Language & Attention
Steve Zhu
• Language Models are Few-Shot Learners

• Attention Is All You Need
What is a Language Model
Let’s go bucks and beat Michigan.
Input

Let’s go bucks and beat Michigan.

Or: Which word is the mostly likely to come next
Language Model:
A model that can generate language in a probabilistic way.

You can train it on any sort of text data.
Common Crawl Dataset

- Filtered for better quality
- A crawl of the entire Internet

GPT-3: An autoregressive language model
Language Models are Few-Shot Learners
Language Models are Few-Shot Learners

- CRAZYY size
- Transformer Model

<table>
<thead>
<tr>
<th>Model Name</th>
<th>( n_{\text{params}} )</th>
<th>( n_{\text{layers}} )</th>
<th>( d_{\text{model}} )</th>
<th>( n_{\text{heads}} )</th>
<th>( d_{\text{head}} )</th>
<th>( \text{Batch Size} )</th>
<th>( \text{Learning Rate} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-3 Small</td>
<td>125M</td>
<td>12</td>
<td>768</td>
<td>12</td>
<td>64</td>
<td>0.5M</td>
<td>( 6.0 \times 10^{-4} )</td>
</tr>
<tr>
<td>GPT-3 Medium</td>
<td>350M</td>
<td>24</td>
<td>1024</td>
<td>16</td>
<td>64</td>
<td>0.5M</td>
<td>( 3.0 \times 10^{-4} )</td>
</tr>
<tr>
<td>GPT-3 Large</td>
<td>760M</td>
<td>24</td>
<td>1536</td>
<td>16</td>
<td>96</td>
<td>0.5M</td>
<td>( 2.5 \times 10^{-4} )</td>
</tr>
<tr>
<td>GPT-3 XL</td>
<td>1.3B</td>
<td>24</td>
<td>2048</td>
<td>24</td>
<td>128</td>
<td>1M</td>
<td>( 2.0 \times 10^{-4} )</td>
</tr>
<tr>
<td>GPT-3 2.7B</td>
<td>2.7B</td>
<td>32</td>
<td>2560</td>
<td>32</td>
<td>80</td>
<td>1M</td>
<td>( 1.6 \times 10^{-4} )</td>
</tr>
<tr>
<td>GPT-3 6.7B</td>
<td>6.7B</td>
<td>32</td>
<td>4096</td>
<td>32</td>
<td>128</td>
<td>2M</td>
<td>( 1.2 \times 10^{-4} )</td>
</tr>
<tr>
<td>GPT-3 13B</td>
<td>13.0B</td>
<td>40</td>
<td>5140</td>
<td>40</td>
<td>128</td>
<td>2M</td>
<td>( 1.0 \times 10^{-4} )</td>
</tr>
<tr>
<td>GPT-3 175B or “GPT-3”</td>
<td>175.0B</td>
<td>96</td>
<td>12288</td>
<td>96</td>
<td>128</td>
<td>3.2M</td>
<td>( 0.6 \times 10^{-4} )</td>
</tr>
</tbody>
</table>
Transformer Model:
• Input Context (we already have)
Let’s go bucks and beat Michigan.
Transformer Model:

- Input Context (we already have)

Let’s go bucks and beat Michigan.

Attention Mechanism
Attention Mechanism:

• A way where information is routed in between the different tokens. As it goes up the layer, the information is routed around, and the model can make various inferences and at the end, the model is supposed to come up next word.

Let’s go bucks and beat Michigan.
How did they train it
Traditional Fine-tuning (Not used for GPT-3)

**Fine-tuning**
The model is trained via repeated gradient updates using a large corpus of example tasks.

1. sea otter => loutre de mer
   - gradient update
   - example #1

2. peppermint => menthe poivrée
   - gradient update
   - example #2

3. plush giraffe => giraffe peluche
   - gradient update
   - example #N

4. cheese => ...........................................
   - prompt

**BERT:**
1. Pretrain
2. Fine-tuning

- Train Set
- Test Set
True Zero-shot

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

1  Translate English to French:  ⇐ task description
2  cheese =>  .........................  ⇐ prompt
One-shot (example comes from training set but not train on it)

**One-shot**

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

1. **task description**: Translate English to French:

2. **example**: sea otter => loutre de mer

3. **prompt**: cheese => ......................................
Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

1. Translate English to French:
2. sea otter => loutre de mer
3. peppermint => menthe poivrée
4. plush giraffe => girafe peluche
5. cheese =>

(task description)
(examples)
(prompt)
Example:
Give the Chinese letters for the numbers
One -> 一
Two -> 二
Three -> 三
Four -> 四
Results
Language Models are Few-Shot Learners

With the parameters number goes up, the Validation Loss goes down. (parameters in log scale)

You can make improvements by scaling up your model on language model.
Individual Tasks:

Alice was friends with Bob. Alice went to visit her friend _______. -> Bob

George bought some baseball equipment, a ball, a glove, and a __________. ->
Question Answering:
Open-Domain means that the model can go and look at some Wikipedia page.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Natural QS</th>
<th>Web QS</th>
<th>TriviaQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAG (Fine-tuned, Open-Domain) [LPP+20]</td>
<td>44.5</td>
<td>45.5</td>
<td>68.0</td>
</tr>
<tr>
<td>T5-11B+SSM (Fine-tuned, Closed-Book) [RRS20]</td>
<td>36.6</td>
<td>44.7</td>
<td>60.5</td>
</tr>
<tr>
<td>T5-11B (Fine-tuned, Closed-Book)</td>
<td>34.5</td>
<td>37.4</td>
<td>50.1</td>
</tr>
<tr>
<td>GPT-3 Zero-Shot</td>
<td>14.6</td>
<td>14.4</td>
<td>64.3</td>
</tr>
<tr>
<td>GPT-3 One-Shot</td>
<td>23.0</td>
<td>25.3</td>
<td>68.0</td>
</tr>
<tr>
<td>GPT-3 Few-Shot</td>
<td>29.9</td>
<td>41.5</td>
<td>71.2</td>
</tr>
</tbody>
</table>
Translation:
<table>
<thead>
<tr>
<th>Setting</th>
<th>En→Fr</th>
<th>Fr→En</th>
<th>En→De</th>
<th>De→En</th>
<th>En→Ro</th>
<th>Ro→En</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOTA (Supervised)</td>
<td>45.6&lt;sup&gt;a&lt;/sup&gt;</td>
<td>35.0&lt;sup&gt;b&lt;/sup&gt;</td>
<td>41.2&lt;sup&gt;c&lt;/sup&gt;</td>
<td>40.2&lt;sup&gt;d&lt;/sup&gt;</td>
<td>38.5&lt;sup&gt;e&lt;/sup&gt;</td>
<td>39.9&lt;sup&gt;e&lt;/sup&gt;</td>
</tr>
<tr>
<td>XLM [LC19]</td>
<td>33.4</td>
<td>33.3</td>
<td>26.4</td>
<td>34.3</td>
<td>33.3</td>
<td>31.8</td>
</tr>
<tr>
<td>MASS [STQ+19]</td>
<td>37.5</td>
<td>34.9</td>
<td>28.3</td>
<td>35.2</td>
<td>35.2</td>
<td>33.1</td>
</tr>
<tr>
<td>mBART [LGG+20]</td>
<td>-</td>
<td>-</td>
<td>29.8</td>
<td>34.0</td>
<td>35.0</td>
<td>30.5</td>
</tr>
<tr>
<td>GPT-3 Zero-Shot</td>
<td>25.2</td>
<td>21.2</td>
<td>24.6</td>
<td>27.2</td>
<td>14.1</td>
<td>19.9</td>
</tr>
<tr>
<td>GPT-3 One-Shot</td>
<td>28.3</td>
<td>33.7</td>
<td>26.2</td>
<td>30.4</td>
<td>20.6</td>
<td>38.6</td>
</tr>
<tr>
<td>GPT-3 Few-Shot</td>
<td>32.6</td>
<td>39.2</td>
<td>29.7</td>
<td>40.6</td>
<td>21.0</td>
<td>39.5</td>
</tr>
</tbody>
</table>
Winograd: The Winograd Schemas Challenge is a classical task in NLP that involves determining which word a pronoun refers to, when the pronoun is grammatically ambiguous but semantically unambiguous to a human.
Language Models are Few-Shot Learners
Physical Q&A:

- ARC (dataset of multiple-choice questions collected from 3rd to 9th grade)
- Asks common sense questions about how the physical world works and is intended as a probe of grounded understanding of the world.
Language Models are Few-Shot Learners

### PhysicalQA

<table>
<thead>
<tr>
<th>Setting</th>
<th>PIQA</th>
<th>ARC (Easy)</th>
<th>ARC (Challenge)</th>
<th>OpenBookQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fine-tuned SOTA</td>
<td>79.4</td>
<td>92.0[KKS+20]</td>
<td>78.5[KKS+20]</td>
<td>87.2[KKS+20]</td>
</tr>
<tr>
<td>GPT-3 Zero-Shot</td>
<td><strong>80.5*</strong></td>
<td>68.8</td>
<td>51.4</td>
<td>57.6</td>
</tr>
<tr>
<td>GPT-3 One-Shot</td>
<td><strong>80.5*</strong></td>
<td>71.2</td>
<td>53.2</td>
<td>58.8</td>
</tr>
<tr>
<td>GPT-3 Few-Shot</td>
<td><strong>82.8*</strong></td>
<td>70.1</td>
<td>51.5</td>
<td>65.4</td>
</tr>
</tbody>
</table>
Reading Comprehension:
Reading Comprehension:

- abstractive, multiple choice, and span-based answer formats in both dialog and single question settings.

<table>
<thead>
<tr>
<th></th>
<th>SuperGLUE Average</th>
<th>BoolQ Accuracy</th>
<th>CB Accuracy</th>
<th>CB F1</th>
<th>COPA Accuracy</th>
<th>RTE Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fine-tuned SOTA</td>
<td>89.0</td>
<td>91.0</td>
<td>96.9</td>
<td>93.9</td>
<td>94.8</td>
<td>92.5</td>
</tr>
<tr>
<td>Fine-tuned BERT-Large</td>
<td>69.0</td>
<td>77.4</td>
<td>83.6</td>
<td>75.7</td>
<td>70.6</td>
<td>71.7</td>
</tr>
<tr>
<td>GPT-3 Few-Shot</td>
<td>71.8</td>
<td>76.4</td>
<td>75.6</td>
<td>52.0</td>
<td>92.0</td>
<td>69.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>WiC Accuracy</th>
<th>WSC Accuracy</th>
<th>MultiRC Accuracy</th>
<th>MultiRC F1a</th>
<th>ReCoRD Accuracy</th>
<th>ReCoRD F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fine-tuned SOTA</td>
<td>76.1</td>
<td>93.8</td>
<td>62.3</td>
<td>88.2</td>
<td>92.5</td>
<td>93.3</td>
</tr>
<tr>
<td>Fine-tuned BERT-Large</td>
<td>69.6</td>
<td>64.6</td>
<td>24.1</td>
<td>70.0</td>
<td>71.3</td>
<td>72.0</td>
</tr>
<tr>
<td>GPT-3 Few-Shot</td>
<td>49.4</td>
<td>80.1</td>
<td>30.5</td>
<td>75.4</td>
<td>90.2</td>
<td>91.1</td>
</tr>
</tbody>
</table>
_BoolQ Dataset:

{ "question": "is france the same timezone as the uk",
  "passage": "At the Liberation of France in the summer of 1944, Metropolitan France kept GMT+2 as it was the time then used by the Allies (British Double Summer Time). In the winter of 1944--1945, Metropolitan France switched to GMT+1, same as in the United Kingdom, and switched again to GMT+2 in April 1945 like its British ally. In September 1945, Metropolitan France returned to GMT+1 (pre-war summer time), which the British had already done in July 1945. Metropolitan France was officially scheduled to return to GMT+0 on November 18, 1945 (the British returned to GMT+0 in on October 7, 1945), but the French government canceled the decision on November 5, 1945, and GMT+1 has since then remained the official time of Metropolitan France.
  
  "answer": false,
  "title": "Time in France", }
COPA Dataset:

1. Examples

Premise: The man broke his toe. What was the CAUSE of this?
Alternative 1: He got a hole in his sock.
Alternative 2: He dropped a hammer on his foot.

Premise: I tipped the bottle. What happened as a RESULT?
Alternative 1: The liquid in the bottle froze.
Alternative 2: The liquid in the bottle poured out.

Premise: I knocked on my neighbor’s door. What happened as a RESULT?
Alternative 1: My neighbor invited me in.
Alternative 2: My neighbor left his house.
NLI (Natural Language Inference):
• Also poorly performed
• Concerns the ability to understand the relationship between two sentences.
Synthetic and Qualitative Tasks:

- Arithmetic
- Word Scrambling and Manipulation Tasks
- SAT Analogies
- News Article Generation
- Learning and Using Novel Words
- Correcting English Grammar
Arithmetic:

- 2 digit addition (2D+) – The model is asked to add two integers sampled uniformly from [0; 100), phrased in the form of a question, e.g. “Q: What is 48 plus 76? A: 124.”
- 2 digit subtraction (2D-) – The model is asked to subtract two integers sampled uniformly from [0; 100); the answer may be negative. Example: “Q: What is 34 minus 53? A: -19”.
- 3 digit addition (3D+) – Same as 2 digit addition, except numbers are uniformly sampled from [0; 1000).
- 3 digit subtraction (3D-) – Same as 2 digit subtraction, except numbers are uniformly sampled from [0; 1000).
- 4 digit addition (4D+) – Same as 3 digit addition, except uniformly sampled from [0; 10000).
- 4 digit subtraction (4D-) – Same as 3 digit subtraction, except uniformly sampled from [0; 10000).
- 5 digit addition (5D+) – Same as 3 digit addition, except uniformly sampled from [0; 100000).
- 5 digit subtraction (5D-) – Same as 3 digit subtraction, except uniformly sampled from [0; 100000).
- 2 digit multiplication (2Dx) – The model is asked to multiply two integers sampled uniformly from [0; 100), e.g. “Q: What is 24 times 42? A: 1008”.
- One-digit composite (1DC) – The model is asked to perform a composite operation on three 1 digit numbers, with parentheses around the last two. For example, “Q: What is 6+(4*8)? A: 38”. The three 1 digit numbers are selected uniformly on [0; 10) and the operations are selected uniformly from f+,-,*g.
Language Models are Few-Shot Learners

![Graph showing the performance of language models in arithmetic tasks](image)

<table>
<thead>
<tr>
<th>Setting</th>
<th>2D+</th>
<th>2D-</th>
<th>3D+</th>
<th>3D-</th>
<th>4D+</th>
<th>4D-</th>
<th>5D+</th>
<th>5D-</th>
<th>2Dx</th>
<th>1DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-3 Zero-shot</td>
<td>76.9</td>
<td>58.0</td>
<td>34.2</td>
<td>48.3</td>
<td>4.0</td>
<td>7.5</td>
<td>0.7</td>
<td>0.8</td>
<td>19.8</td>
<td>9.8</td>
</tr>
<tr>
<td>GPT-3 One-shot</td>
<td>99.6</td>
<td>86.4</td>
<td>65.5</td>
<td>78.7</td>
<td>14.0</td>
<td>14.0</td>
<td>3.5</td>
<td>3.8</td>
<td>27.4</td>
<td>14.3</td>
</tr>
<tr>
<td>GPT-3 Few-shot</td>
<td>100.0</td>
<td>98.9</td>
<td>80.4</td>
<td>94.2</td>
<td>25.5</td>
<td>26.8</td>
<td>9.3</td>
<td>9.9</td>
<td>29.2</td>
<td>21.3</td>
</tr>
</tbody>
</table>
Word Scrambling and Manipulation Tasks

• Cycle letters in word (CL) – The model is given a word with its letters cycled, then the “=” symbol, and is expected to generate the original word. For example, it might be given “lyinevitab” and should output “inevitably”.
• Anagrams of all but first and last characters (A1) – The model is given a word where every letter except the first and last have been scrambled randomly, and must output the original word. Example: criroptuon = corruption.
• Anagrams of all but first and last 2 characters (A2) – The model is given a word where every letter except the first 2 and last 2 have been scrambled randomly, and must recover the original word. Example: opoepnnt !opponent.
• Random insertion in word (RI) – A random punctuation or space character is inserted between each letter of a word, and the model must output the original word. Example: s.u!c/c!e.s s i/o/n = succession.
• Reversed words (RW) – The model is given a word spelled backwards, and must output the original word. Example: stcejbo!objects.
Language Models are Few-Shot Learners

$K = 100$. 

![Graph showing wordscramble accuracy vs parameters in LM (Billions)]
SAT Analogies:

• Example: “audacious is to boldness as (a) sanctimonious is to hypocrisy, (b) anonymous is to identity, (c) remorseful is to misdeed, (d) deleterious is to result, (e) impressionable is to temptation”.

• On this task GPT-3 achieves 65.2% in the few-shot setting, 59.1% in the one-shot setting, and 53.7% in the zero-shot setting, whereas the average score among college applicants was 57%
News Article Generation:

• They gave it a beginning of a news article and they let human decide whether the news article is written by a machine or by a human

• By contrast, mean human accuracy at detecting articles that were produced by the 175B parameter model was barely above chance at 52%. Human abilities to detect model generated text appear to decrease as model size increases: there appears to be a trend towards chance accuracy with model size, and human detection of GPT-3 is close to chance
Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination

Article: After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church’s annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.

The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.,” but that it has been shrinking in recent decades. The new split will be the second in the church’s history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.
Learning and Using Novel Words:

A "Burringo" is a car with very fast acceleration. An example of a sentence that uses the word Burringo is:
In our garage we have a Burringo that my father drives to work every day.

A "Gigamuru" is a type of Japanese musical instrument. An example of a sentence that uses the word Gigamuru is:
I have a Gigamuru that my uncle gave me as a gift. I love to play it at home.

To "screeg" something is to swing a sword at it. An example of a sentence that uses the word screeg is:
We screeged at each other for several minutes and then we went outside and ate ice cream.
## Correcting English Grammar:

<table>
<thead>
<tr>
<th>Poor English input</th>
<th>Good English output</th>
</tr>
</thead>
<tbody>
<tr>
<td>I eated the purple berries.</td>
<td>I ate the purple berries.</td>
</tr>
<tr>
<td>Thank you for picking me as your designer. I’d appreciate it.</td>
<td>Thank you for choosing me as your designer. I appreciate it.</td>
</tr>
<tr>
<td>The mentioned changes have done. or I did the alteration that you requested. or I changed things you wanted and did the modifications.</td>
<td>The requested changes have been made. or I made the alteration that you requested. or I changed things you wanted and made the modifications.</td>
</tr>
<tr>
<td>I’d be more than happy to work with you in another project.</td>
<td>I’d be more than happy to work with you on another project.</td>
</tr>
<tr>
<td>Please provide me with a short brief of the design you’re looking for and that’d be nice if you could share some examples or projects you did before.</td>
<td>Please provide me with a brief description of the design you’re looking for and that would be nice if you could share some examples or projects you have done before.</td>
</tr>
<tr>
<td>The patient was died.</td>
<td>The patient died.</td>
</tr>
<tr>
<td>We think that Leslie likes ourselves.</td>
<td>We think that Leslie likes us.</td>
</tr>
<tr>
<td>Janet broke Bill on the finger.</td>
<td>Janet broke Bill’s finger.</td>
</tr>
<tr>
<td>Mary arranged for, in St. Louis, John to rent a house cheap.</td>
<td>Mary arranged for John to rent a house in St. Louis.</td>
</tr>
<tr>
<td>Today I have went to the store to to buys some many bottle of water.</td>
<td>Today I went to the store to buy some bottles of water.</td>
</tr>
<tr>
<td>I have tried to hit ball with bat, but my swing is has miss.</td>
<td>I tried to hit the ball with the bat, but my swing missed.</td>
</tr>
</tbody>
</table>
Limitations
• Language Models are Few-Shot Learners

• Attention Is All You Need
Traditional RNN
The cat eats the mouse.
Traditional RNN

The cat eats the mouse.

The
- Word Vector
- NN
- Hidden States

The
- Word Vector
- NN
- Hidden States

Eats
- Word Vector
- NN
- Hidden States

The
- Word Vector
- NN
- Hidden States

Mouse
- Word Vector
- NN
- Hidden States
Traditional RNN

The cat eats the mouse.

Die Katze Firsst Die Maus
Transformer
Attention Is All You Need
Attention Is All You Need

INPUT

Je suis étudiant

ENCODERS

OUTPUT

I am a student

DECODERS
Attention$(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$

$Z$

$\text{FFN}(Z) = \max(0, ZW_1 + b_1)W_2 + b_2 \quad (2)$
Attention Is All You Need
1. Self-Attention: Relationship between current translation and previous
2. Encoder-Decoder Attention: Relationship between current translation and feature vectors
Bringing the Tensors into the Picture
Each word is embedded into a vector of size 512. We'll represent those vectors with these simple boxes.
Attention Is All You Need
Encoding
Attention Is All You Need
Self-Attention
The animal didn’t cross the street because it was too tired
Attention Is All You Need

<table>
<thead>
<tr>
<th>Input</th>
<th>Thinking</th>
<th>Machines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queries</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keys</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Values</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Score</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Embedding:
  - $x_1$ (green)
  - $x_2$ (green)

- Queries:
  - $q_1$ (purple)
  - $q_2$ (purple)

- Keys:
  - $k_1$ (orange)
  - $k_2$ (orange)

- Values:
  - $v_1$ (blue)
  - $v_2$ (blue)

- Score:
  - $q_1 \cdot k_1 = 112$
  - $q_1 \cdot k_2 = 96$
Attention Is All You Need

Input

Embedding

Queries

Keys

Values

Score

Divide by 8 ($\sqrt{d_k}$)

Softmax

Thinking

x₁

q₁

k₁

v₁

q₁ \cdot k₁ = 112

Machines

x₂

q₂

k₂

v₂

q₁ \cdot k₂ = 96

14

12

0.88

0.12
Attention Is All You Need

Input
Embedding
Queries
Keys
Values
Score
Divide by $8 \left( \sqrt{d_k} \right)$
Softmax
Softmax $\times$
Value
Sum

Thinking

Machines

$x_1$
$q_1$
$k_1$
$v_1$

$x_2$
$q_2$
$k_2$
$v_2$

$q_1 \cdot k_1 = 112$
$q_1 \cdot k_2 = 96$

14
0.88

12
0.12

$v_1$
$z_1$
$v_2$
$z_2$
To Sum up:

1. Embedding of each word
2. Get Q, K, V based on embedded vector
3. Calculate a score
4. Divide the scores by $\sqrt{d_k}$
5. Pass the result through a softmax operation
6. Multiply each value vector by the softmax score
7. Sum up the weighted value vectors
Attention Is All You Need
The self-attention calculation in matrix form
Multi-Head Attention
Attention Is All You Need

Calculating attention separately in eight different attention heads

Thinking Machines

ATTENTION HEAD #0

Z_0

ATTENTION HEAD #1

Z_1

ATTENTION HEAD #7

Z_7
1) Concatenate all the attention heads

\[
\begin{bmatrix}
Z_0 & Z_1 & Z_2 & Z_3 & Z_4 & Z_5 & Z_6 & Z_7
\end{bmatrix}
\]

2) Multiply with a weight matrix \( W^o \) that was trained jointly with the model

\[
X \cdot W^o
\]

3) The result would be the \( Z \) matrix that captures information from all the attention heads. We can send this forward to the FFNN

\[
Z
\]
1) This is our input sentence
2) We embed each word
3) Split into 8 heads. We multiply $X$ or $R$ with weight matrices
4) Calculate attention using the resulting $Q/K/V$ matrices
5) Concatenate the resulting $Z$ matrices, then multiply with weight matrix $W^o$ to produce the output of the layer

* In all encoders other than $#0$, we don't need embedding. We start directly with the output of the encoder right below this one.
Attention Is All You Need
Layer: 5  
Attention: Input - Input

The_
animal_
didn_
'_'
t_
cross_
the_
street_
because_
it_
was_
too_
tire
d_
Encoder-Decoder Attention

- In decoder, Transformer block has one more encoder-decoder attention than encoder. In encoder-decoder attention, Q is from the last output of decoder, K,V are from the output of encoder.

- Because in machine translation, decoding is a sequential operation, which means when decoding the kth feature vector, we can only know the (k-1)th and results from before. In the paper, the authors named the multi-head attention under this condition “masked multi-head attention”.
Positional Encoding
Attention Is All You Need

ENCODER #0

ENCODER #1

DECODER #0

DECODER #1

EMBEDDING WITH TIME SIGNAL

POSITIONAL ENCODING

EMBEDDINGS

INPUT

x1 = 

x2 = 

x3 = 

t1 +

t2 +

t3 +

x1

x2

x3

 INPUT

 Je

 suis

étudiant
\[ PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \]  

(3)

\[ PE(pos, 2i + 1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \]  

(4)

Pos represents the location of the word and i represents the dimension. We can find code from Google open source algorithm get_timing_signal_1d().
Result
<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>Training Cost (FLOPs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EN-DE</td>
<td>EN-FR</td>
</tr>
<tr>
<td>ByteNet [18]</td>
<td>23.75</td>
<td>39.2</td>
</tr>
<tr>
<td>Deep-Att + PosUnk [39]</td>
<td>24.6</td>
<td>39.92</td>
</tr>
<tr>
<td>GNMT + RL [38]</td>
<td>25.16</td>
<td>40.46</td>
</tr>
<tr>
<td>ConvS2S [9]</td>
<td>26.03</td>
<td>40.56</td>
</tr>
<tr>
<td>MoE [32]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deep-Att + PosUnk Ensemble [39]</td>
<td>26.30</td>
<td>41.16</td>
</tr>
<tr>
<td>GNMT + RL Ensemble [38]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ConvS2S Ensemble [9]</td>
<td>26.36</td>
<td>41.29</td>
</tr>
<tr>
<td>Transformer (base model)</td>
<td>27.3</td>
<td>38.1</td>
</tr>
<tr>
<td>Transformer (big)</td>
<td>28.4</td>
<td>41.8</td>
</tr>
<tr>
<td></td>
<td>$N$</td>
<td>$d_{\text{model}}$</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>----------------</td>
</tr>
<tr>
<td>base</td>
<td>6</td>
<td>512</td>
</tr>
<tr>
<td>(A)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>(B)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C)</td>
<td>2</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1024</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>4096</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1024</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4096</td>
</tr>
<tr>
<td>(D)</td>
<td></td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
</tr>
<tr>
<td>(E)</td>
<td></td>
<td>positional embedding instead of sinusoids</td>
</tr>
<tr>
<td>big</td>
<td>6</td>
<td>1024</td>
</tr>
</tbody>
</table>
To Sum up
Advantages:
1. Only used attention and got a very good result.
2. Not only NLP
3. Work well on GPU
Limitations:

1. Losing the ability to capture the portion features. RNN+CNN+Transformer could be better

2. Position embedding did not change the defect of structure
Thanks!
Reference:
http://jalammar.github.io/illustrated-transformer/