Neural Machine Translation

Implementation by using TensorFlow 2.0

Harshith Belagur (177221)

Paper Title - Neural Machine Translation by Jointly Learning to Align and Translate

- Authors - Dzmitry Bahdanau (Jacobs University Bremen, Germany), KyungHyun Cho (University of Montreal) and Yoshua Bengio (University of Montreal and CIFAR Senior Fellow)
- Conference Presented - International Conference on Learning Representations (ICLR), 2015

NLP

Brief History
The Timeline of NLP

Before 2000's - Statistical Modeling

2013 - Neural Networks for Translation

2014 - RNN for Translation - Cho *et al.* and Sutskever *et al. ,* Teacher Forcing methods

Attention - First introduced in 1997, but brought back by Google in 2014

2015 - Attention in NLP for the first time

2017 - Seq2Seq Models

2018 - Transformers!!! - Attention is All You Need! by Vaswani et al. of Google Brain and UToronto

2019 - BERT, ALBERT, BERTSUM

2020 - RoBERTa, GPT3

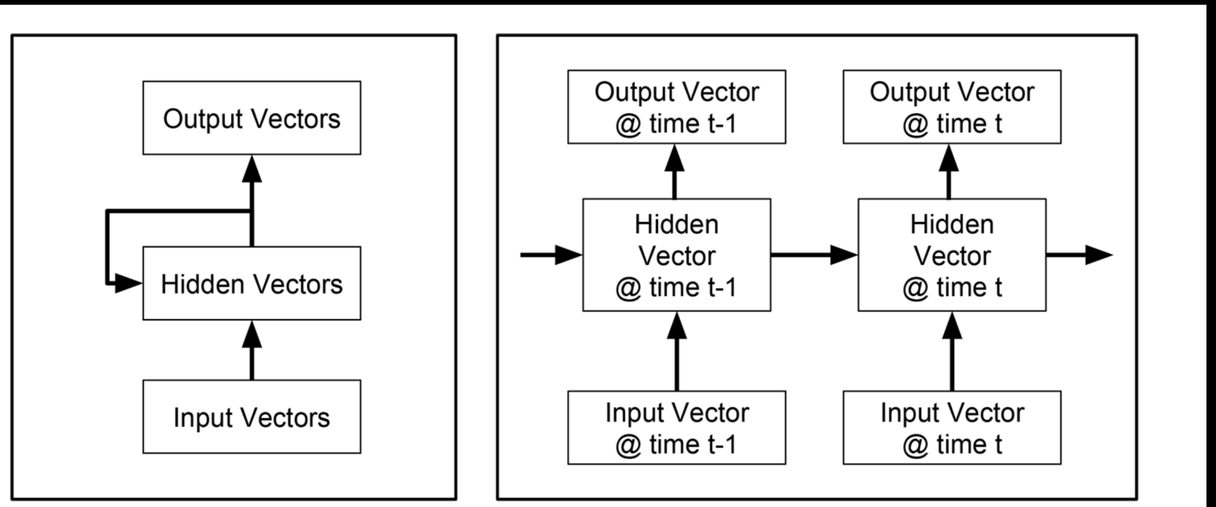
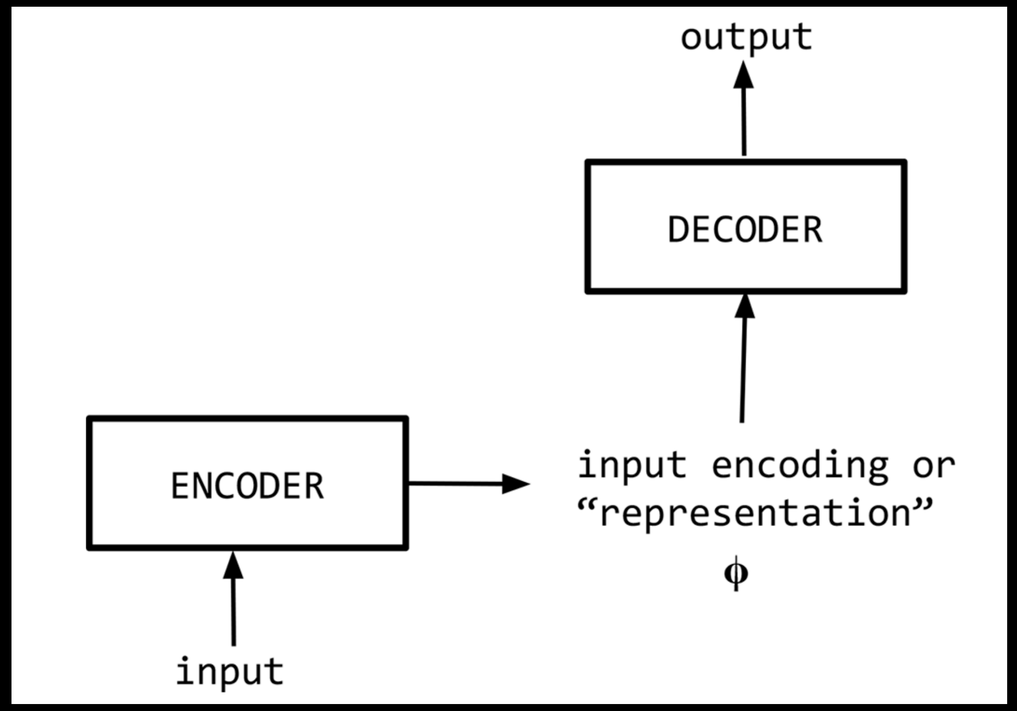


Figure 6-1. The functional view of the Elman RNN (left) displays the recurrent relationship as a feedback loop on the hidden vectors. The "unrolled" view (right) displays the computational relationship clearly; the hidden vector of each time step is dependent on both the input at that time step and the hidden vector from the previous time step.

(3)

This is done by maintaining a hidden state vector that captures the current state of the sequence. The hidden state vector is computed from both a current input vector and the previous hidden state vector.

Motivation
Encoder-Decoder Model



(3)

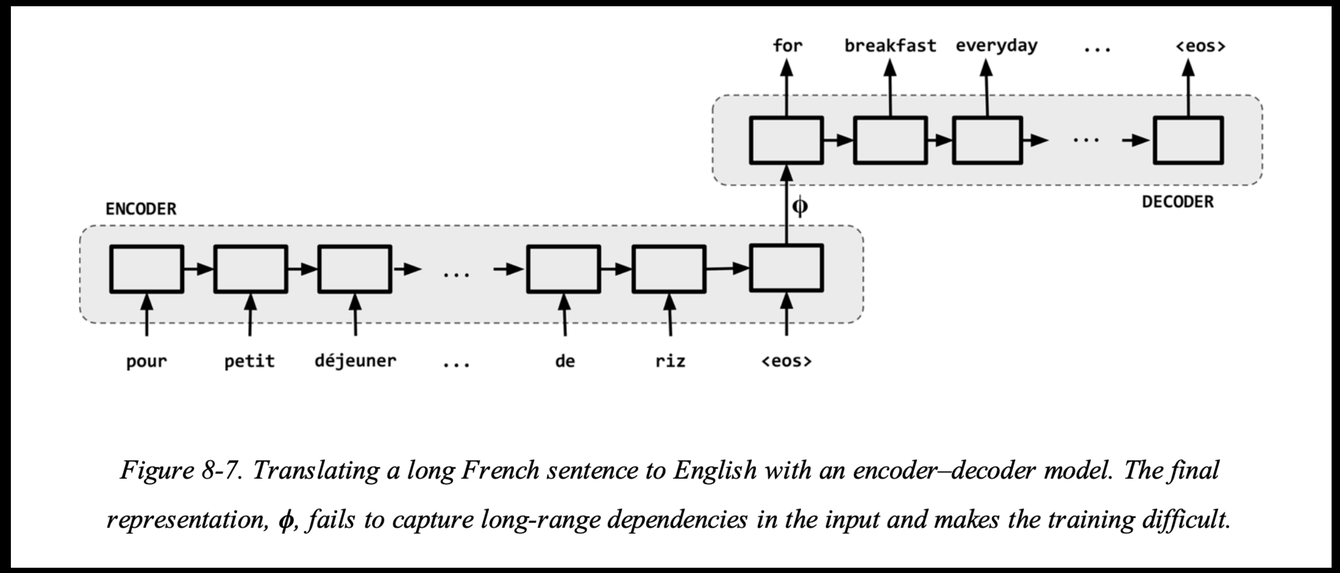
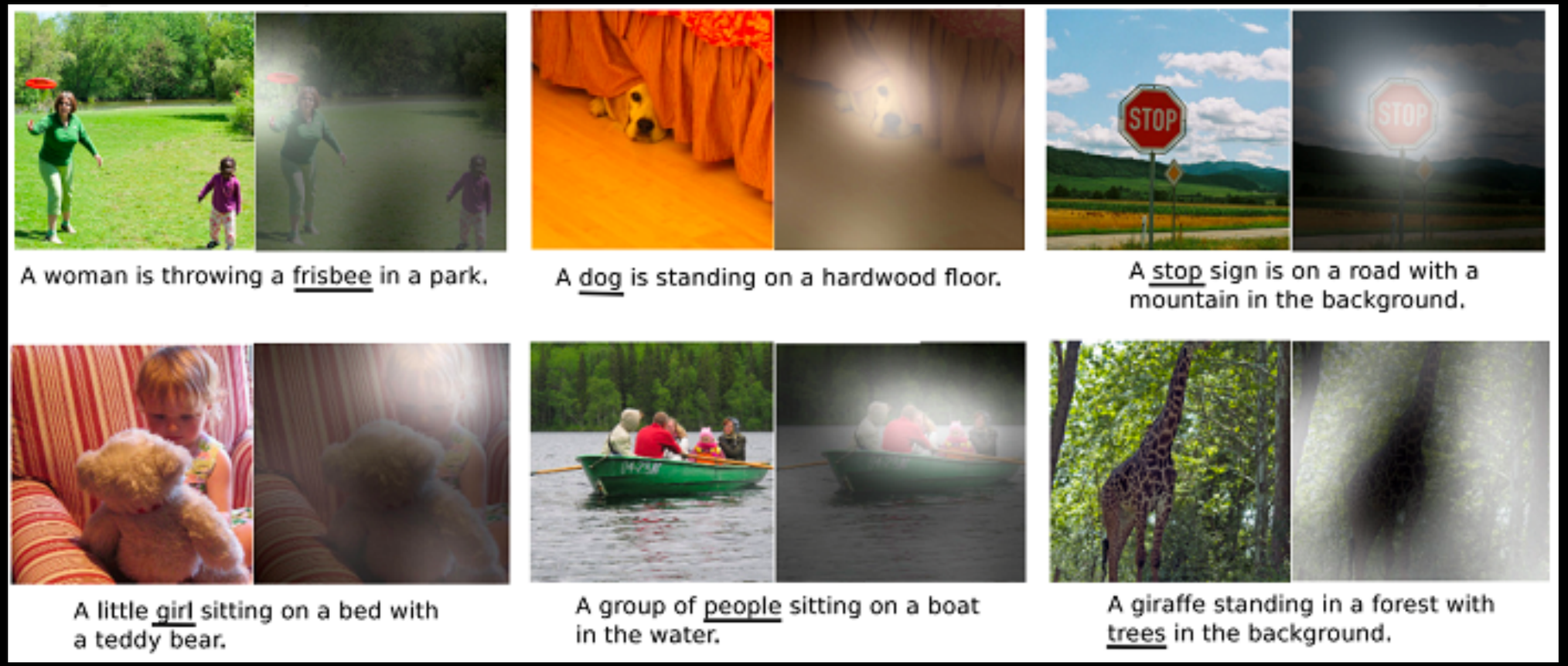


Figure 8-7. Translating a long French sentence to English with an encoder–decoder model. The final representation, ϕ, fails to capture long-range dependencies in the input and makes the training difficult.

(3)



Source – Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

BLEU

- Bilingual Evaluation Understudy, is a score for comparing a candidate translation of text to one or more reference translations.
- A perfect match results in a score of 1.0, whereas a perfect mismatch results in a score of 0.0.
- The approach works by counting matching n-grams in the candidate translation to n-grams in the reference text, where 1-gram or unigram would be each token and a bigram comparison would be each word pair. The comparison is made regardless of word order.
- First introduced by Kishore Papineni in 2002

Previous SOTA for Translation

Phrase-based Statistical Machine Translation

Phrase-based SMT

bacche	bagiche mein	khel rahe hai
Children	in garden	are playing
Children	are playing	in garden

Previous SOTA for Translation

Phrase-based Statistical Machine Translation

Phrase-based SMT

The main component
- Phrase Table
- Language Model

Source Sentences : तुम स्कूल चले जाओ

Source Phrase	Target Phrase	Score
चले जाओ	Go to	0.7
	Go away to	0.8
	Play to	0.1

Target Sentences	
You go to School	0.7 0.7 0.9
You go away to School	0.7 0.8 0.9
You play to School	0.7 0.1 0.9

Key Concept in the Paper

Variable Length Vectors

- Fixes the bottleneck of the encoder-decoder model
- Helps to align and translate jointly through a kind of soft-search

“It encodes the input sentence into a sequence of vectors and chooses a subset of these vectors adaptively while decoding the translation. This frees a neural translation model from having to squash all the information of a source sentence, regardless of its length, into a fixed-length vector. We show this allows a model to cope better with long sentences.” - Bahdanau et al.

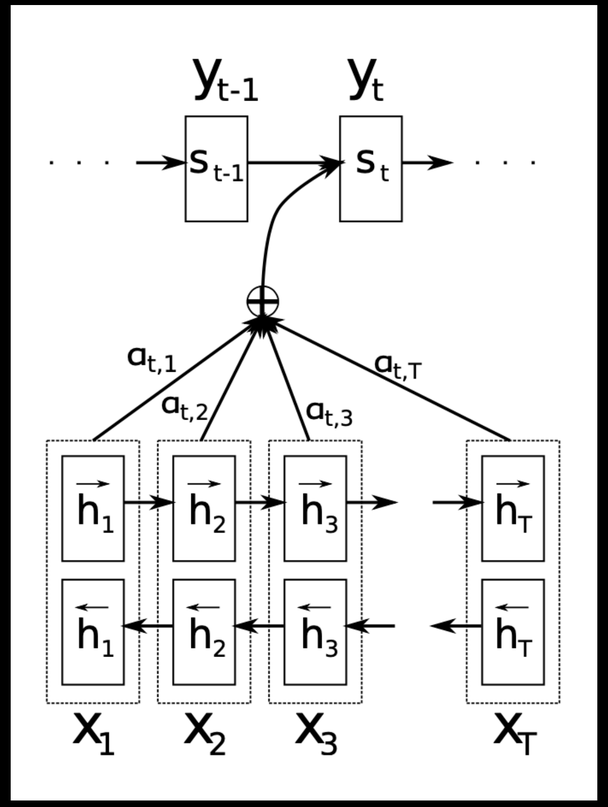
Basic Encoder-Decoder Model for NMT

$$p(\mathbf{y}) = \prod_{t=1}^{T} p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c),$$

$$p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c),$$

Proposed Model - Decoder Side

$$p(y_i|y_1,\ldots,y_{i-1},\mathbf{x}) = g(y_{i-1}, s_i, c_i),$$



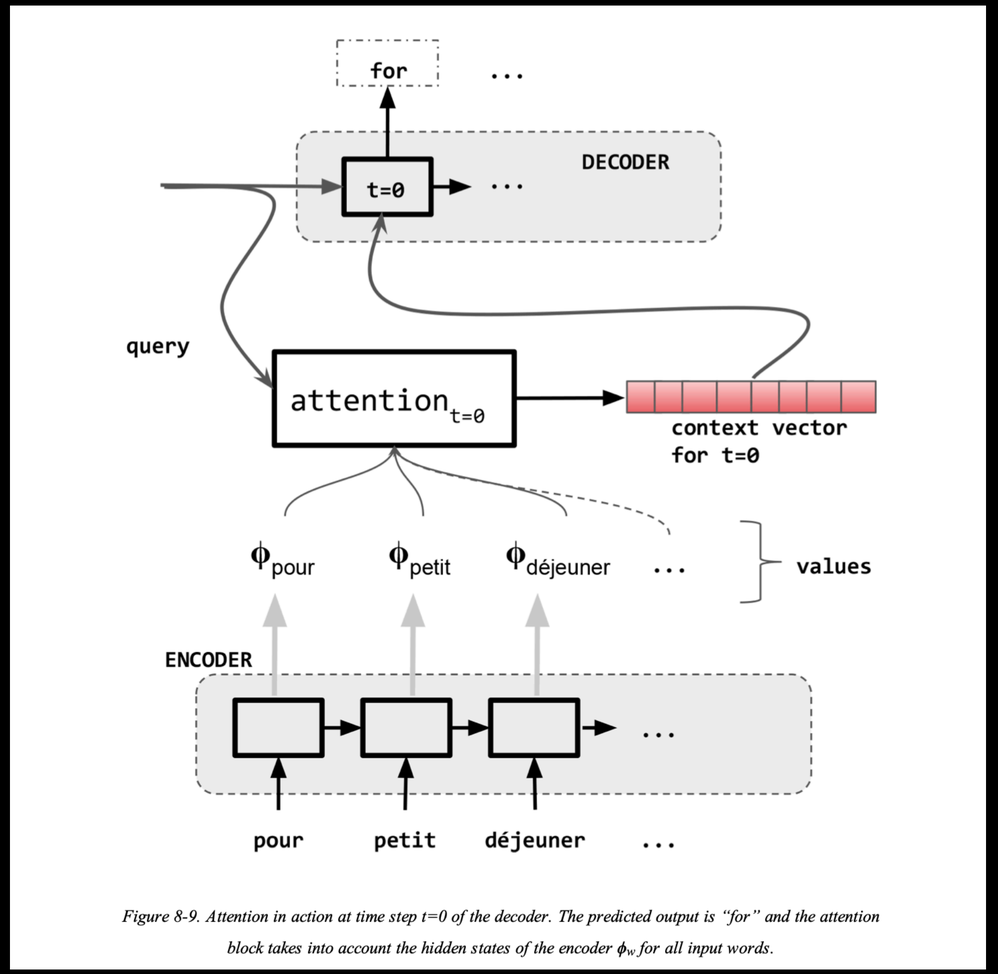


Figure 8-9. Attention in action at time step t=0 of the decoder. The predicted output is "for" and the attention block takes into account the hidden states of the encoder ϕ_w for all input words.

(3)

Proposed Model - Decoder Side

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T_x} \exp\left(e_{ik}\right)},$$

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

Proposed Model - Decoder Side

“By letting the decoder have an attention mechanism, we relieve the encoder from the burden of having to encode all information in the source sentence into a fixed- length vector. With this new approach the information can be spread throughout the sequence of annotations, which can be selectively retrieved by the decoder accordingly.” - Bahdanau et al.

Proposed Model - Encoder Side

Bidirectional RNN

"….in the proposed scheme, we would like the annotation of each word to summarize not only the preceding words, but also the following words. Hence, we propose to use a bidirectional RNN (BiRNN, Schuster and Paliwal, 1997), which has been successfully used recently in speech recognition (see, e.g., Graves *et al.*, 2013)." - Bahdanau et al.

A BiRNN consists of forward and backward RNN's. The forward RNN $\overrightarrow{f}$ reads the input sequence as it is ordered (from x_1 to x_{T_x}) and calculates a sequence of *forward hidden states* $(\overrightarrow{h}_1, \cdots, \overrightarrow{h}_{T_x})$. The backward RNN $\overleftarrow{f}$ reads the sequence in the reverse order (from x_{T_x} to x_1), resulting in a sequence of *backward hidden states* $(\overleftarrow{h}_1, \cdots, \overleftarrow{h}_{T_x})$.

Experiment Settings - Preprocessing

As proposed by the authors and in our implementation

- Dataset - ACL WMT ’14 English-French parallel corpora (850,000,000 words)
- Shortlist Top - 30,000 words in each language and map rest to ‘UNK’
- No other complex preprocessing techniques used, eg. stemming, lowercasing, etc.
- Max sentence length is 30 words

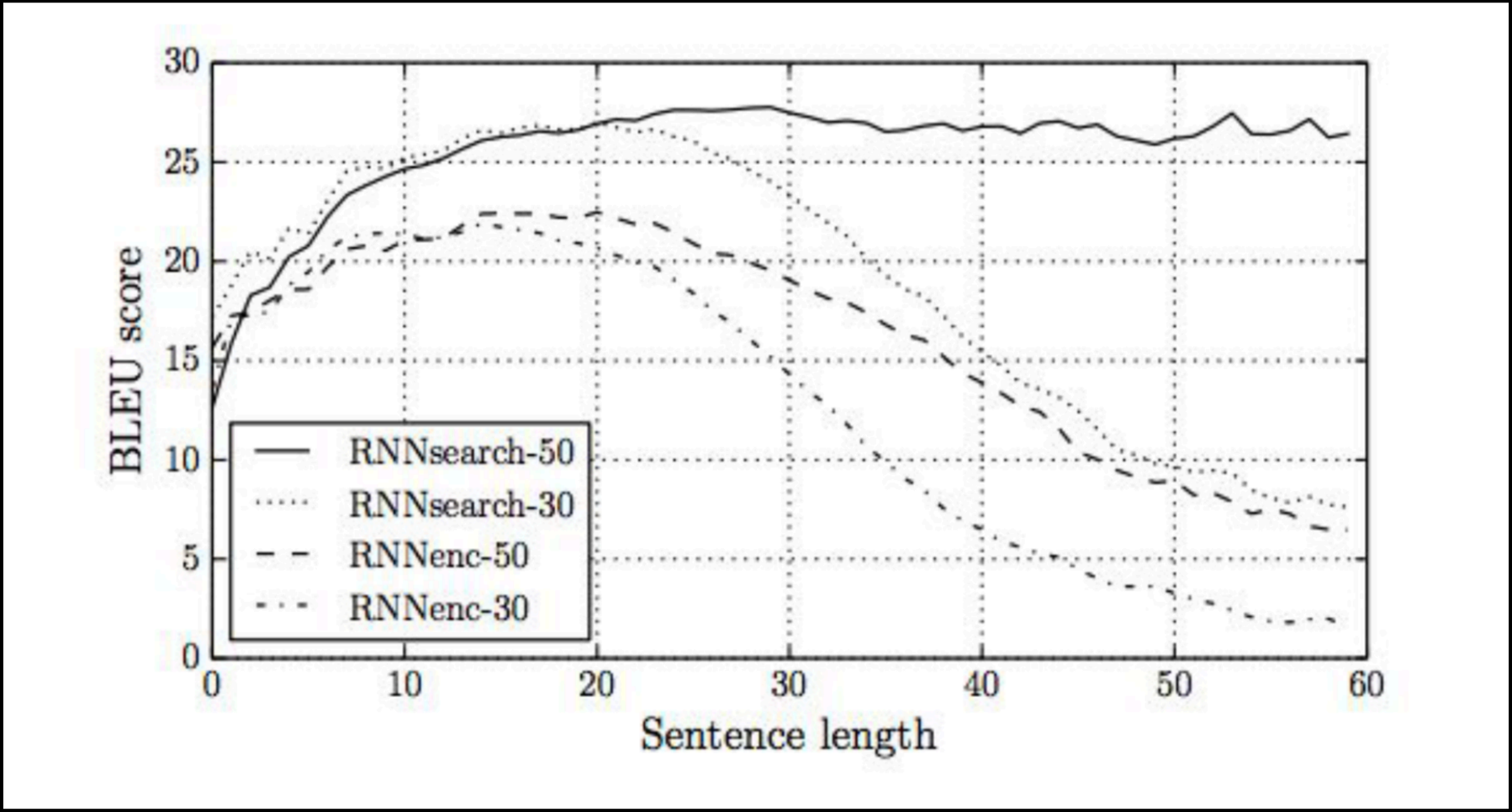
Experiment Settings

As proposed by the authors and in our implementation

- AdaDelta Optimizer with epsilon = 10^{-6} , rho = 0.95
- Minibatch SGD with batch_size = 80
- Embedding Dimension = 620
- Hidden Layer Size = 1000
- Output Layer Size = 500
- Weights initialization = RandomNormal with Mean = 0 and Standard Deviation = 0.001
- Bias initialization = Zeros
- Regularization - L2

“We trained each model for approximately 5 days” - Bahdanau et al.

Findings in the Paper



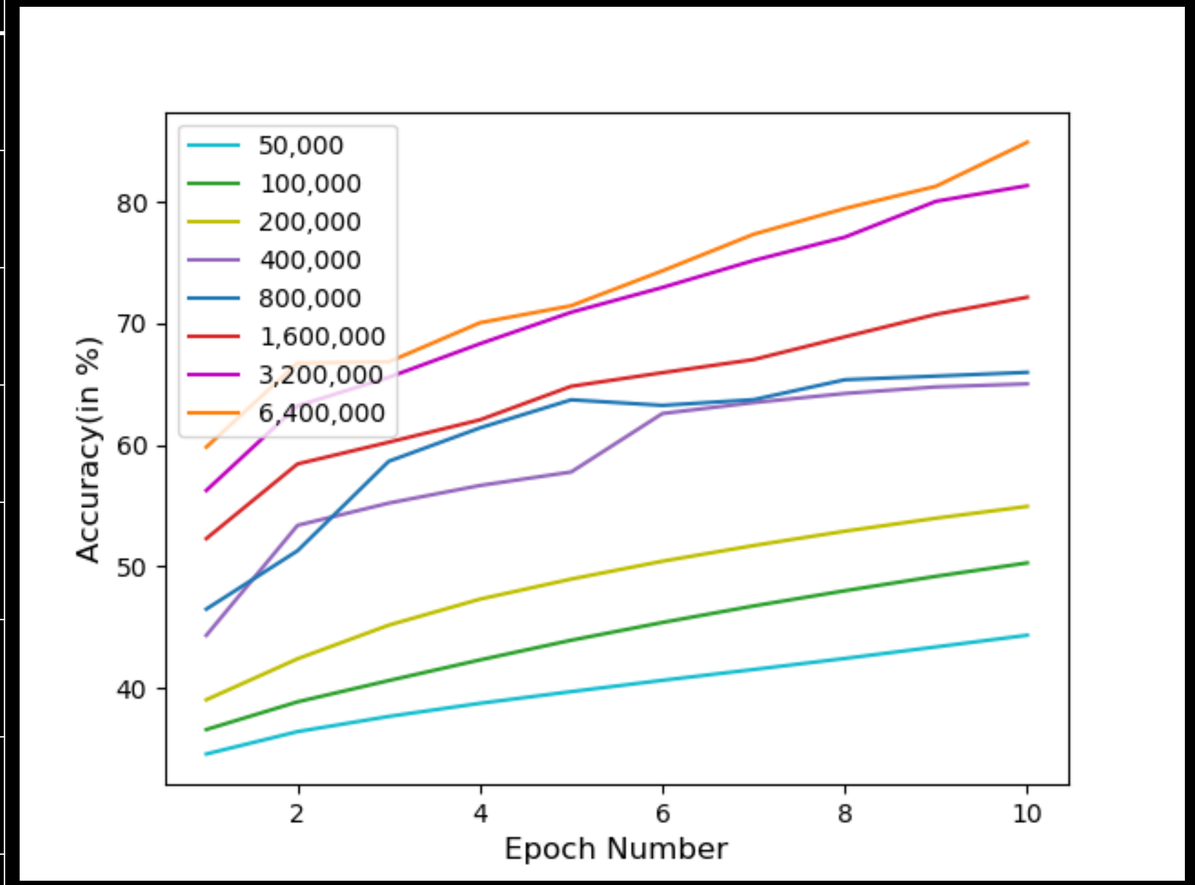
Conclusions in the Paper

“…the proposed approach achieved a translation performance comparable to the existing phrase-based statistical machine translation. It is a striking result, considering that the proposed architecture, or the whole family of neural machine translation, has only been proposed as recently as this year. We believe the architecture proposed here is a promising step toward better machine translation and a better understanding of natural languages in general.” - Bahdanau et al.

Our Training

Trained on Google Colab

Number of Sentences	Time per Epoch	Time for 10 Epochs	Accuracy
50,000	342s	57m	44.31%
100,000	673s	1h 53m	50.3%
200,000	1348s	3h 44m	54.93%
400,000	4800s	13h 27m	65.11%
800,000	6300s	17h 39m	65.96%
1,600,000	12300s	1d 10h 11m	72.14%
3,200,000	21330s	2d 11h 16m	81.32%
6,400,000	38241s	4d 17h 42m	84.89%



Our Training

Number of Parameters

```
Model: "encoder_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_8 (Embedding)      multiple                  18600000
_________________________________________________________________
bidirectional_4 (Bidirection multiple                  9732000
=================================================================
Total params: 28,332,000
Trainable params: 28,332,000
Non-trainable params: 0
_________________________________________________________________
```

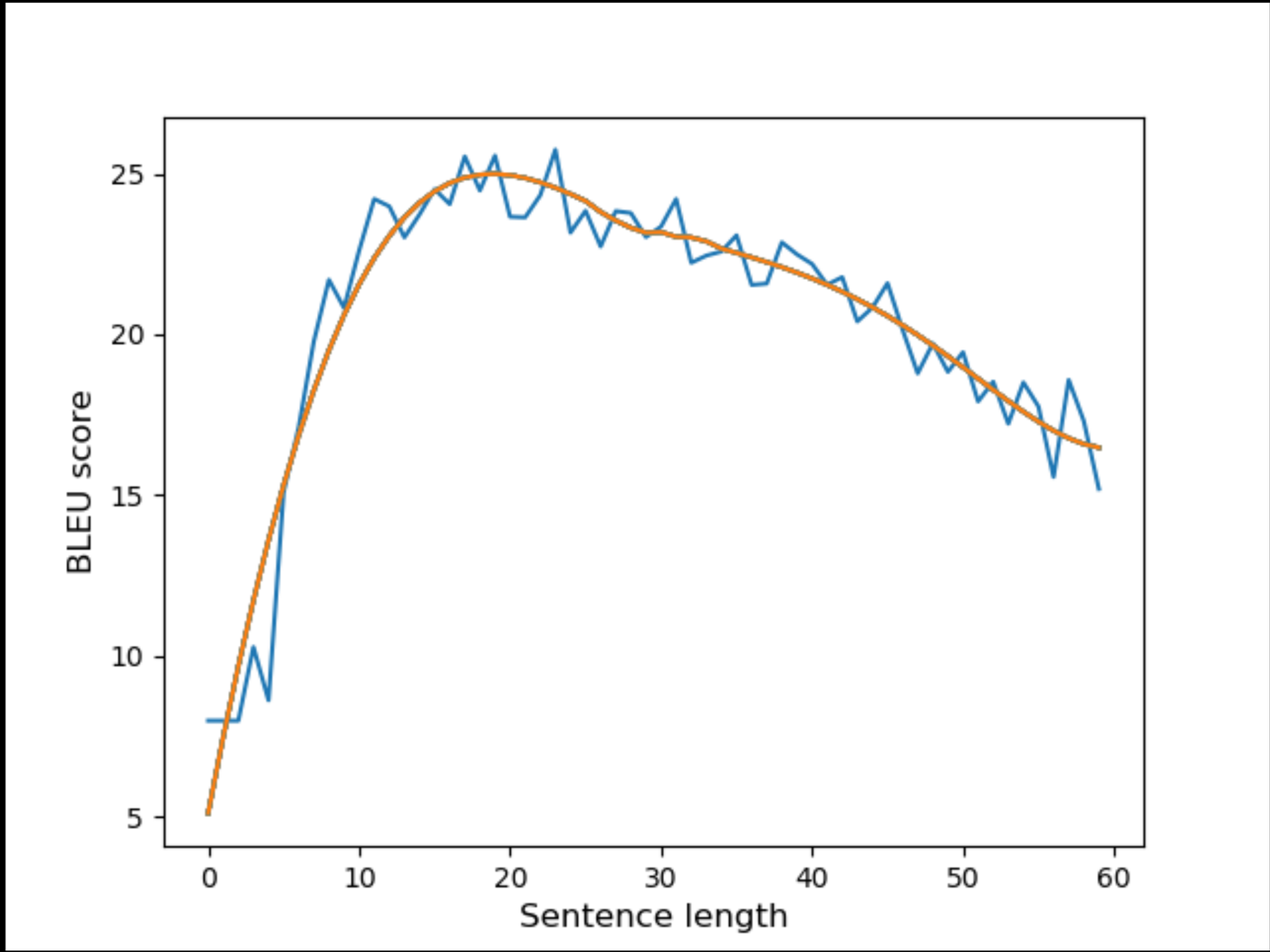
```
Model: "decoder_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_9 (Embedding)      multiple                  18600000
_________________________________________________________________
gru_9 (GRU)                  multiple                  10866000
_________________________________________________________________
attention_4 (Attention)      multiple                  3003001
_________________________________________________________________
dense_19 (Dense)             multiple                  30030000
=================================================================
Total params: 62,499,001
Trainable params: 62,499,001
Non-trainable params: 0
_________________________________________________________________
```

Our Training

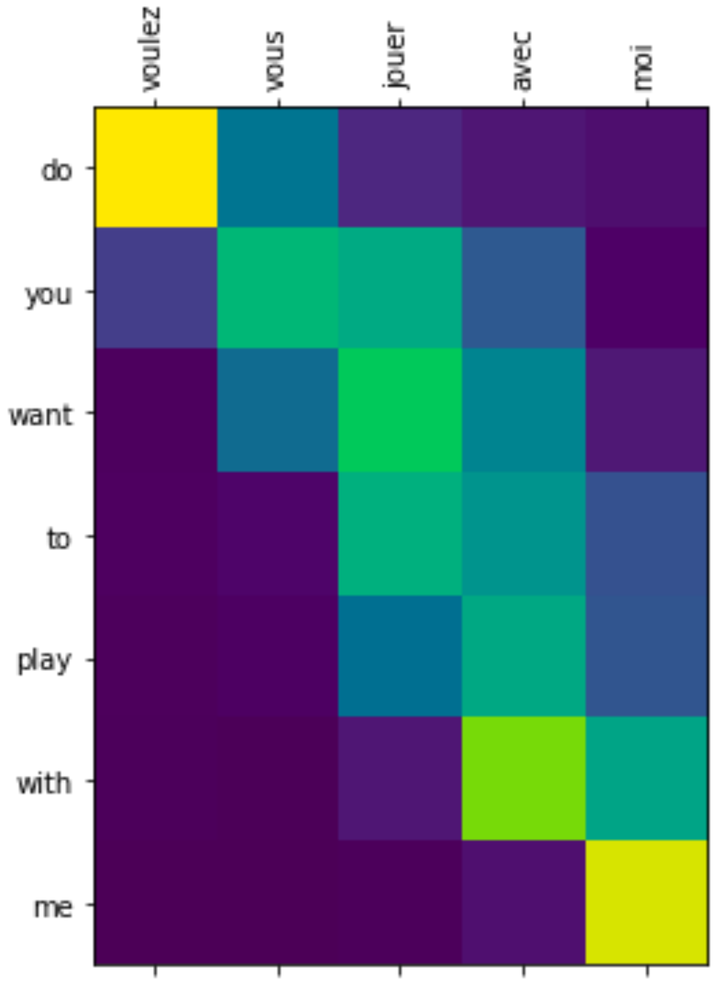
Total Time Trained =

9 days and 11 hours

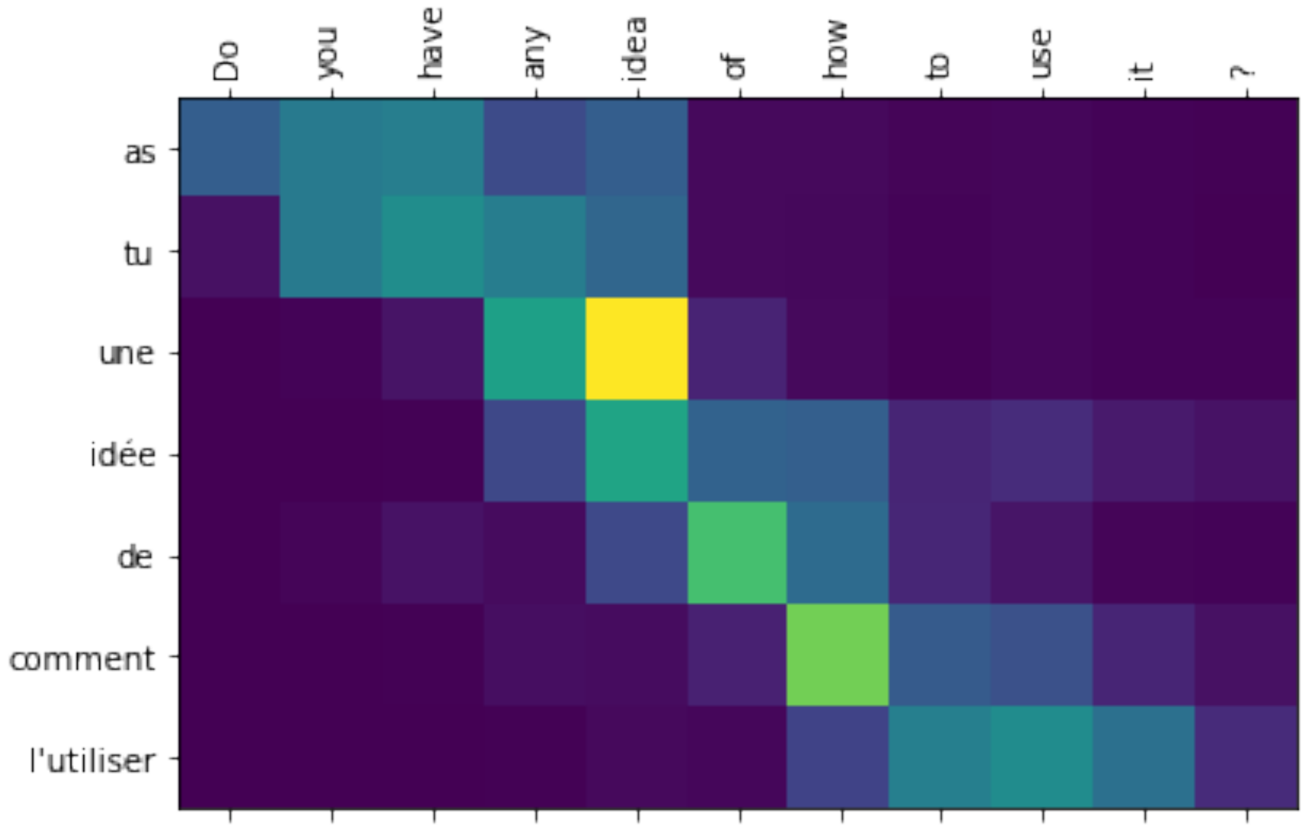
Our Results



Our Results




```
--------------------------------------------------------------------------------
FR :  c'est l'espace économique européen
EN :  this is the european economic area
OR :  this is the european economic area
--------------------------------------------------------------------------------
```



Future - Transformers

Attention is all you need!

- Parallelized
- Multi-headed Attention
- Skip-connections

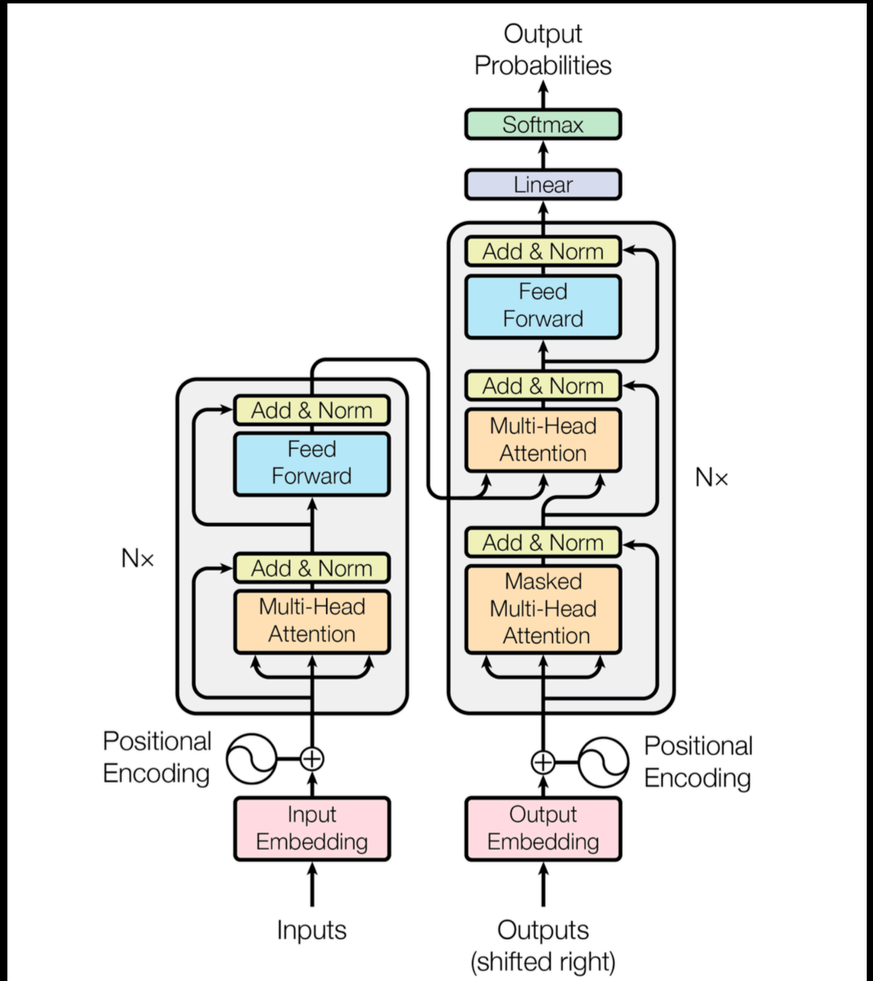


Figure 1: The Transformer - model architecture.

Future - T5

by Google

(4)

OpenAI's GPT3

Release - 28th May, 2020

- #Parameters - 175,000,000,000 (vs 1,500,000,000 in GPT2)
- GPT-3 is trained on the CommonCrawl data set of nearly 1,000,000,000,000 words collected between 2016 and 2019, as well as data sets related to web text, books, and Wikipedia.
- "In an assessment of associations between gender and occupation, GPT-3 demonstrated that it's most likely to suggest a male identifier, based on analysis of almost 400 occupations. A recent analysis of pre-trained language models found race, gender, occupation, and religious bias prevalent among pre-trained language models,…." - Venture Beat
- Beautiful SOTA Results (obviously, because they trained on 1,000,000,000,000 words)

Cost to train? **$12,000,000** (just for the electricity to run the GPUs)

References

Main Source - Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations (ICLR), 2015*

1. TensorFlow documentations - https://www.tensorflow.org/guide
2. TensorFlow tutorials - https://www.tensorflow.org/tutorials
3. Delip Rao and Brian McMohan - *Natural Language Processing with PyTorch (the book)*
4. Adam Roberts and Colin Raffel - Google Research - https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html
5. Prudhvi Potungati - CEO of *Avani.ai* and *SupervisedLearning.com*
6. Srihari Humbarwadi - Senior Computer Vision Engineer, *Bosch Research and Development, India*
7. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.
8. Khari Johnson, https://venturebeat.com/2020/05/29/openai-debuts-gigantic-gpt-3-language-model-with-175-billion-parameters/
9. Rishabh Kumar - PhD *IIT Bombay*
10. stackoverflow.com - *Our Lord Savior*

Huge Thank You to Dr. Venkateswara Rao Sir