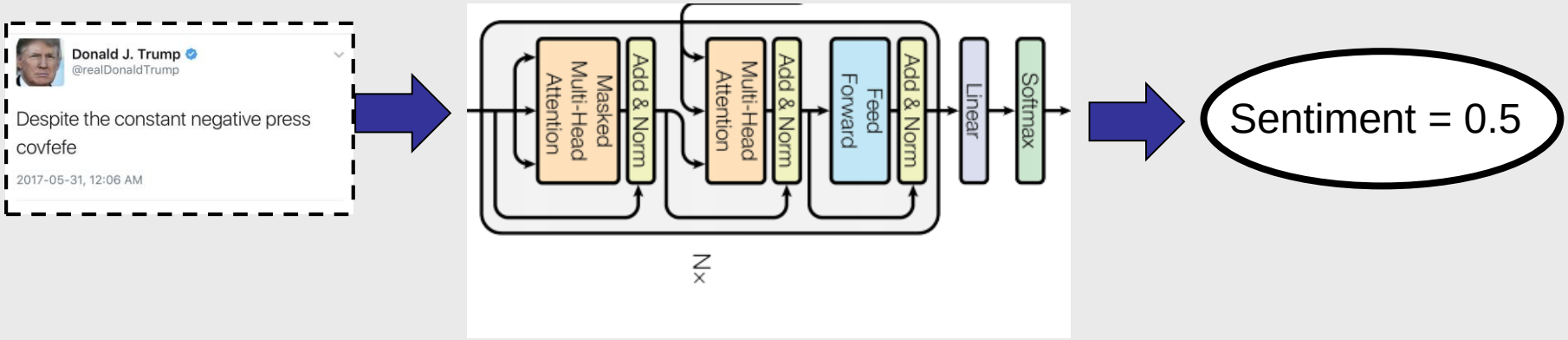
Sentiment Analysis with Neural Network Transformers



Sentiment

Sentiment

- **Tweet 1: My birthday cake was awful**

Sentiment

- **Tweet 1: My birthday cake was awful**

- **Tweet 2: My birthday cake was great**

Sentiment

- **Sentiment is conveyed by specific words**

Sentiment

- **Sentiment is conveyed by specific words**
- **Maybe we could use a word frequency approach to measure sentiment**

Sentiment

- **Sentiment is conveyed by specific words**
- **Maybe we could use a word frequency approach to measure sentiment**
- **Naïve Bayes classifier – uses this approach**

Sentiment and Context

Sentiment and Context

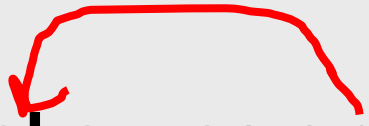
- **Tweet 1: My birthday cake was great, if you want my honest opinion**

Sentiment and Context

- **Tweet 1: My birthday cake was great, if you want my honest opinion**
- **Tweet 2: My birthday cake was great, if you want me to get diabetes**

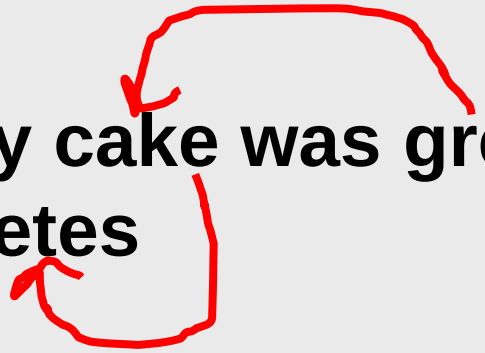
Sentiment and Context

- **Tweet 1: My birthday cake was great, if you want my honest opinion**
- **Tweet 2: My birthday cake was great, if you want me to get diabetes**



Sentiment and Context

- **Tweet 1: My birthday cake was great, if you want my honest opinion**
- **Tweet 2: My birthday cake was great, if you want me to get diabetes**



Sentiment and Context

- **Sentiment is conveyed by specific words**

Sentiment and Context

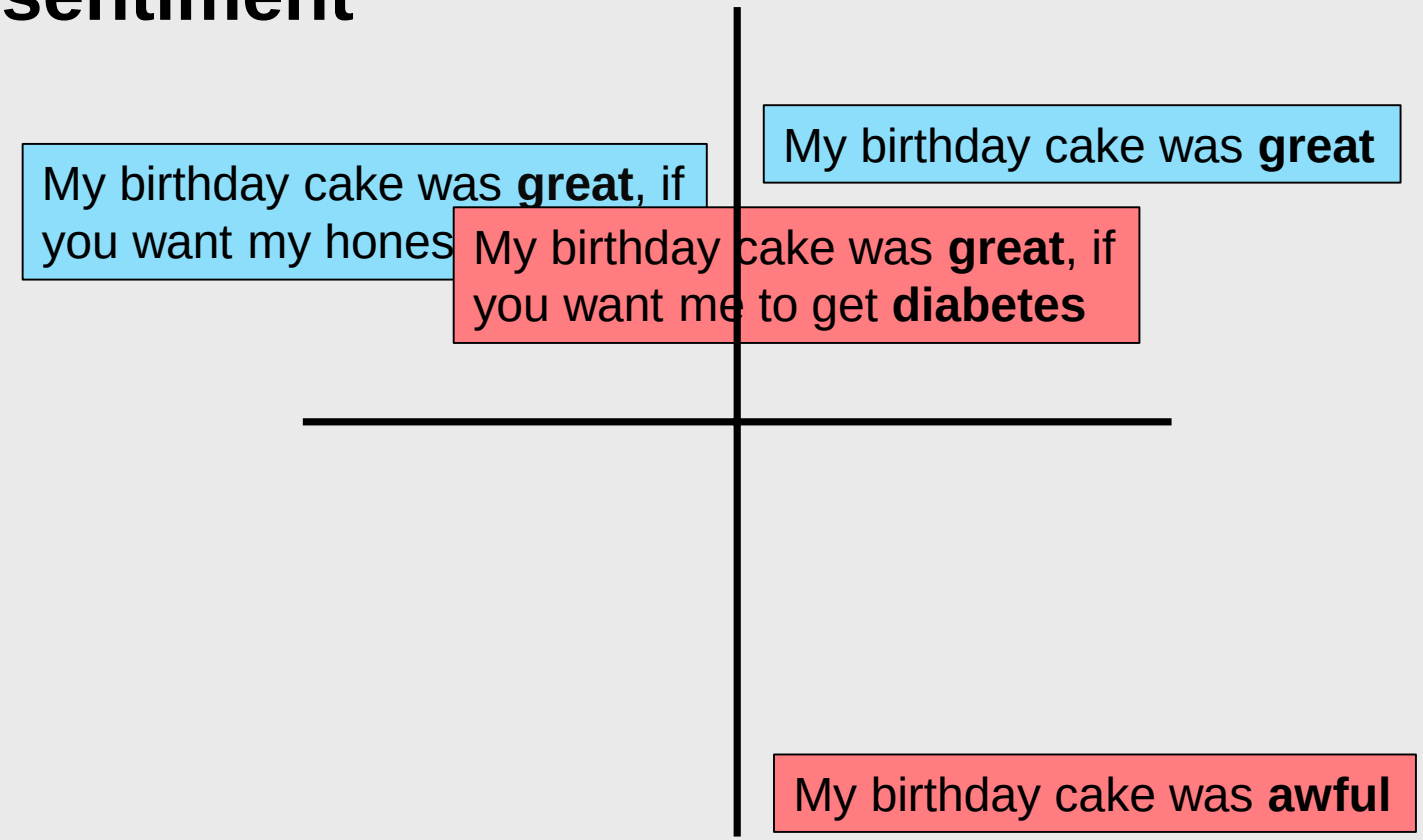
- **Sentiment is conveyed by specific words**
- **We also need to know the context of the words**

Sentiment and Context

- Sentiment is conveyed by specific words
- We also need to know the context of the words
- Context = which words pay attention to which words

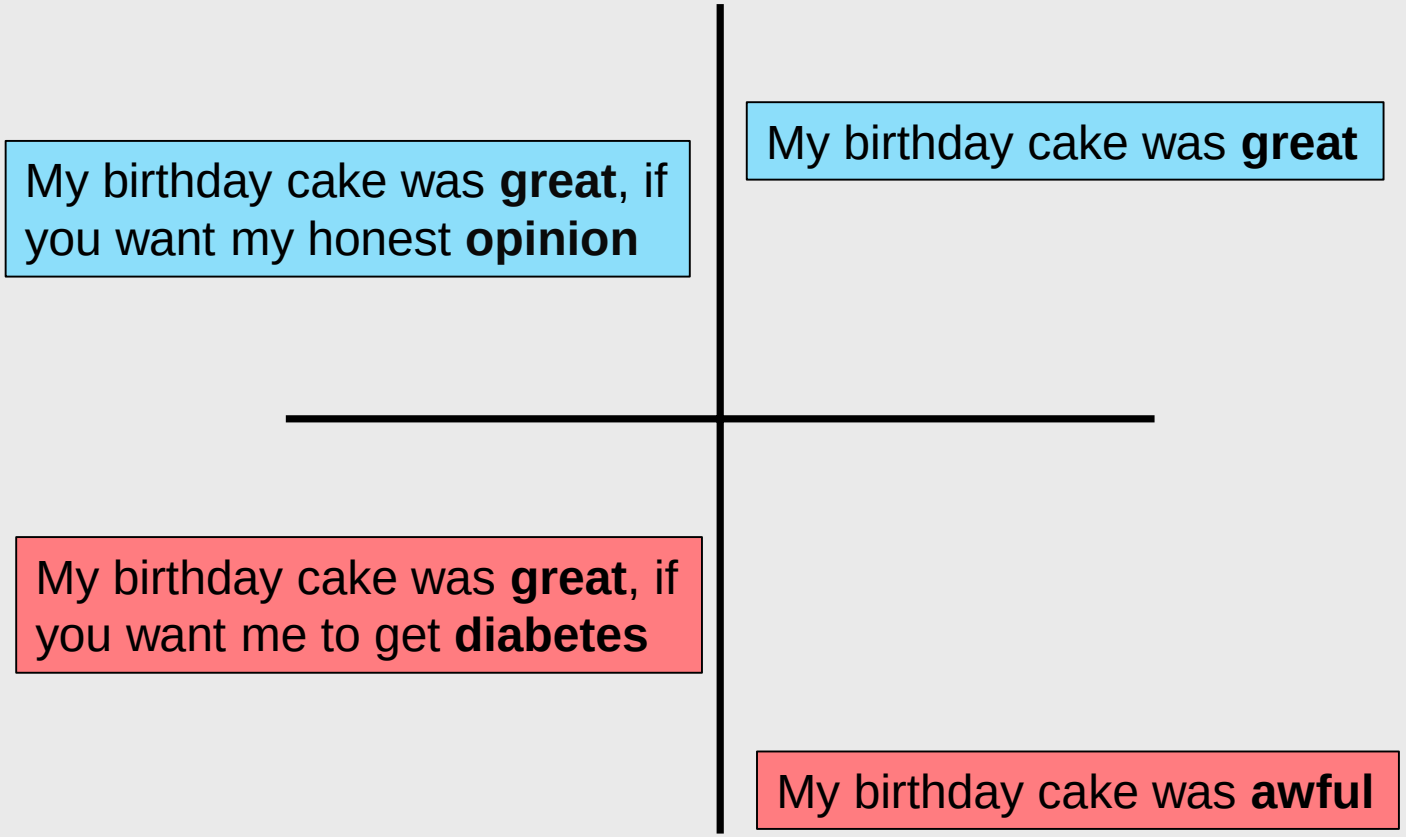
No Context Embeddings

- **A clustering type of embedding may cluster tweets with similar words, but different sentiment**



Context Dependent Embeddings

- **Context dependent embedding can cluster by sentiment**



Context Dependent Embeddings

Context Dependent Embeddings

- **We need a model that allows words in a sentence to pay “attention” to other words**

Context Dependent Embeddings

- We need a model that allows words in a sentence to pay “attention” to other words
- Words can pay attention in different ways

Context Dependent Embeddings

- We need a model that allows words in a sentence to pay “attention” to other words
- Words can pay attention in different ways
- We can choose the type of “attention” that captures sentiment

Context Dependent Embeddings

- We need a model that allows words in a sentence to pay “attention” to other words
- Words can pay attention in different ways
- We can choose the type of “attention” that captures sentiment
- Solution: Neural Network Transformers

Transformers

- **Developed in 2017 by Google**
- **Revolutionized natural language processing**

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

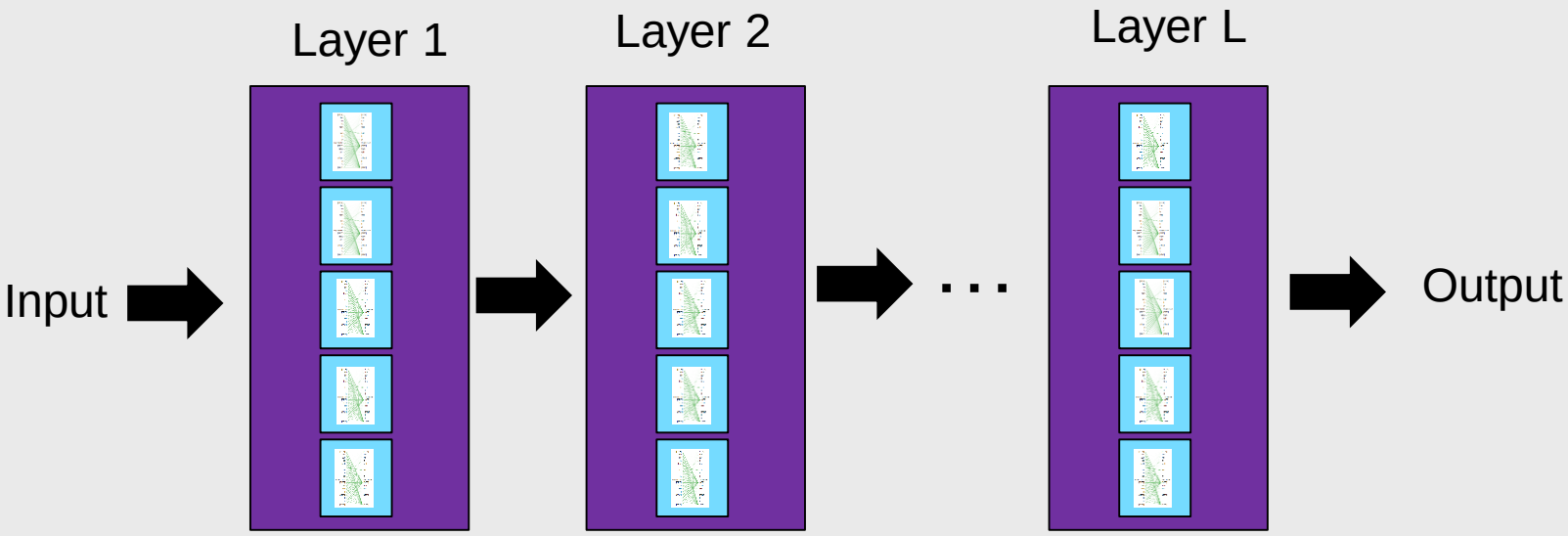
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention

What Can Transformers Do?

- Measure sentiment
- Translation
- Web search
- Text summarization
- Question answering
- Generate text

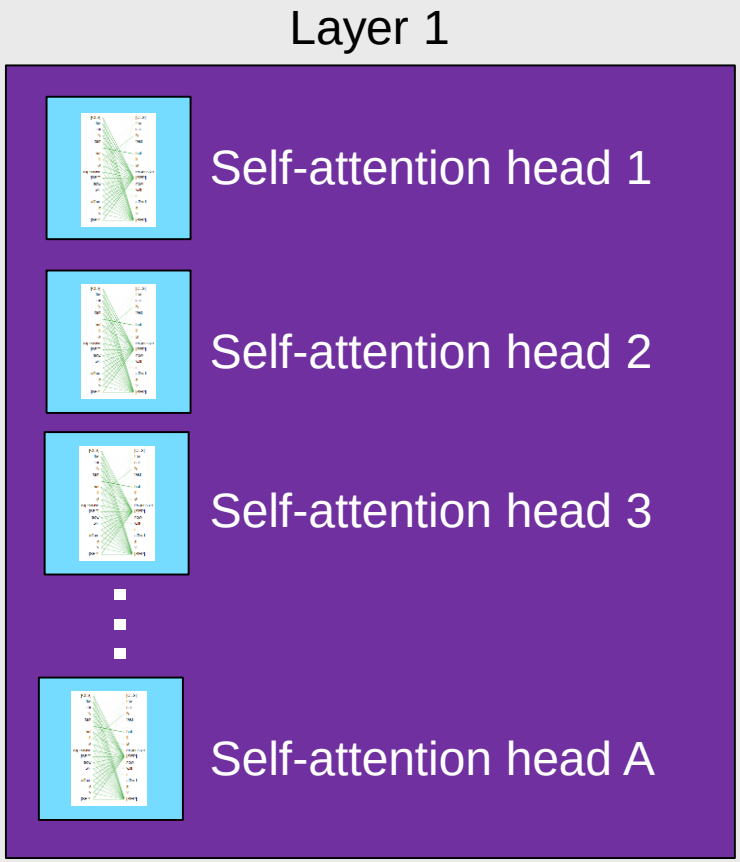
Transformer Architecture

- **The transformer has many layers**



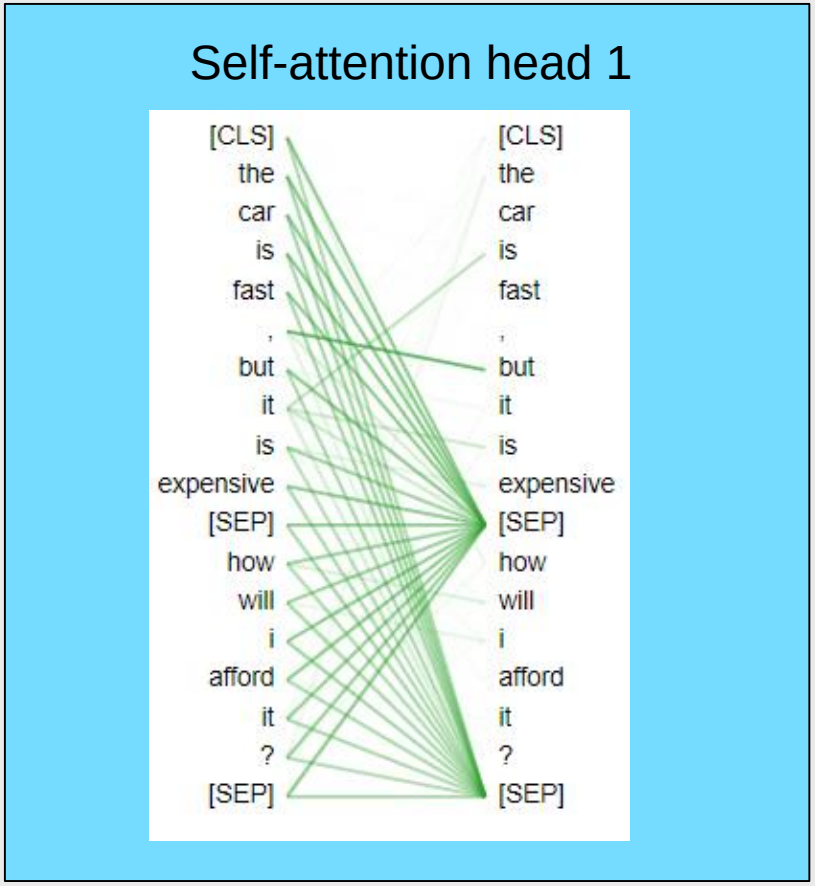
Transformer Layers

- **Each layer has many self-attention heads**

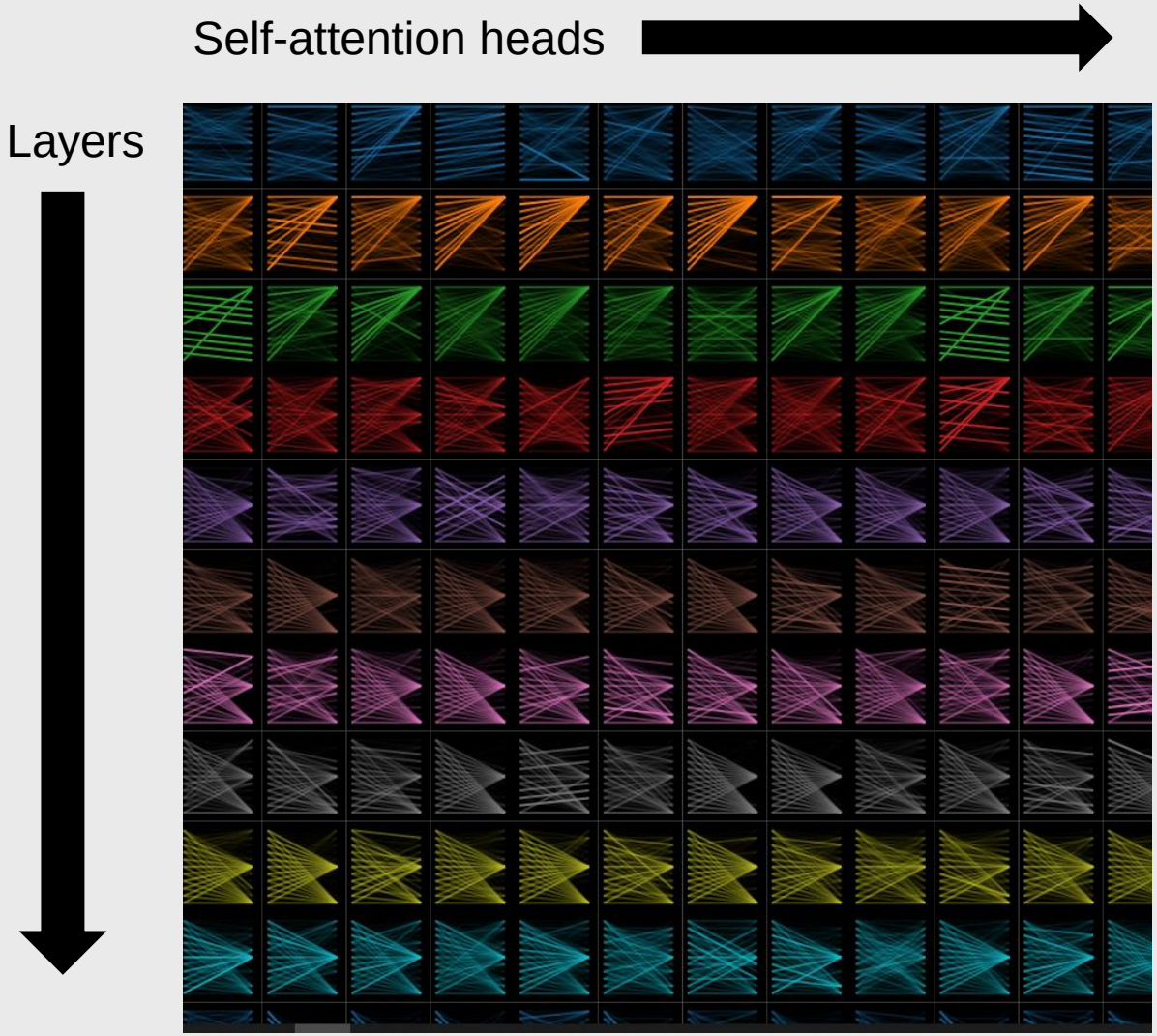


Self-Attention Head

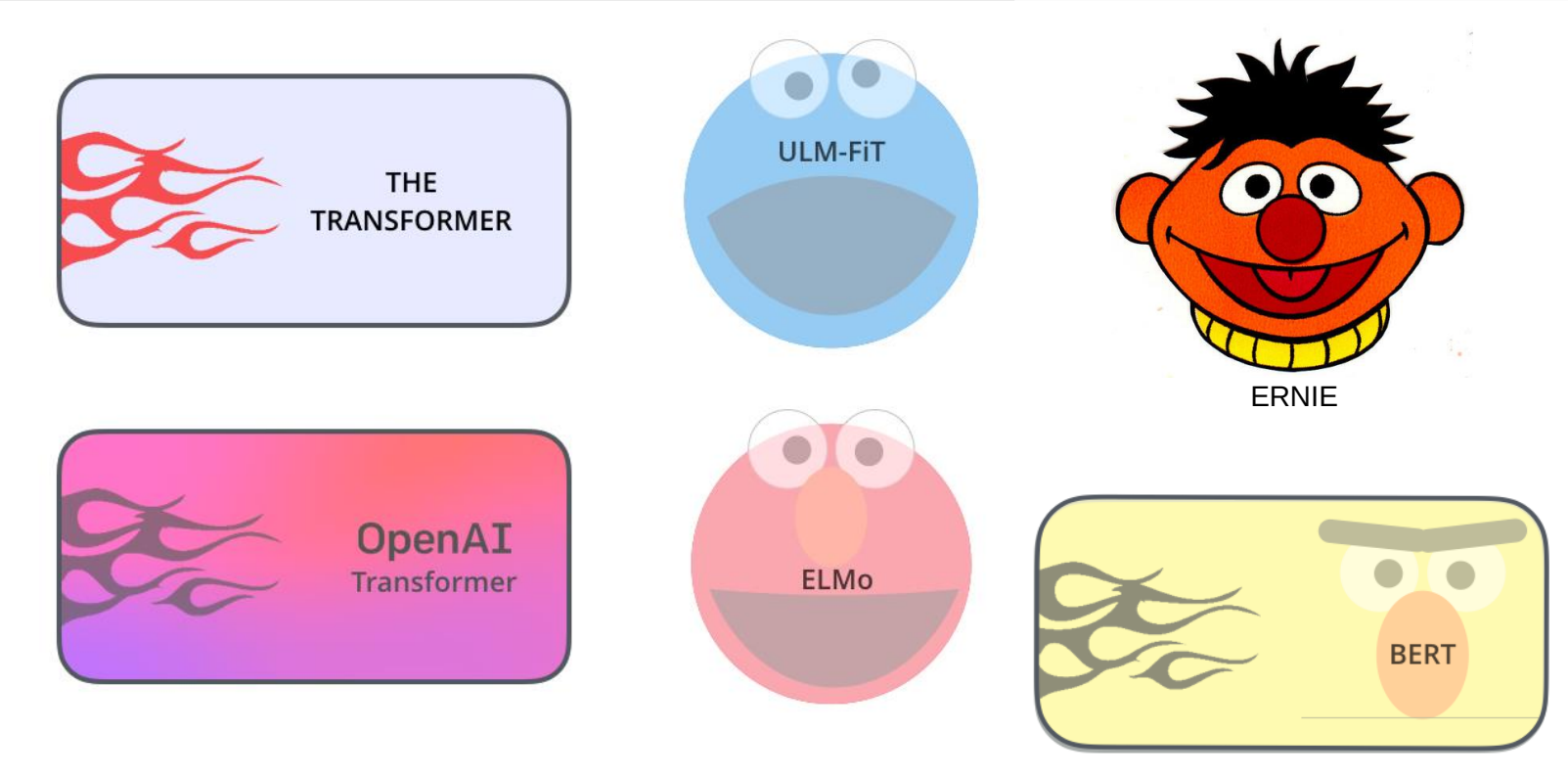
- **Each self-attention head contains attention weights from each word to each other word**



Visualizing the Brain of a Transformer

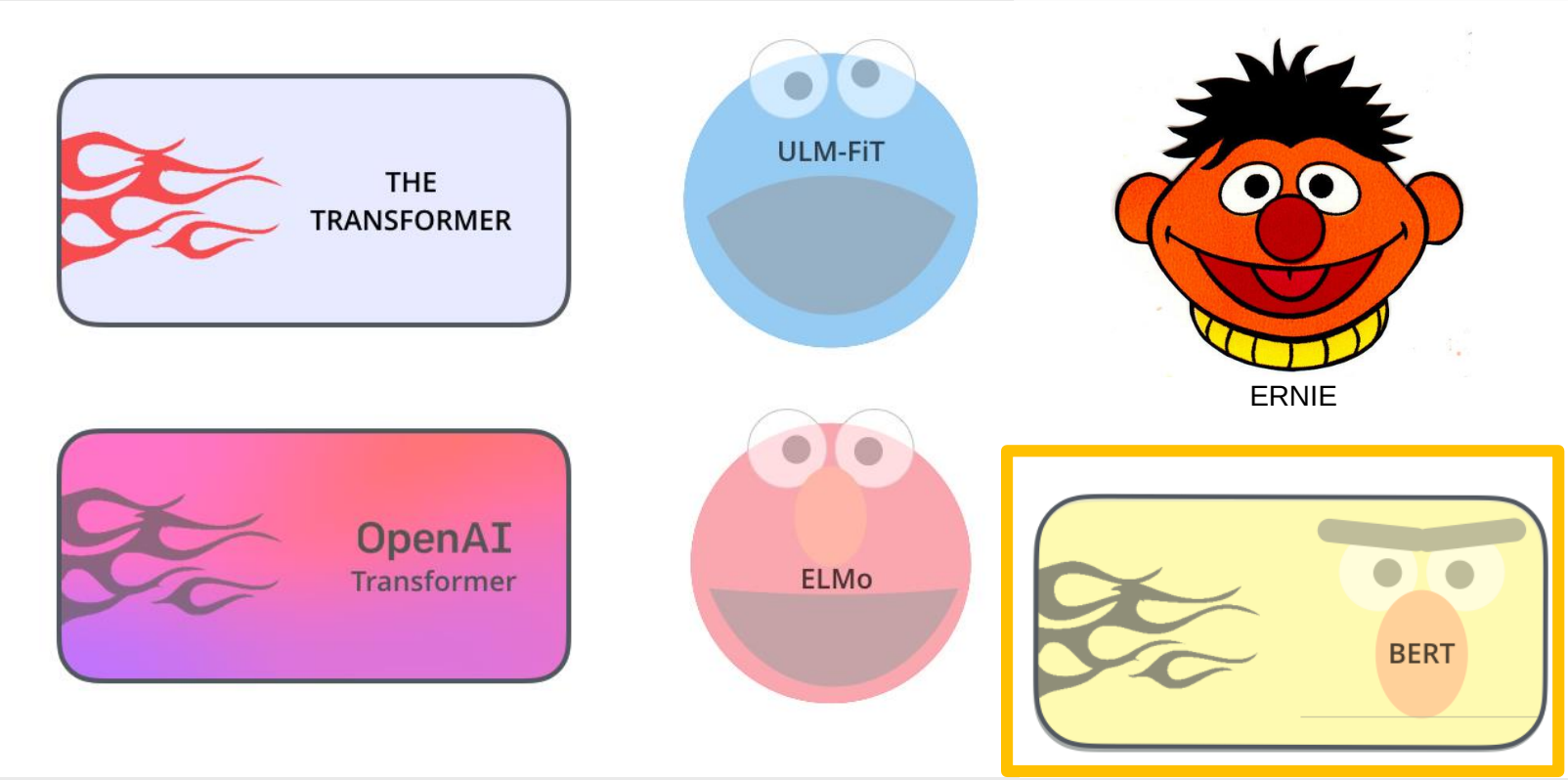


Popular Transformers



http://jalammar.github.io/illustrated-bert/

Popular Transformers

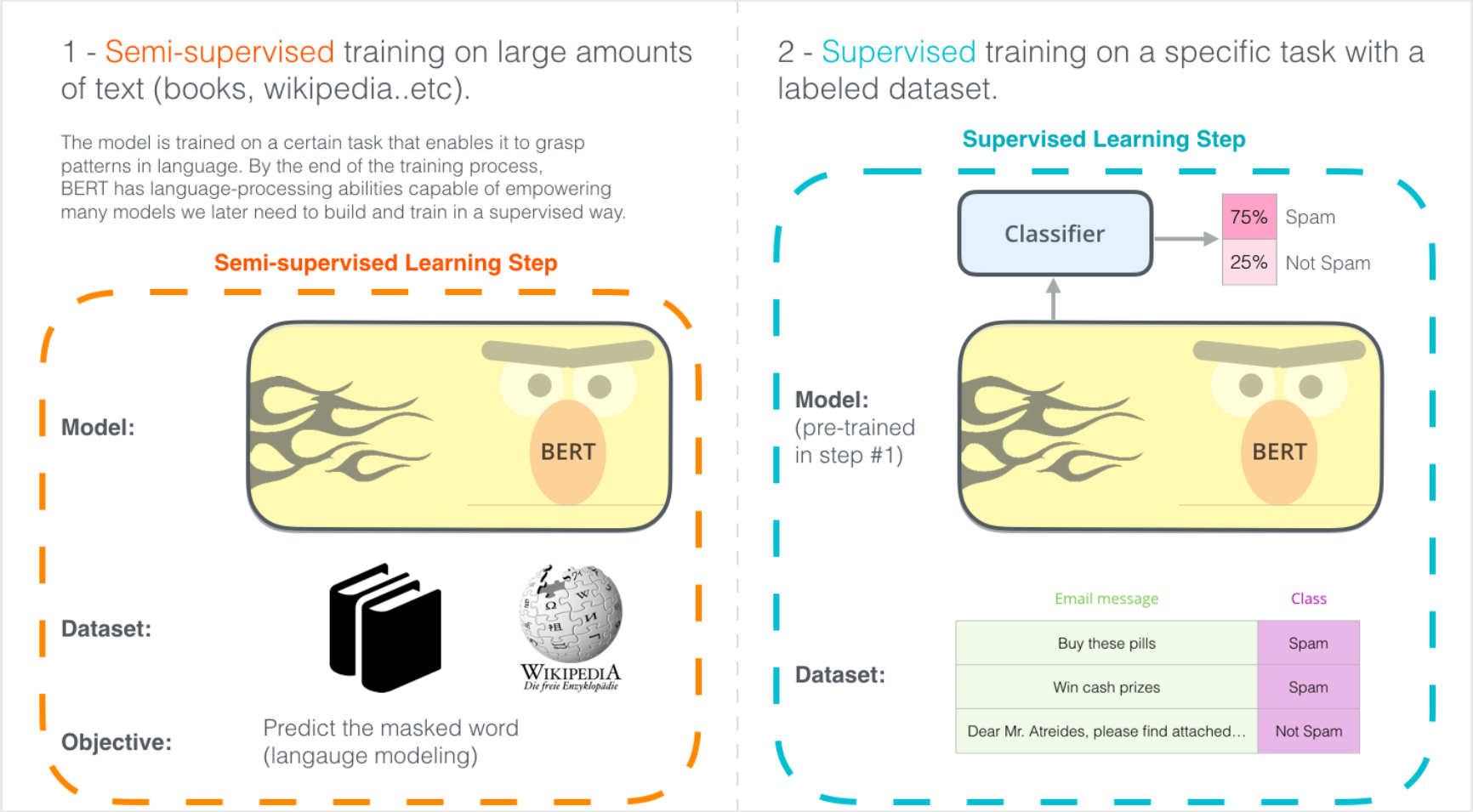


http://jalammar.github.io/illustrated-bert/

BERT

- **BERT = Bi-directional Encoder Representations From Transformers**
- **Released in 2018 by Google**
- **Base BERT has 100 million parameters**
  - **12 layers**
  - **12 attention heads per layer**
  - **768 dimensional word embedding**
- **Trained on books and Wikipedia (3.3 billion words)**

Training BERT



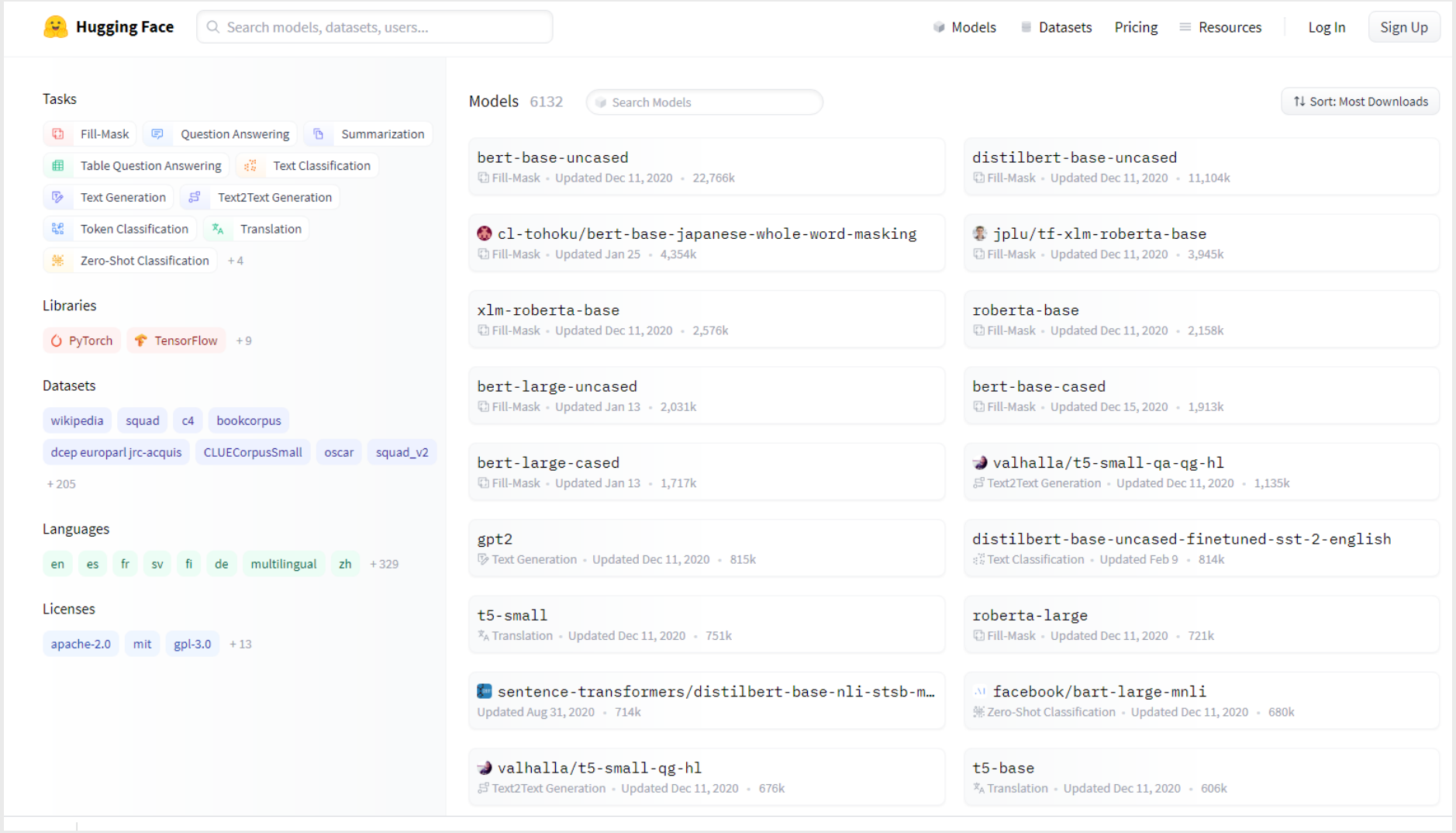
http://jalammar.github.io/illustrated-bert/

Masked Language Model Task

- **BERT is trained to learn a masked language model**
  - **Guess [MASK] words in a sentence**

Data	Prediction
I went to the [MASK] to buy milk.	[MASK] = store
I graduated from [MASK] and got a degree.	[MASK] = college
I had a [MASK] and it tasted [MASK]!	[MASK] = hamburger [MASK] = amazing

Pre-Trained Transformers: Hugging Face



Evaluating Language Models: GLUE

- **GLUE = general language understanding and evaluation**
- **GLUE is a set of benchmark tasks to evaluate language models like BERT**

GLUE Tasks

Task type	Description
Acceptability	Is the sentence grammatically correct
Sentiment	Can you predict the sentiment of the sentence
Question answering	Does the second sentence answer the question in the first sentence
Natural language inference	Does the second sentence entail the hypothesis in the first sentence
Pronoun referral	To what does the pronoun in a sentence refer
Sentence similarity	Are the two sentences paraphrases of each other

GLUE Leaders - 2021

- **Human GLUE score = 87.1**
- **GLUE leaderboard: https://gluebenchmark.com/leaderboard**

Rank	Name	Model	URL	Score
1	ERNIE Team - Baidu	ERNIE		91.1
2	AliceMind & DIRL	StructBERT + CLEVER		91.0
3	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.8
4	HFL iFLYTEK	MacALBERT + DKM		90.7
5	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6
6	Liangzhu Ge	Deberta + CLEVER		90.5
7	T5 Team - Google	T5		90.3
8	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART		89.9
9	Huawei Noah's Ark Lab	NEZHA-Large		89.8
10	Zihang Dai	Funnel-Transformer (Ensemble B10-10-10H1024)		89.7

SuperGLUE Leaders 2021

- **Human SuperGLUE score = 89.8**
- **SuperGLUE leaderboard: https://super.gluebenchmark.com/leaderboard**

	Rank	Name	Model	URL	Score
	1	ERNIE Team - Baidu	ERNIE 3.0		90.6
+	2	Zirui Wang	T5 + UDG, Single Model (Google Brain)		90.4
+	3	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.3
	4	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8
+	5	T5 Team - Google	T5		89.3
+	6	Huawei Noah's Ark Lab	NEZHA-Plus		86.7
+	7	Alibaba PAI&ICBU	PAI Albert		86.1
+	8	Infosys : DAWN : AI Research	RoBERTa-iCETS		86.0
+	9	Tencent Jarvis Lab	RoBERTa (ensemble)		85.9
	10	Zhuiyi Technology	RoBERTa-mtl-adv		85.7

SuperGLUE Leaders 2021

- **Human SuperGLUE score = 89.8**
- **SuperGLUE leaderboard: https://super.gluebenchmark.com/leaderboard**

	Rank	Name	Model	URL	Score
	1	ERNIE Team - Baidu	ERNIE 3.0		90.6
+	2	Zirui Wang	T5 + UDG, Single Model (Google Brain)		90.4
+	3	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.3
	4	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8
+	5	T5 Team - Google	T5		89.3
+	6	Huawei Noah's Ark Lab	NEZHA-Plus		86.7
+	7	Alibaba PAI&ICBU	PAI Albert		86.1
+	8	Infosys : DAWN : AI Research	RoBERTa-iCETS		86.0
+	9	Tencent Jarvis Lab	RoBERTa (ensemble)		85.9
	10	Zhuiyi Technology	RoBERTa-mtl-adv		85.7

SuperGLUE Leaders 2021

- **Human SuperGLUE score = 89.8**
- **SuperGLUE leaderboard: https://super.gluebenchmark.com/leaderboard**

	Rank	Name	Model	URL	Score
	1	ERNIE Team - Baidu	ERNIE 3.0		90.6
+	2	Zirui Wang	T5 + UDG, Single Model (Google Brain)		90.4
+	3	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.3
	4	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8
+	5	T5 Team - Google	T5		89.3
+	6	Huawei Noah's Ark Lab	NEZHA-Plus		86.7
+	7	Alibaba PAI&ICBU	PAI Albert		86.1
+	8	Infosys : DAWN : AI Research	RoBERTa-iCETS		86.0
+	9	Tencent Jarvis Lab	RoBERTa (ensemble)		85.9
	10	Zhuiyi Technology	RoBERTa-mtl-adv		85.7

Coding Session

- **Learn how to use any model in the huggingface library**
- **Use a pre-trained sentiment classifier to measure sentiment of tweets**
- **Perform analysis of the tweets and their sentiment**
- **Code located at http://github.com/zlisto/sentiment_analysis**