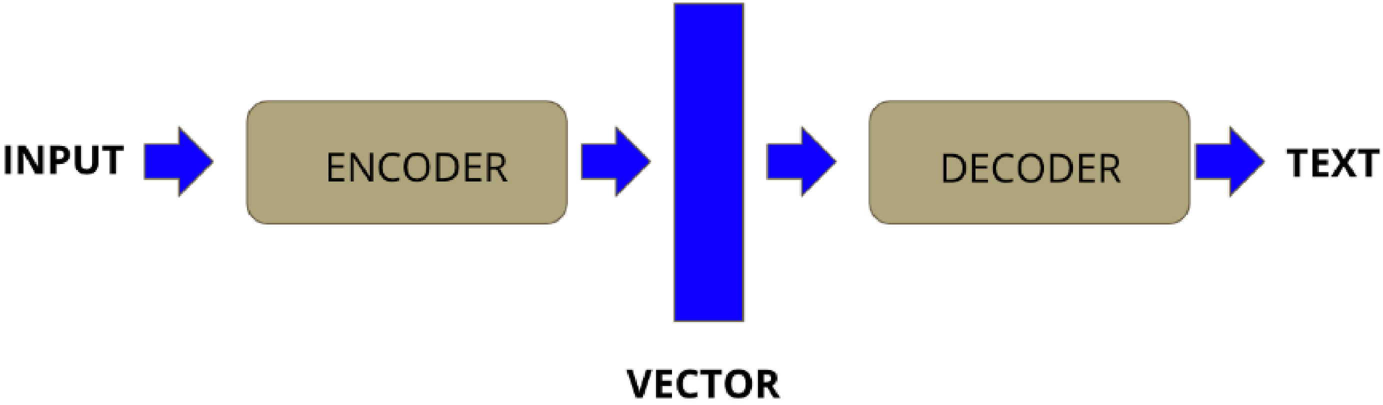
$$c_t = softmax(\sum_j \alpha_{t,j}.h_j)$$



Attention

- Attention is a way to obtain a fixed-size representation
  - of an arbitrary set of representations (the values),
  - dependent on some other representation (the query)
- Encoder-Decoder
  - Query = current decoder state
  - Values = encoder hidden states
- Transformer
  - Query = token embedding
  - Values = surrounding tokens embeddings

The Encoder-Decoder Model



- Encoder: vectorises the input
- Decoder: autoregressively generates from this input
- Attention: helps the decoder focus on the relevant part of the input

2017 - Transformer Network

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

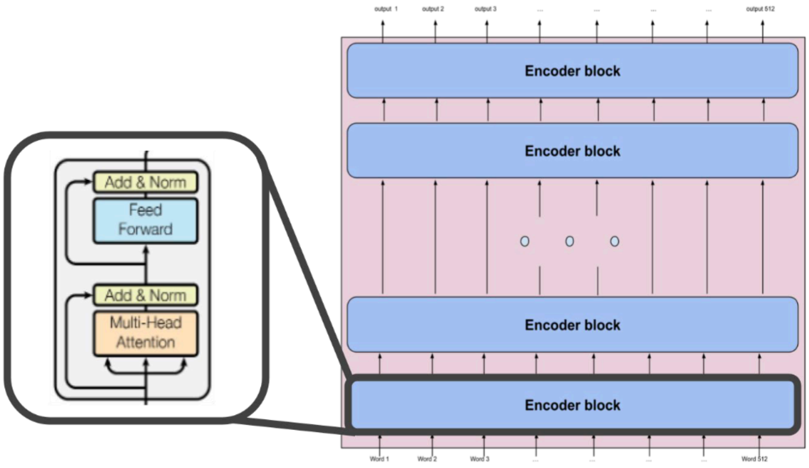
Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or

The Transformer Encoder



- ***Deep and structured*** model Stack of Encoder Blocks
- ***No sequential dependencies*** (different from RNN)
- ***Self-attention*** -- better word representations
- ***Parallel*** processing -- Scaling, Pre-training and fine-tuning

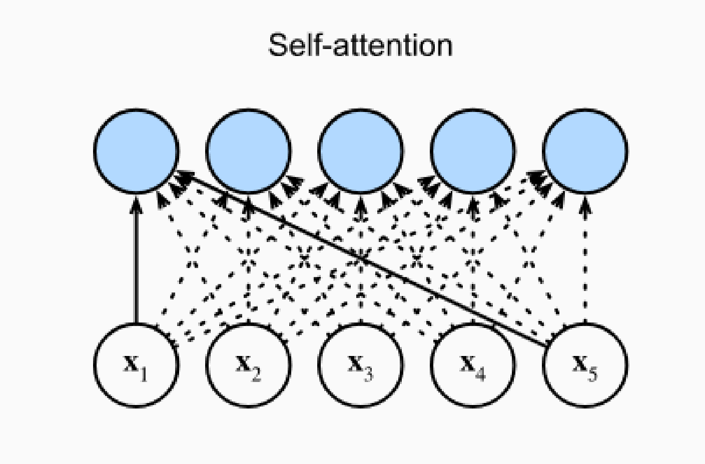
Vaswani et al. NIPS 2017

Self-Attention Layer

Computes a ***context dependent representation of each word*** in the input sequence

- Score the encoding of each input word against the encoding of each other input words
- The output representation of each word is the weighted sum of the representations of its surrounding words

  Capture ***lexical ambiguity*** : the same word will have different representations depending on its context



Jean lit un **livre**

Jean **livre** un colis

Ce colis pèse une **livre**

La **livre** sterling est la monnaie du Royaume Uni

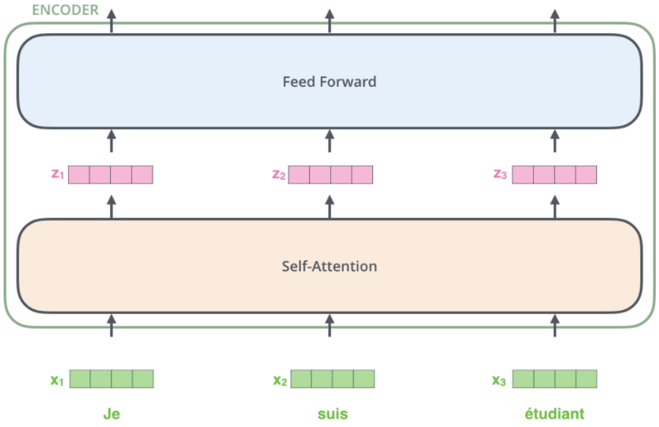
Scaling

No sequential dependencies.

Facilitates parallelism

- Different processors can be used to process input tokens in parallel.

This enabled scaling, training on larger amounts of data than was possible before.



*Transformers lead to the introduction of the **pre-training and fine-tuning** paradigm (BERT, T5, BART) and facilitated the creation of **very large models** (e.g., ChatGPT).*

Pre-training and Fine Tuning

Pre-train once, fine-tune many times

How ?

- Find a task (e.g., Language Modeling) for which it is easy to generate labels and for which you can get large quantities of training data
- ***Pre-training*** : train a model on this large data
- ***Fine-tuning*** : adapt it to a task using labelled data

Pre-training and Fine Tuning - Benefits

A pre-trained model encodes a lot of information about language

Data

> Less labeled data required

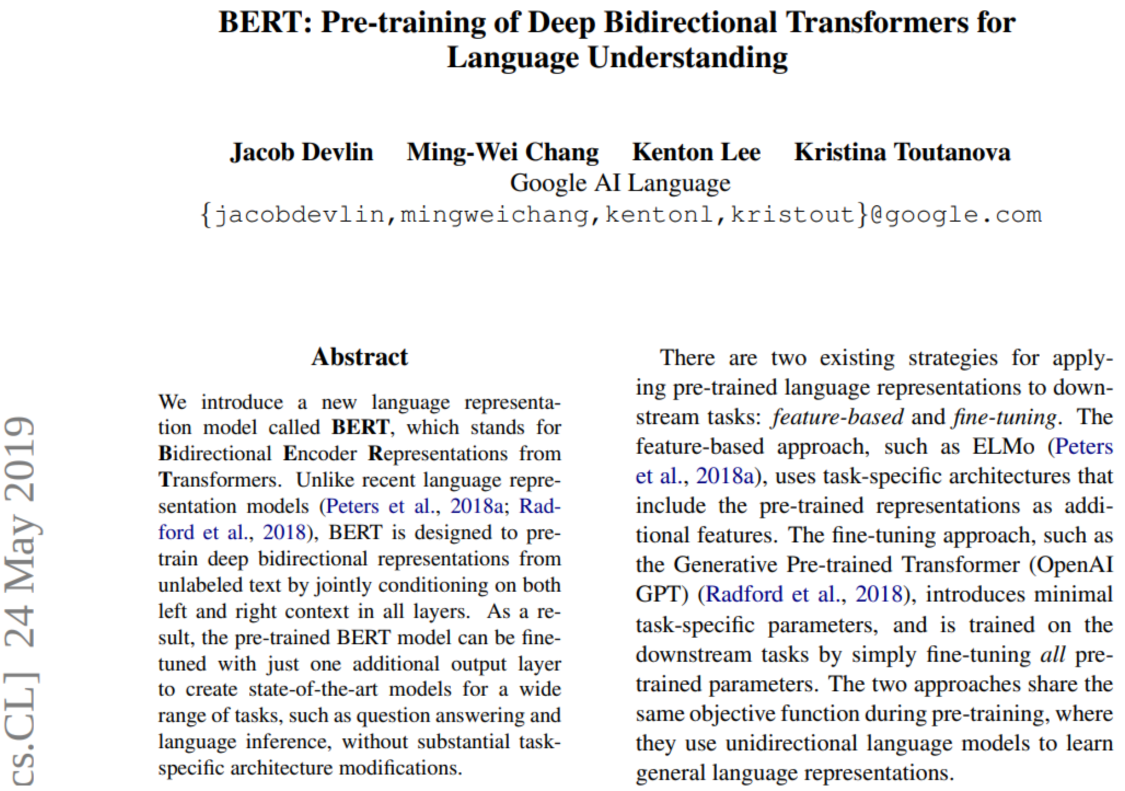
Efficiency

> Less time to fine-tune than to train from scratch

Generalisation

> Achieves state of the art results for a wide variety of tasks: classification, language inference, semantic similarity, question answering, etc.

2019 - BERT



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pretrained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

BERT

Pre-Training

Large Transformer Encoder - 340M parameters, 24 layers

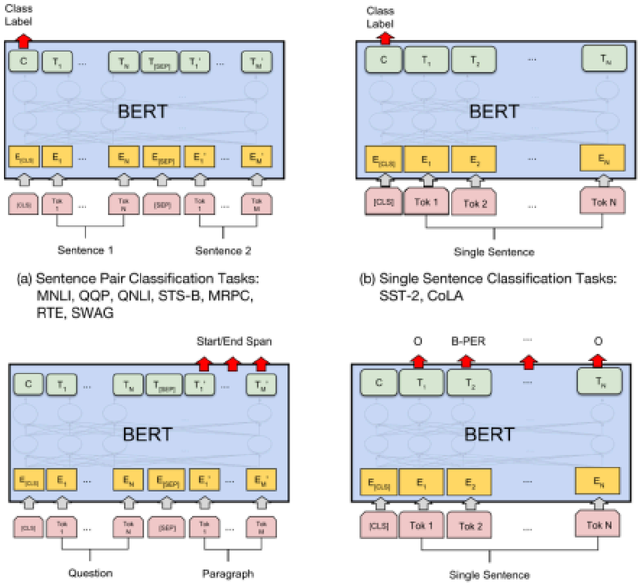
Pre-trained on a large quantity of text - BooksCorpus (800M words) and English Wikipedia (2,500M words)

Masked Language Modeling Objective - Predict missing word

> ***Improved word representations (Self Attention)***
>
> ***A generic model that can be fine-tuned for multiple NLU tasks***

Fine-Tuning

Adapts the model parameters to the target task by further training on labeled data from various target tasks



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

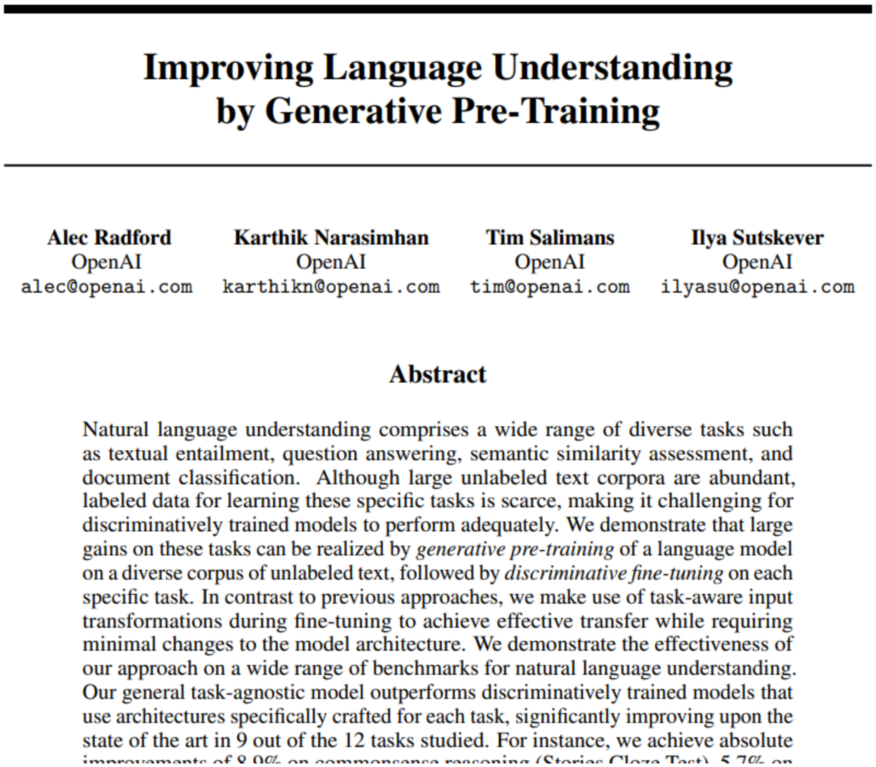
(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

BERT Impact

- Open sourced by Google in 2018
- Achieved ***state-of-the-art results in 11 natural language understanding (NLU) tasks***, including sentiment analysis, semantic role labeling, text classification and the disambiguation of words with multiple meanings.
- In contrast to previous models, such as word2vec and GloVe, BERT effectively addresses ***ambiguity***, a key challenge to NLU.
- Estimated to ***enhance Google's understanding of approximately 10% of U.S.-based English language Google search queries*** .

2018 - GPT



Improving Language Understanding by Generative Pre-Training

Alec Radford
OpenAI
alec@openai.com

Karthik Narasimhan
OpenAI
karthikn@openai.com

Tim Salimans
OpenAI
tim@openai.com

Ilya Sutskever
OpenAI
ilyasu@openai.com

Abstract

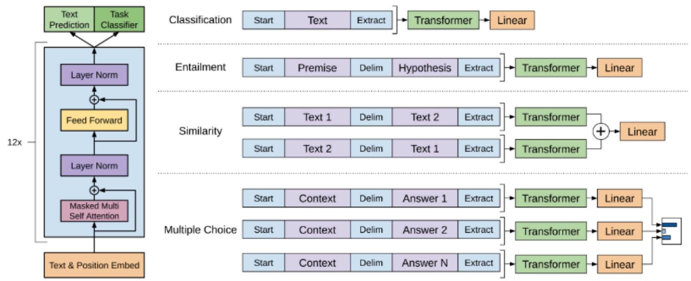
Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by *generative pre-training* of a language model on a diverse corpus of unlabeled text, followed by *discriminative fine-tuning* on each specific task. In contrast to previous approaches, we make use of task-aware input transformations during fine-tuning to achieve effective transfer while requiring minimal changes to the model architecture. We demonstrate the effectiveness of our approach on a wide range of benchmarks for natural language understanding. Our general task-agnostic model outperforms discriminatively trained models that use architectures specifically crafted for each task, significantly improving upon the state of the art in 9 out of the 12 tasks studied. For instance, we achieve absolute improvements of 8.9% on commonsense reasoning (Stories Cloze Test), 5.7% on

GPT - Generative Pre-Trained Transformer

Pre-Training

Large Transformer ***decoder*** - 117M parameters, 12 layers

Pre-trained on a large corpus of text using ***Language Modeling objective*** - BookCorpus 7K books



Radford et al. 2018

Fine-Tuning

GPT can be fine-tuned on NLU tasks such as classification, entailment, sentence similarity, question answer task

Input sequences are processed by the pre-trained model.

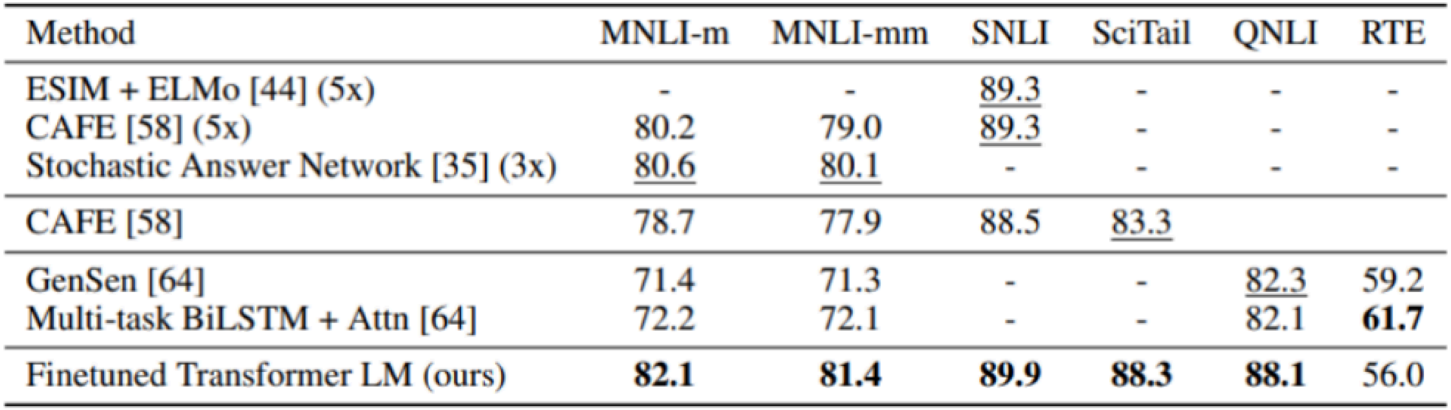
During fine-tuning, the model has two heads:

- the standard ***LM head*** for predicting the next word as an auxiliary head
- a ***task specific head*** e.g., a classification head (an additional linear+softmax layer) as main head

GPT Fine-tuning

Significantly improves upon the SOTA in 9 out of 12 NLU tasks

Results for NLI

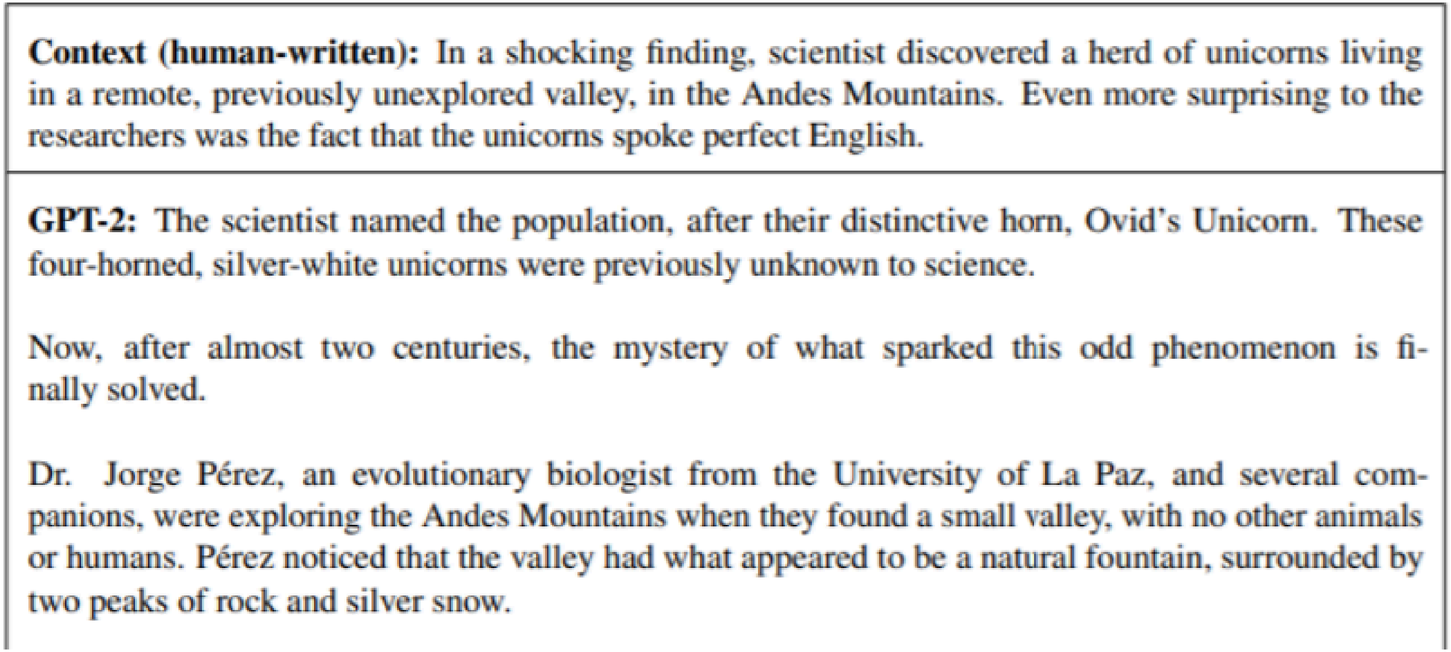


Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

Improved Text Generation

GPT-2, a larger version (1.5B) of GPT trained on more data was shown to produce convincing text e.g.,

Story Generation



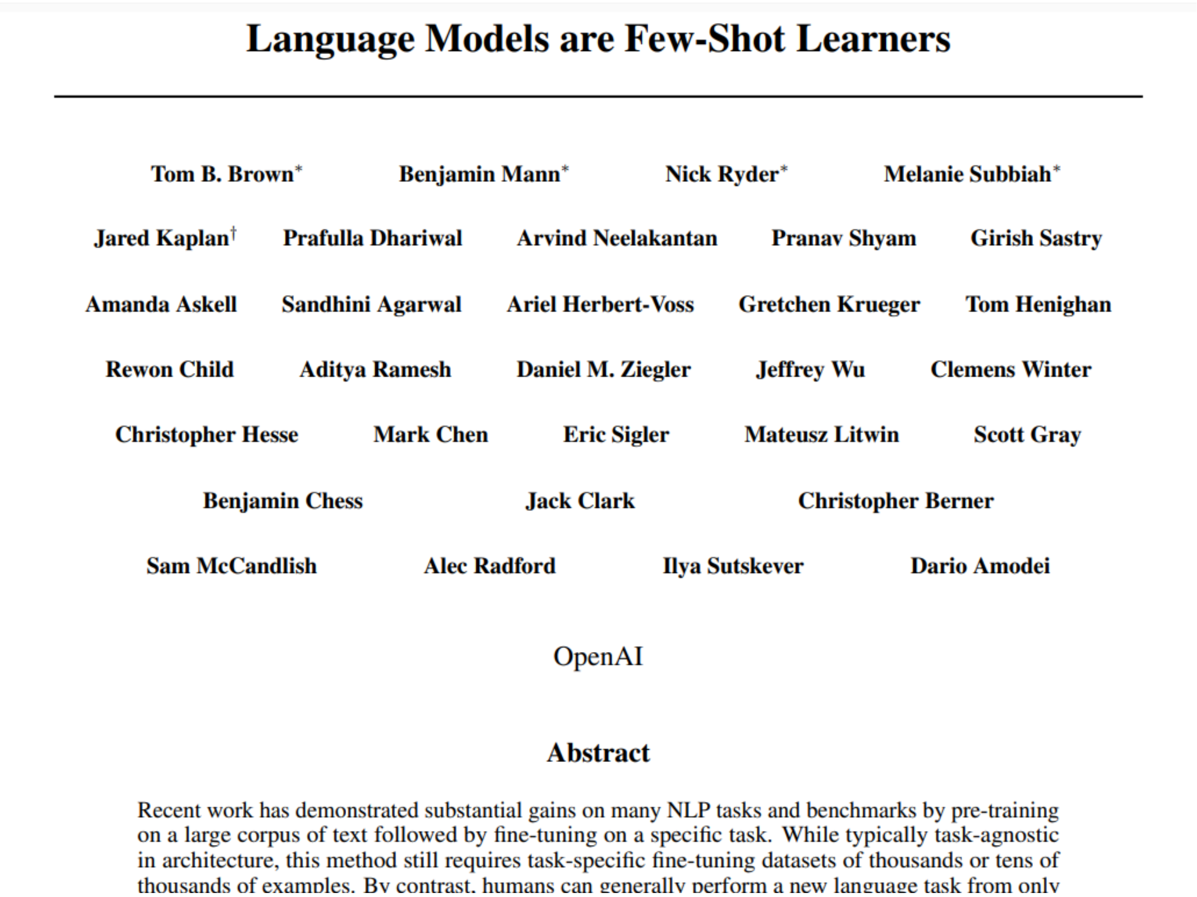
Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

2020 - GPT3



Language Models are Few-Shot Learners

Tom B. Brown[*] Benjamin Mann[*] Nick Ryder[*] Melanie Subbiah[*]

Jared Kaplan[†] Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

Abstract

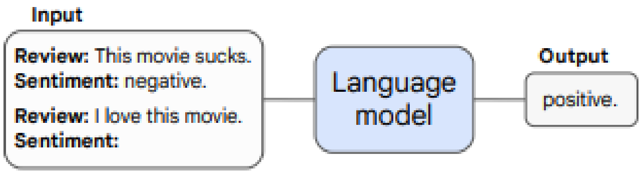
Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only

2020 - GPT3

Large Language Model (LLM)

- Transformer Decoder
- 175B parameters
- Trained on 500B words

Prompting suffices, no Fine-Tuning



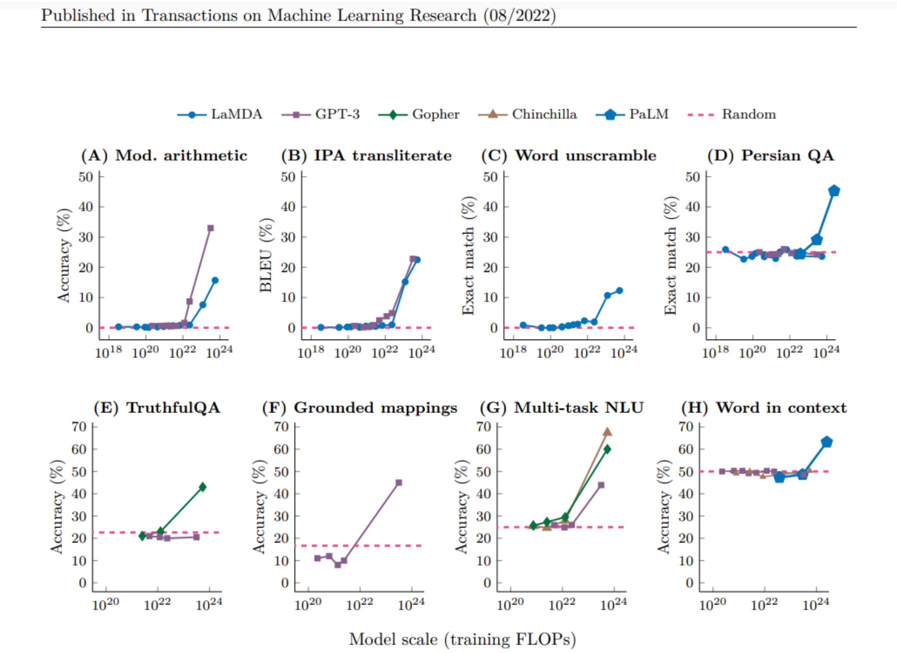
Review: This movie sucks.
Sentiment: negative.

Review: I love this movie.
Sentiment:

The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin.

Rich Sutton. "The bitter Lesson", March 2019

2020 - GPT3



Published in Transactions on Machine Learning Research (08/2022)

Emergent Properties - *An ability is emergent if it is not present in smaller models but is present in larger models*

Wei et al., TMLR 2022

2023 - ChatGPT and InstructGPT

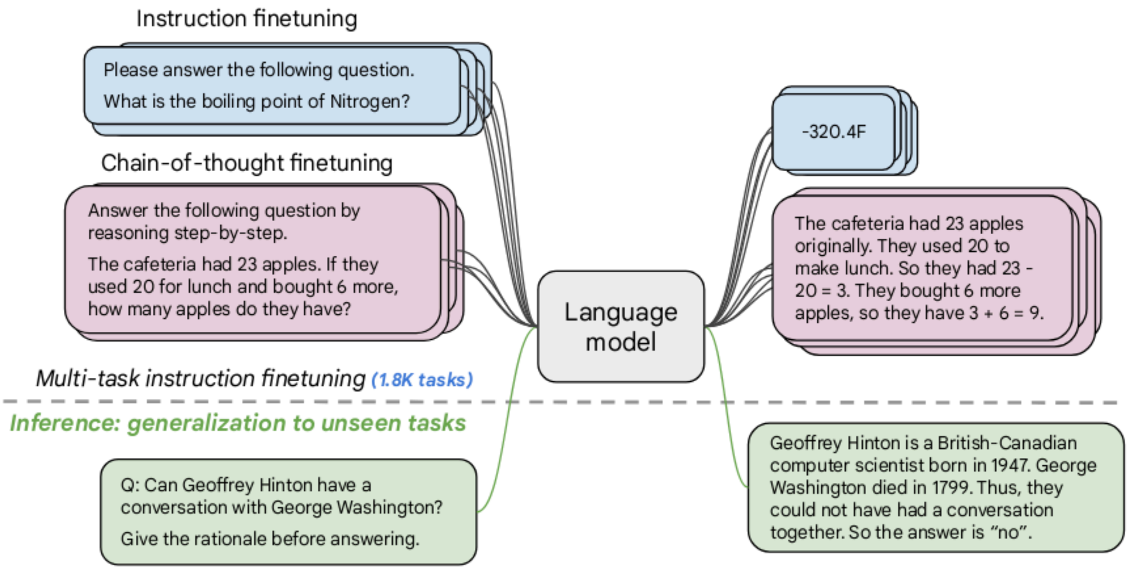
ChatGPT

- A variant of GPT-3 optimised for conversation
- Fine tuned on conversational data
- Better suited for chatbots and conversational interaction

InstructGPT

- Initialised with GPT-3
- Fine-tuned on Human instructions
    - Supervised fine-tuning on tasks specific data
    - Alignement with human preferences using Reinforcement Learning with Human Feedback (RLHF)

Instruction Tuning



Instruction finetuning

Please answer the following question.
What is the boiling point of Nitrogen?

Chain-of-thought finetuning

Answer the following question by reasoning step-by-step.

The cafeteria had 23 apples. If they used 20 for lunch and bought 6 more, how many apples do they have?

Multi-task instruction finetuning ***(1.8K tasks)***

Inference: generalization to unseen tasks

Q: Can Geoffrey Hinton have a conversation with George Washington?

Give the rationale before answering.

Language model

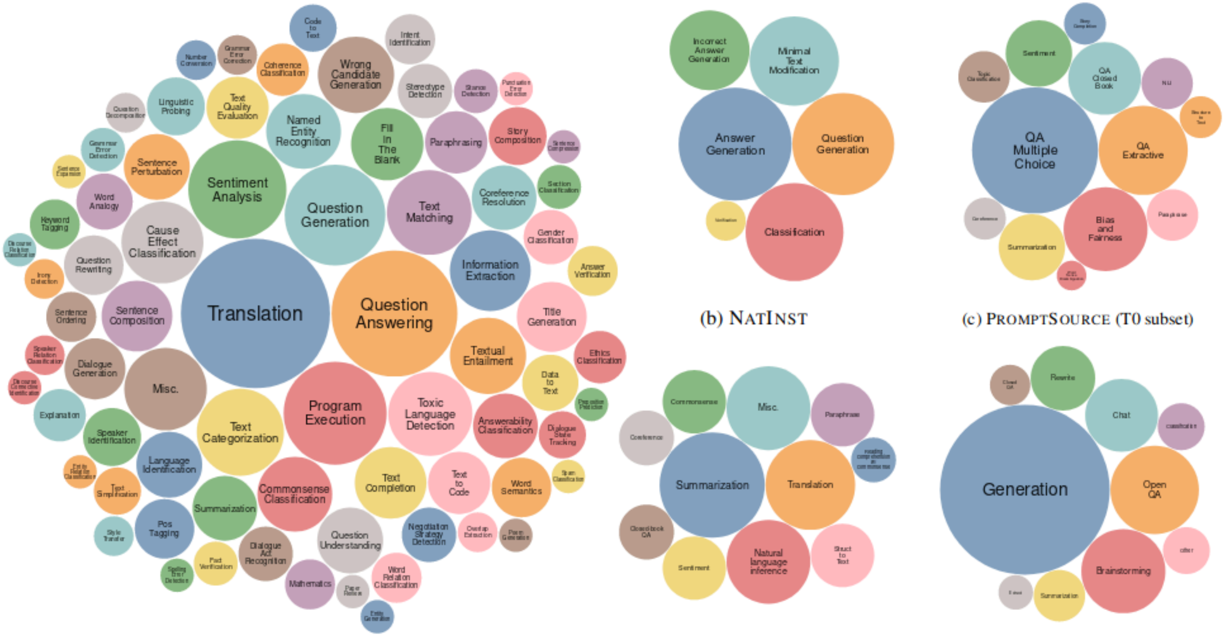
-320.4F

The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9.

Geoffrey Hinton is a British-Canadian computer scientist born in 1947. George Washington died in 1799. Thus, they could not have had a conversation together. So the answer is “no”.

SuperNatural Instruction Dataset

The SuperNatural Instruction Dataset contains over 1.6K tasks, 3M examples



(a) SUP-NATINST (this work)

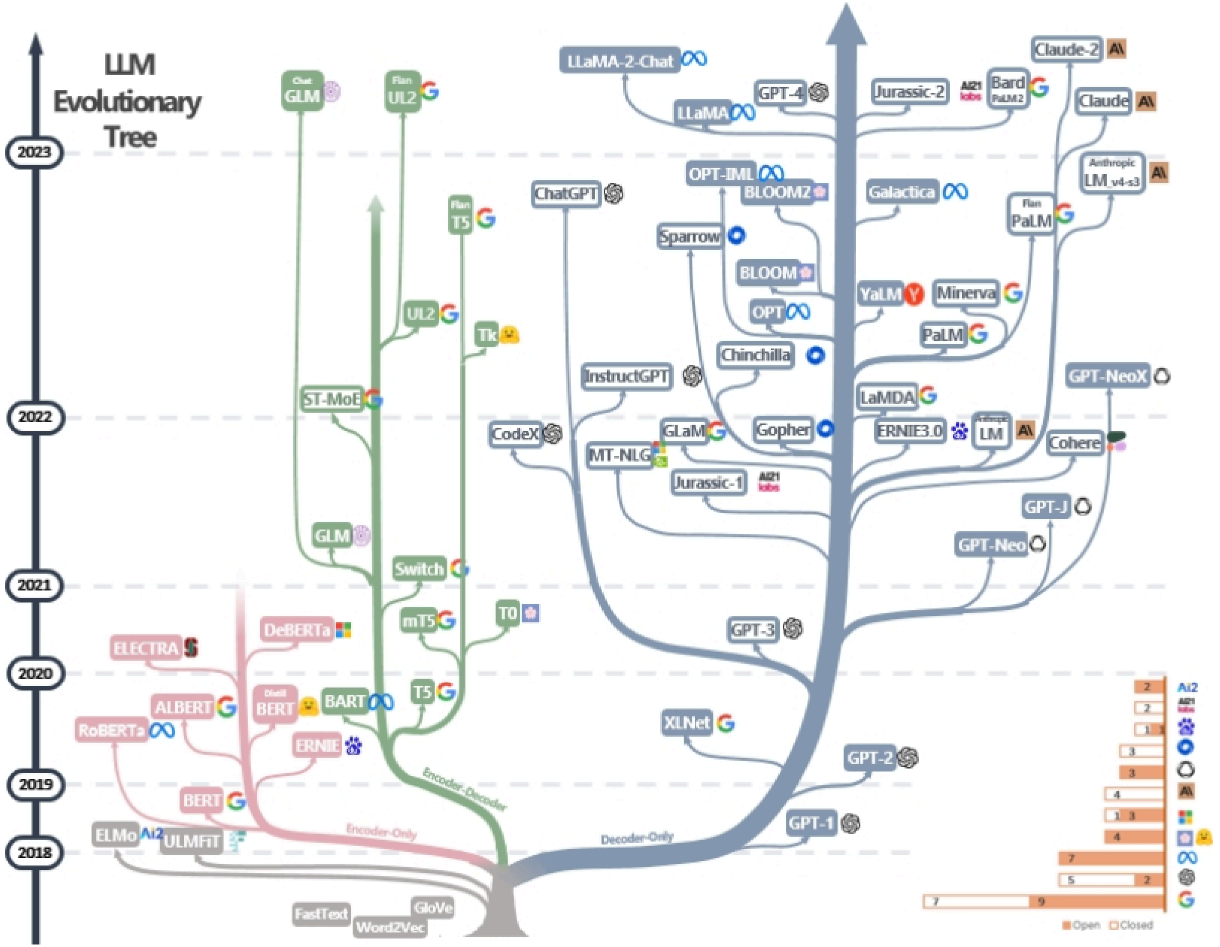
(b) NATINST

(c) PROMPTSOURCE (T0 subset)

(d) FLAN

(e) INSTRUCTGPT

Wang et al. 2022

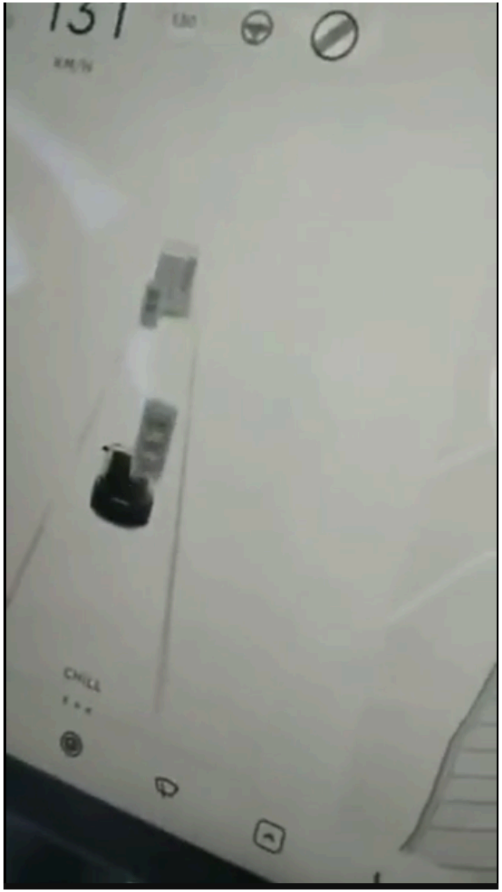


Yang et al. 2023, "Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond". arXiv 2304.13712

Issues with the technology

wbr>

- LLMs get things wrong! They have no notion of truth.
- Bias and toxicity
- Copyright and Intellectual Property
- GDPR/Personal Data
- Generalisation to out of domain data
  Tesla Example
- Environmental Cost



NLP Challenges

Improving factuality, consistency

From generic to specific

- Adapting LLMs to a new domain, task, language

Evaluation

- NLU is easy (accuracy, F1, etc.)
- NLG (LLM output) is hard because language has high paraphrastic power

Generation from data

- Verbalisation of knowledge graphs, numerical, tabular data etc.

Multilinguality

- Not all languages are handled equal by LLMs

Trends

Retrieval Augmented Generation (RAG)

- Augment LLMs with knowledge from external sources
- Helps with: Hallucination, Lack of attribution, Data Privacy, Limited context

Fine-Tuning

- Parameter efficient fine tuning (LoRA, Adapters, etc.); Helps with domain/language/task adaptation
- Preference learning (DPO); Helps with OOD generalisation, alignemnt with human preferences

Agentic AI

- Using multiple LLMs together: Helps with Data Creation, Evaluation

Questions ?

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In ICLR.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei. Language Models are Few-Shot Learners. arXiv:2005.14165, Jul 2020.

Deng, Guoqiang, Min Tang, Yuhao Zhang, Ying Huang, and Xuefeng Duan. 2022. "Privacy-Preserving Outsourced Artificial Neural Network Training for Secure Image Classification" Applied Sciences 12, no. 24: 12873. https://doi.org/10.3390/app122412873

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long- short-term memory. Neural computation 9(8):1735–1780

Kyunghyun Cho, Bart Van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the Properties of Neural Machine Translation: Encoder-decoder Approaches. In Proc. of SSST.

Radford et al. 2018. Improving Language Understanding by Generative Pre-Training

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323, 533--536.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks.. In Proc. of NIPS. pages 3104–3112

Rich Sutton, March 2019. The bitter Lesson]

Vaswani et al. NIPS 2017. Attention is all you need]

Wang et al. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks.]

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean and William Fedus. Emergent Abilities of Large Language Models. Transactions on Machine Learning Research (08/2022)

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned Language Models Are Zero-Shot Learners. ICLR 2022.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le and Jason Wei. Scaling Instruction-Finetuned Language Models. arXiv, 2022.