Natural Language Processing

Training Head: Koby

31-10-23

Feedback & Attendance

Feedback/Queries (Anonymous)

- https://forms.gle/GWoC4of5uyz58kVT9

Attendance

- https://forms.gle/7YJivGwY1PaDoZeB9

Attendance code: 25

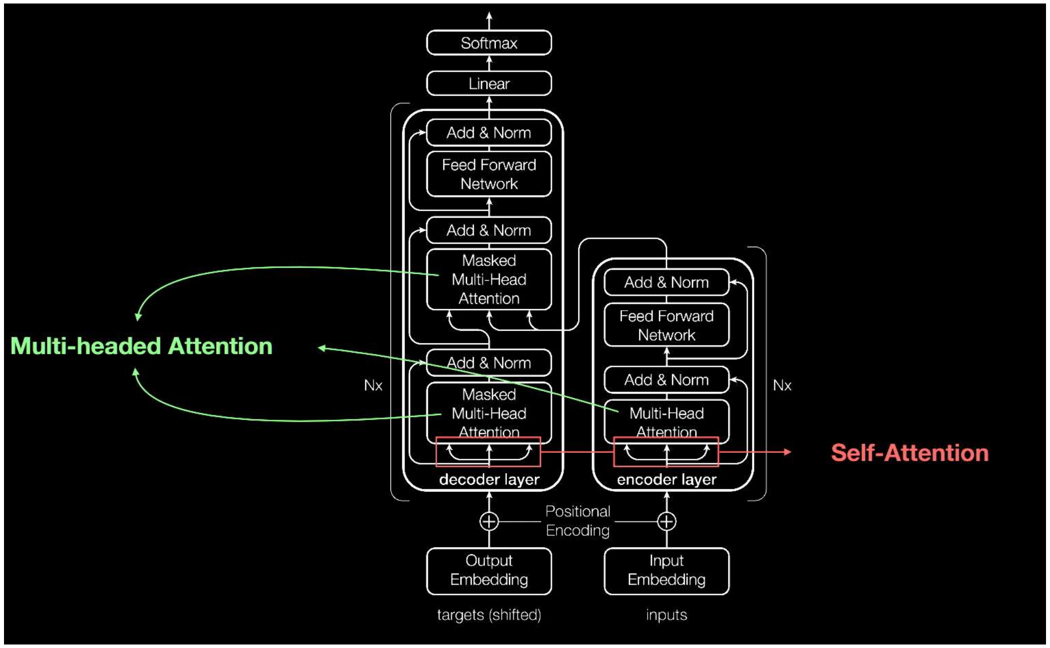
Table of Contents

- NLP Tasks
- Preprocessing
- Tokenization
- Lemmatization and Stemming
- Vectorization
- Transformers
- BERT
- ChatGPT

Now, for something scarier than Halloween…

- Language models in AI are handing complexity, ambiguity, context a lot better.
- Could this turn into machine consciousness?
- Statistical feature functions allowed for the understanding of more complex and long-distance linguistic relations.
- Word-embeddings such as word-2-vec were better at synonyms.
- Bi-directional word-embeddings such as Elmo and BERT were better for homonyms.
- Encoding-decoding mechanisms dramatically increased the quality Q&A (ChatBots) and translation allowing the use of machine learning.
- **Attention** addressed alignment problems in translation.

Transformer model



- Published in 2017
- A full encoder-decoder model with multiple layers of self-attention.
- Very basic compared to BERT and GPT, but it was a giant leap as it was a model that handled almost all linguistic challenges in NLP
- Softmax layer: good for multi-class classification

Generative AI models attempt to tell jokes:

- https://ipro.com/resources/articles/why-understanding-human-language-is-hard-for-computers-and-how-modern-ai-is-changing-the-game/#:~:text=Human%20language%20is%20difficult%20for,situation%2C%20and%20the%20cultural%20context.
- NLP in generative AI: very cringe jokes almost 10 years ago e.g. "One morning I shot an elephant in my pajamas. How he got in my pajamas, I don't know."
- Compare this with ChatGPT being able to give (somewhat) funny jokes e.g. "Why did the scarecrow win an award? Because he was outstanding in his field"

NLP Challenges

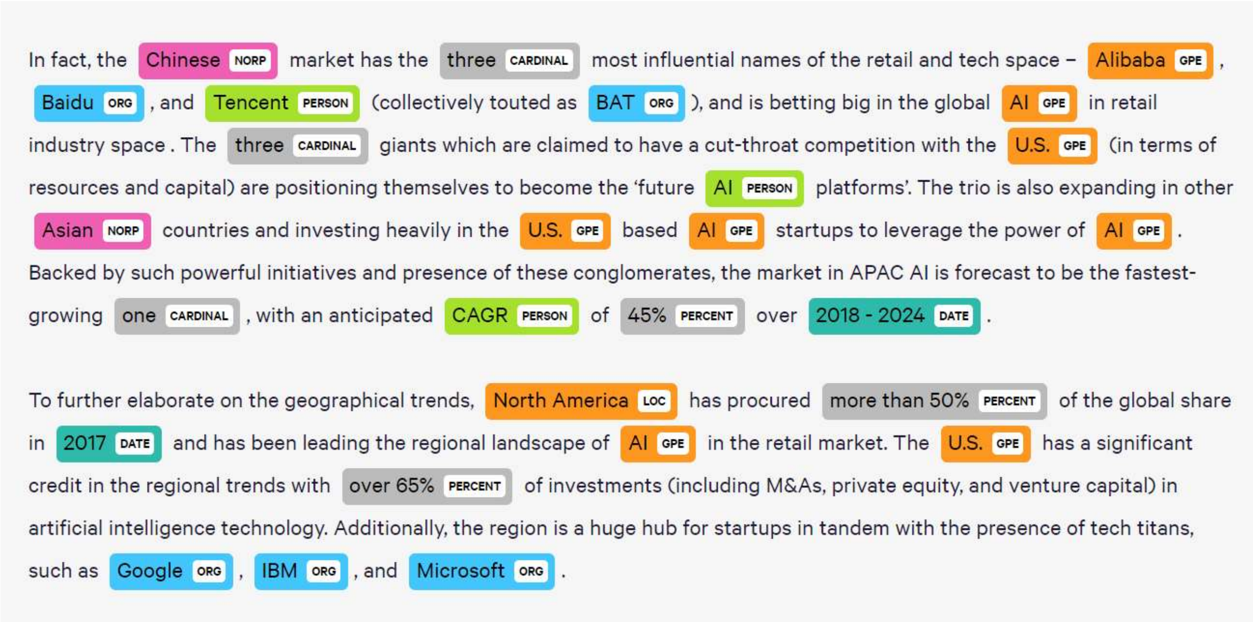
- Words vs machines
- Previously output grammar was more an issue than input grammar
- Ambiguity of user’s input context

NLP Tasks

- Machine translation
- Sentiment Analysis
- Spam detector
- Named Entity Recognition (NER)
- Topic identification
- Similarity
- Chatbots
- Speech recognition
- Next word recommendation
- Text to image generation

Named Entity Recognition (NER)

- Information extraction
- Pre-defined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc



In fact, the Chinese NORP market has the three CARDINAL most influential names of the retail and tech space – Alibaba GPE , Baidu ORG , and Tencent PERSON (collectively touted as BAT ORG), and is betting big in the global AI GPE in retail industry space . The three CARDINAL giants which are claimed to have a cut-throat competition with the U.S. GPE (in terms of resources and capital) are positioning themselves to become the 'future AI PERSON platforms'. The trio is also expanding in other Asian NORP countries and investing heavily in the U.S. GPE based AI GPE startups to leverage the power of AI GPE . Backed by such powerful initiatives and presence of these conglomerates, the market in APAC AI is forecast to be the fastest-growing one CARDINAL , with an anticipated CAGR PERSON of 45% PERCENT over 2018 - 2024 DATE .

To further elaborate on the geographical trends, North America LOC has procured more than 50% PERCENT of the global share in 2017 DATE and has been leading the regional landscape of AI GPE in the retail market. The U.S. GPE has a significant credit in the regional trends with over 65% PERCENT of investments (including M&As, private equity, and venture capital) in artificial intelligence technology. Additionally, the region is a huge hub for startups in tandem with the presence of tech titans, such as Google ORG , IBM ORG , and Microsoft ORG .

Preprocessing

Basics

- Lower casing
- Remove URL
- Remove punctuations
- Remove stopwords

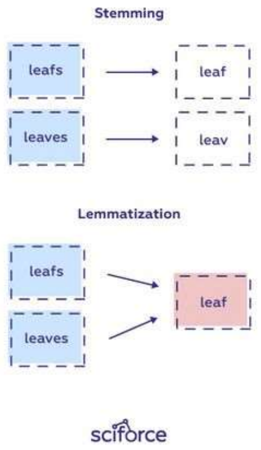
https://www.kaggle.com/code/redwankarimsony/nlp-101-tweet-sentiment-analysis-preprocessing

Stopwords

- Remove commonly used word (such as “the”, “a”, “an”, “in”)
- No meaningful information, especially if we are building a text classification model

Stemming and Lemmatization

- Stemming
  - Removes last few characters from a word
- Lemmatization
  - Considers context and convert to base form (lemma)
  - Computationally more expensive

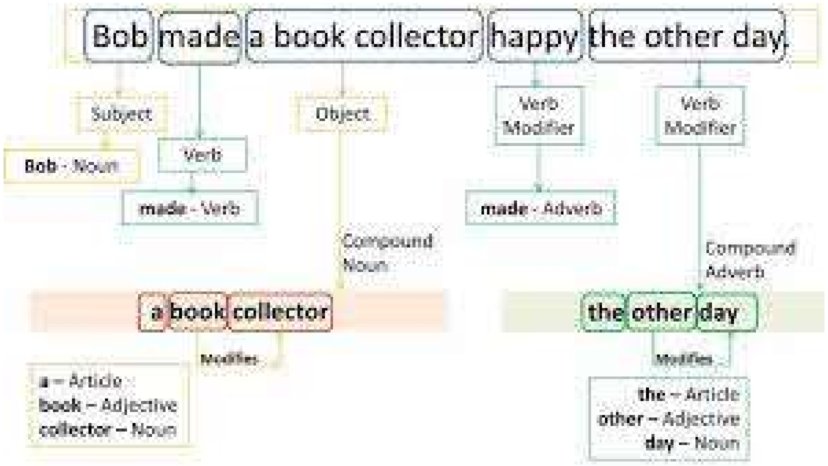


Chunking (aka partial parsing)

- Extracting phrases from unstructured text (IE)
    - extract the important words alone from lengthy descriptions
- Group the required phrases and exclude all the modifiers around them which are not necessary for our analysis
- Very important when you want to extract information from text (eg NER)

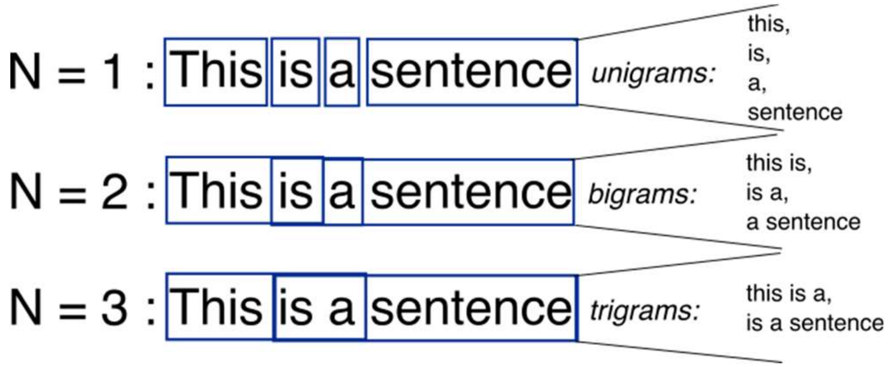
Part-Of-Speech tagging

- Converting a sentence to forms – list of words, list of tuples (where each tuple is having a form (word, tag))
- Parse trees used for NER



N-grams

- Finds probability distribution over a sequence of words
- Applications:
  - Speech recognition to remove noise
  - Bag-of-words
- Out-of-vocab words



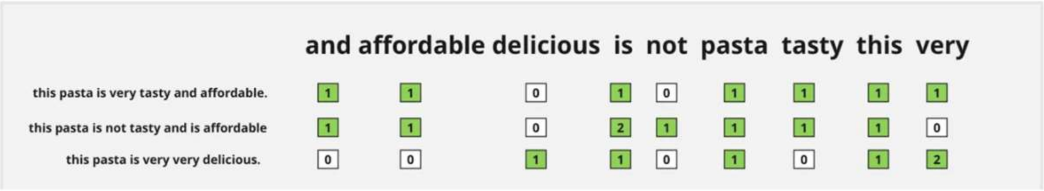
Tokenization

- Separating a piece of text into smaller units called tokens
  - Words, characters, subwords (n-gram characters)
- Pre-trained word embedding
  - Word2Vec and GloVe
  - Out Of Vocabulary (OOV) words
- Word-piece tokenization
  - ‘sleeping’ word is tokenized into ‘sleep’ and ‘##ing’
  - 2 special tokens ‘[CLS]’ and ‘[SEP]’
    - BERT model is designed in such a way that the sentence has to start with the [CLS] token and end with the [SEP] token

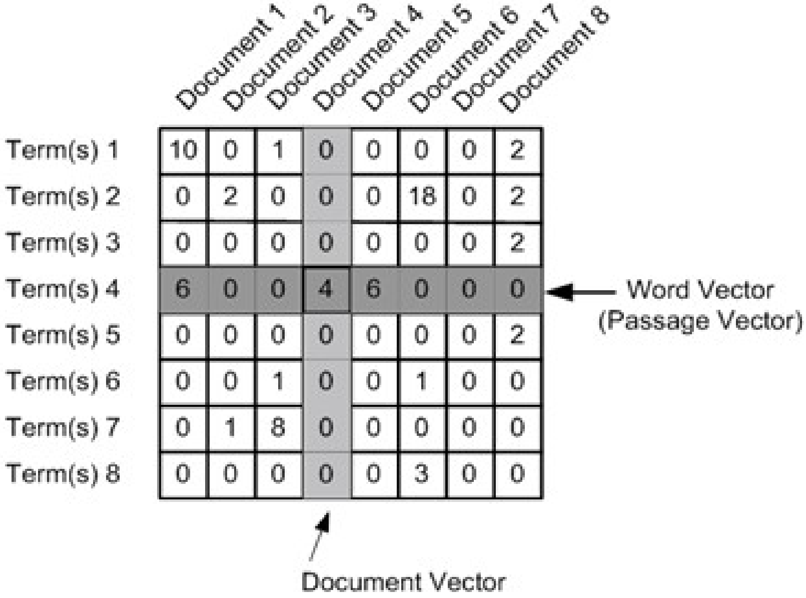
Vectorization

Bag-Of-Words (BoW)/Count Vectorizer

- Disregards grammar and word order
- Final BoW representation is the sum of the words feature vector



	and	affordable	delicious	is	not	pasta	tasty	this	very
this pasta is very tasty and affordable.	1	1	0	1	0	1	1	1	1
this pasta is not tasty and is affordable	1	1	0	2	1	1	1	1	0
this pasta is very very delicious.	0	0	1	1	0	1	0	1	2



	Document 1	Document 2	Document 3	Document 4	Document 5	Document 6	Document 7	Document 8
Term(s) 1	10	0	1	0	0	0	0	2
Term(s) 2	0	2	0	0	0	18	0	2
Term(s) 3	0	0	0	0	0	0	0	2
Term(s) 4	6	0	0	4	6	0	0	0
Term(s) 5	0	0	0	0	0	0	0	2
Term(s) 6	0	0	1	0	0	1	0	0
Term(s) 7	0	1	8	0	0	0	0	0
Term(s) 8	0	0	0	0	0	3	0	0

TF-IDF

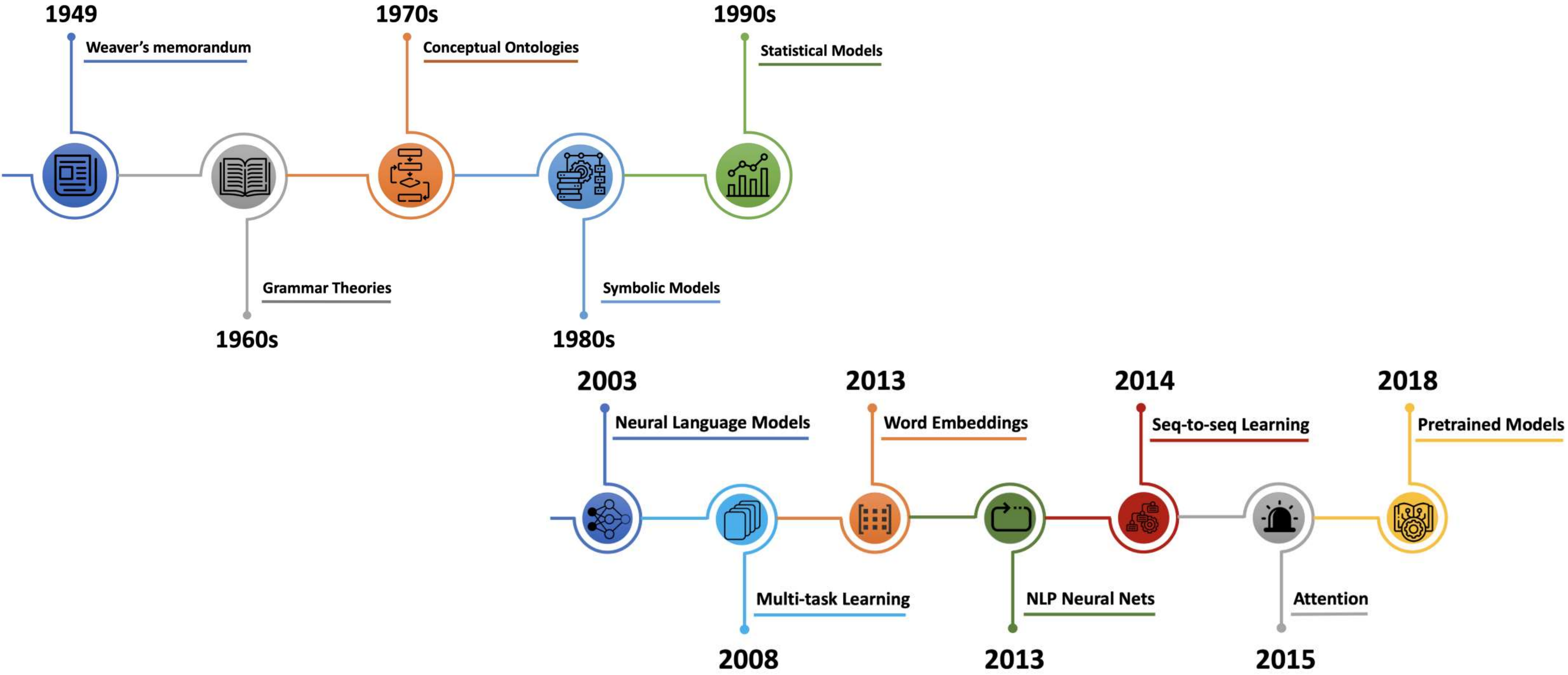
$$IDF(\boldsymbol{term}) = \log\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ \boldsymbol{term}\ in\ it}\right)$$

$$TFIDF(\boldsymbol{term}) = TF(\boldsymbol{term}) * \mathrm{IDF}(\mathbf{term})$$

- Indicates how often the token is found in the document and how unique the token is to the whole entire corpus of documents.
- Gives more weight to less frequently occurring events and less weight to expected events
  - Penalizes frequently occurring words that appear frequently in a document such as “the”, “is” but assigns greater weight to less frequent or rare words.

Language Models

History of Language Models



1949 Weaver's memorandum

1960s Grammar Theories

1970s Conceptual Ontologies

1980s Symbolic Models

1990s Statistical Models

2003 Neural Language Models

2008 Multi-task Learning

2013 Word Embeddings

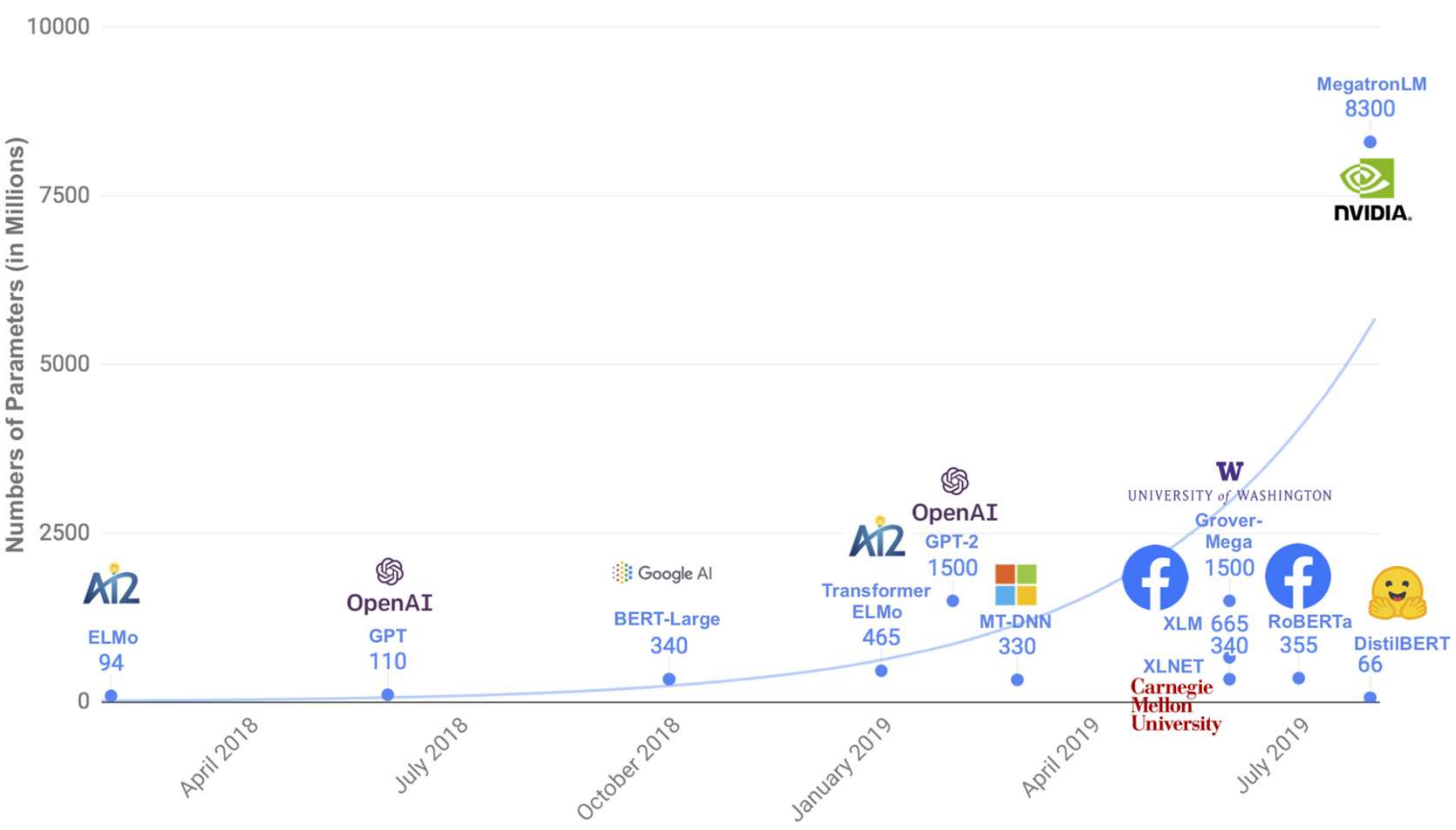
2013 NLP Neural Nets

2014 Seq-to-seq Learning

2015 Attention

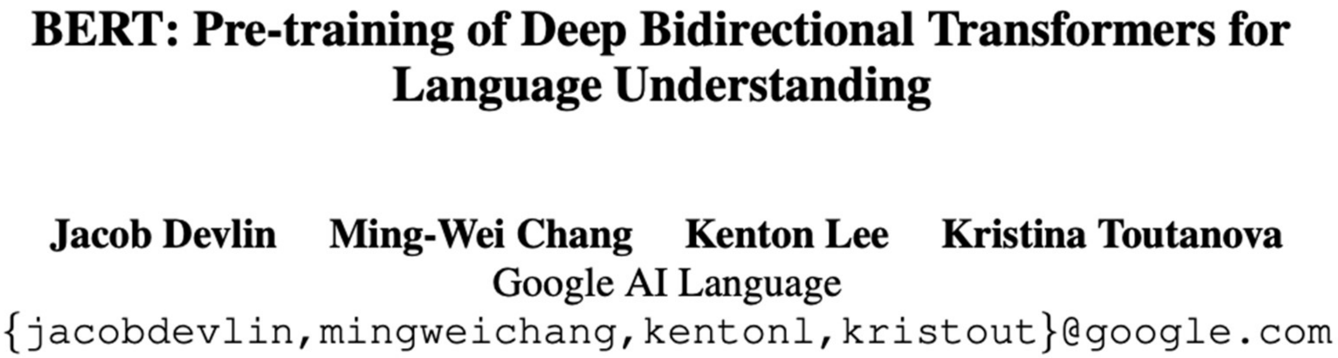
2018 Pretrained Models

Parameter counts of Pre-trained Language Models



Bidirectional Encoder Representation from Transformers (BERT)

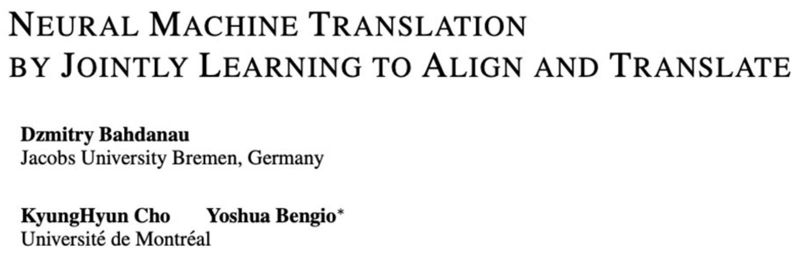
“the game will lead to a **tie** if both the guys **tie** their final **tie** at the same time.”



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`

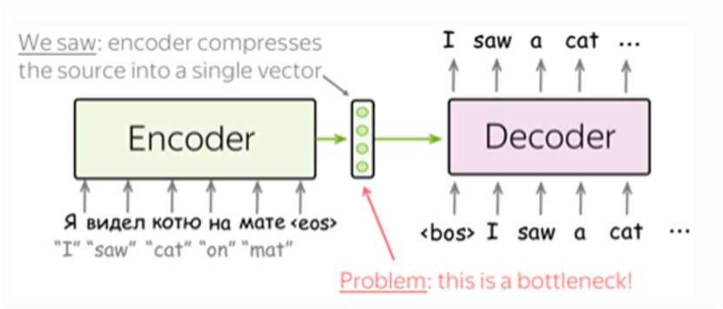
Attention

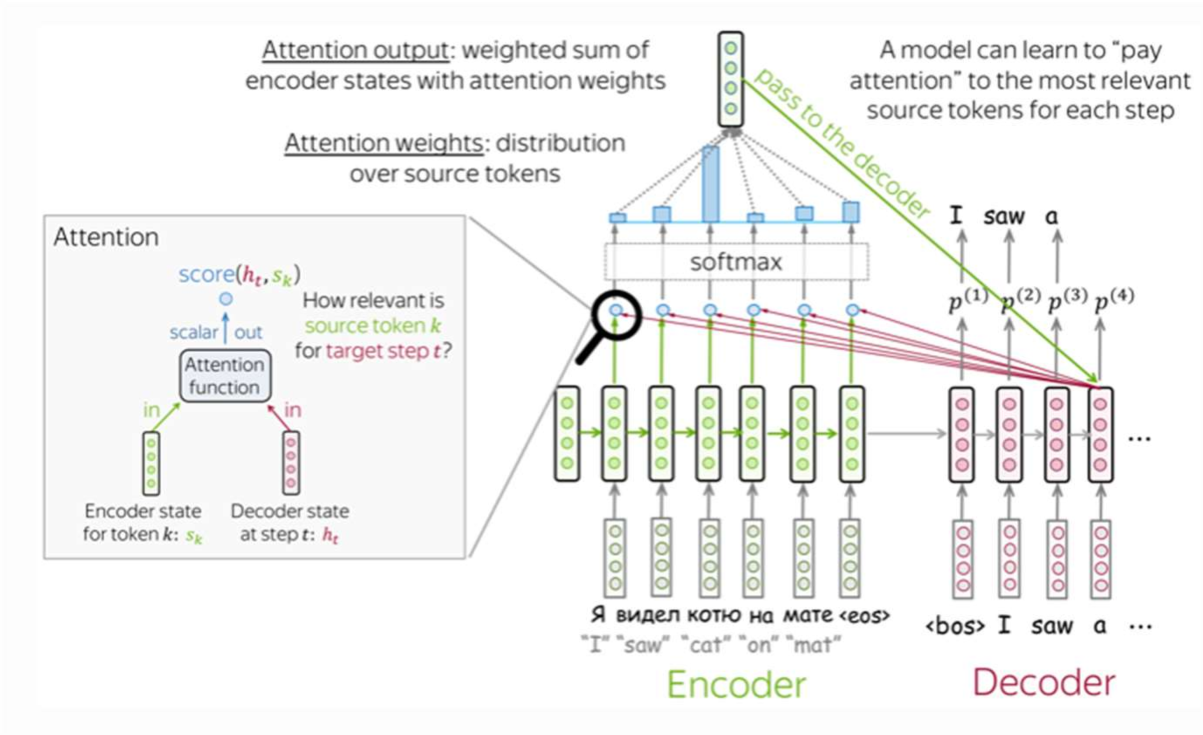


NEURAL MACHINE TRANSLATION
BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

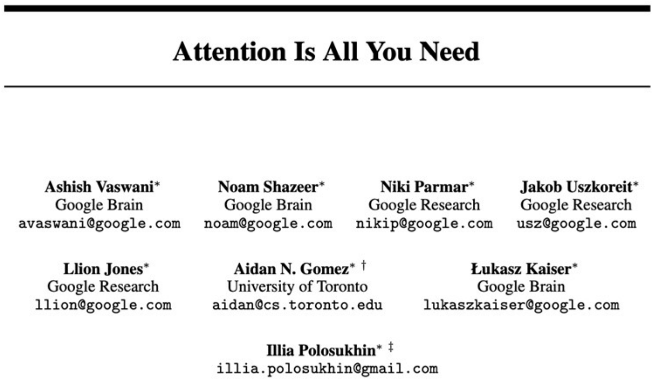
Dzmitry Bahdanau
Jacobs University Bremen, Germany

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


We saw: encoder compresses the source into a single vector
Encoder
Decoder
I saw a cat …
Я видел котю на мате <eos>
"I" "saw" "cat" "on" "mat"
<bos> I saw a cat …
Problem: this is a bottleneck!


Attention output: weighted sum of encoder states with attention weights
Attention weights: distribution over source tokens
A model can learn to "pay attention" to the most relevant source tokens for each step
pass to the decoder
softmax
I saw a
$p^{(1)}$ $p^{(2)}$ $p^{(3)}$ $p^{(4)}$
Attention
score(h_t, s_k)
scalar
out
How relevant is source token k for target step t?
Attention function
in
in
Encoder state for token k: s_k
Decoder state at step t: h_t
Я видел котю на мате <eos>
"I" "saw" "cat" "on" "mat"
<bos> I saw a …
Encoder
Decoder

Transformers



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

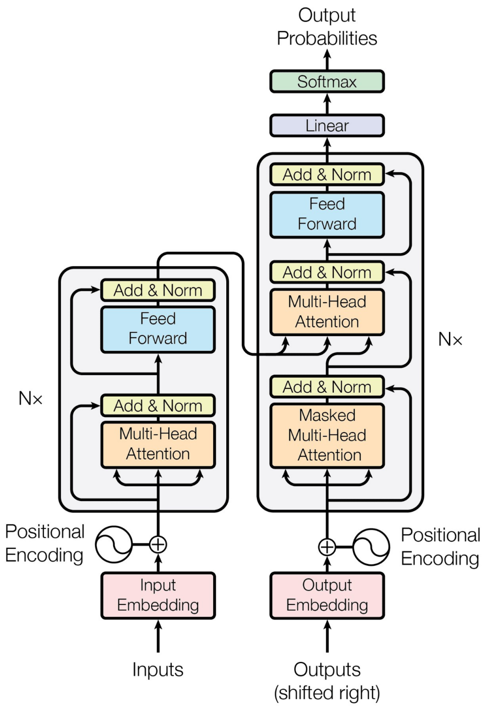
Llion Jones*
Google Research
llion@google.com

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

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

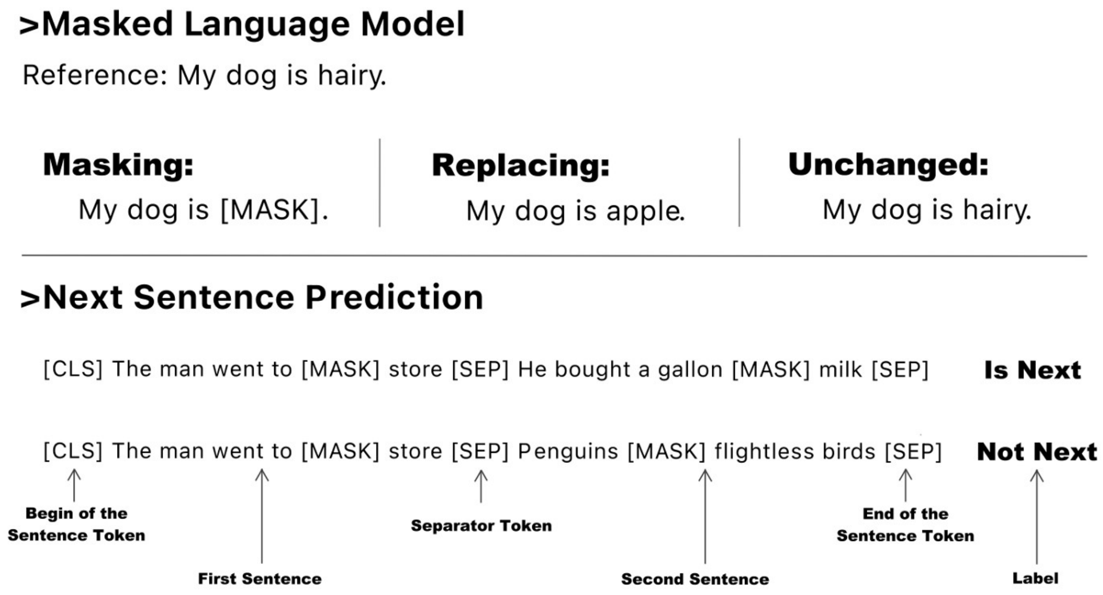
Illia Polosukhin* ‡
illia.polosukhin@gmail.com

	Seq2seq without attention	Seq2seq with attention	Transformer
processing within encoder	RNN/CNN	RNN/CNN	attention
processing within decoder	RNN/CNN	RNN/CNN	attention
decoder-encoder interaction	static fixed-sized vector	attention	attention



BERT (1/2)

- Pre-training tasks
  1. Masked Language Modeling (Bi-directionality)
  2. Next sentence prediction



>Masked Language Model

Reference: My dog is hairy.

Masking: My dog is [MASK].

Replacing: My dog is apple.

Unchanged: My dog is hairy.

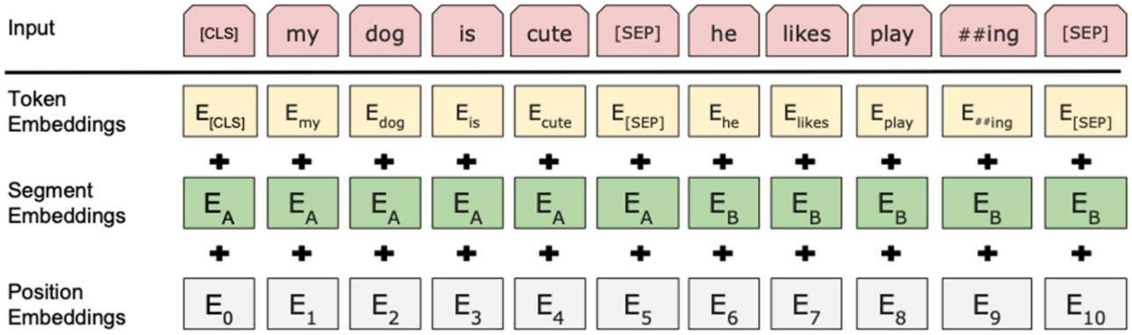
>Next Sentence Prediction

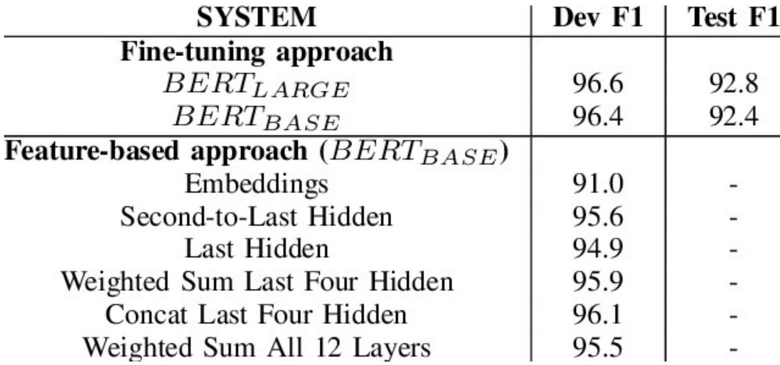
[CLS] The man went to [MASK] store [SEP] He bought a gallon [MASK] milk [SEP] **Is Next**

[CLS] The man went to [MASK] store [SEP] Penguins [MASK] flightless birds [SEP] **Not Next**

BERT (2/2)

- 12-layer, 768-hidden, 12-heads, 110M parameters (Base)
- Wikipedia(that’s 2,500 million words) and Book Corpus (800 million words).

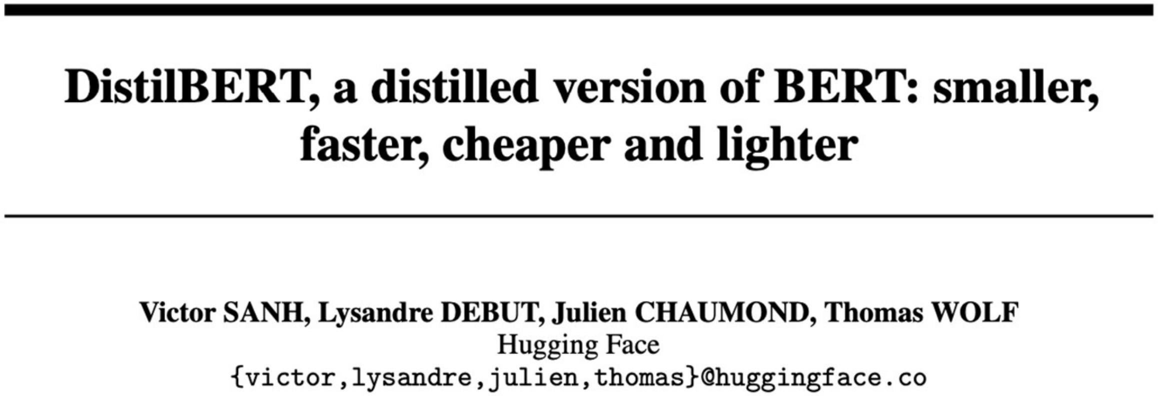




SYSTEM	Dev F1	Test F1
Fine-tuning approach		
$BERT_{LARGE}$	96.6	92.8
$BERT_{BASE}$	96.4	92.4
Feature-based approach ($BERT_{BASE}$)		
Embeddings	91.0	-
Second-to-Last Hidden	95.6	-
Last Hidden	94.9	-
Weighted Sum Last Four Hidden	95.9	-
Concat Last Four Hidden	96.1	-
Weighted Sum All 12 Layers	95.5	-

DistilBERT

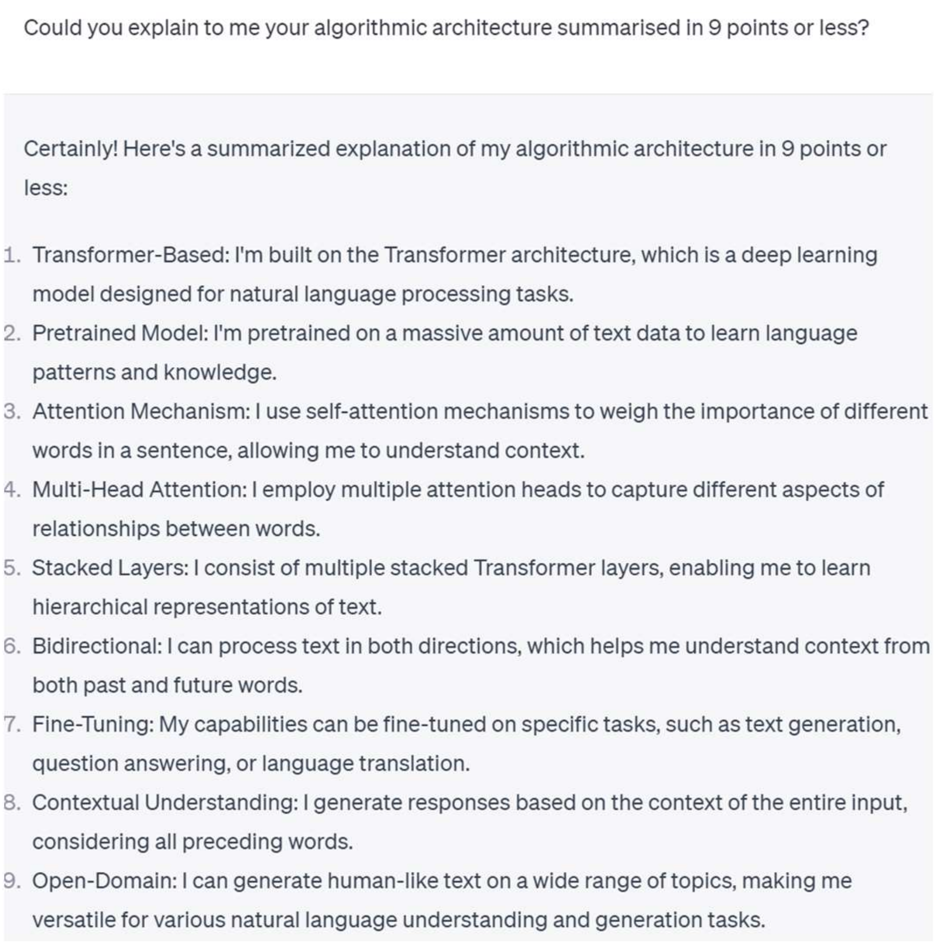
- Reduced size of BERT model by 40%
- Retain 97% of its language understanding capabilities
- 60% faster



DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF
Hugging Face
`{victor,lysandre,julien,thomas}@huggingface.co`

I asked ChatGPT…



Could you explain to me your algorithmic architecture summarised in 9 points or less?

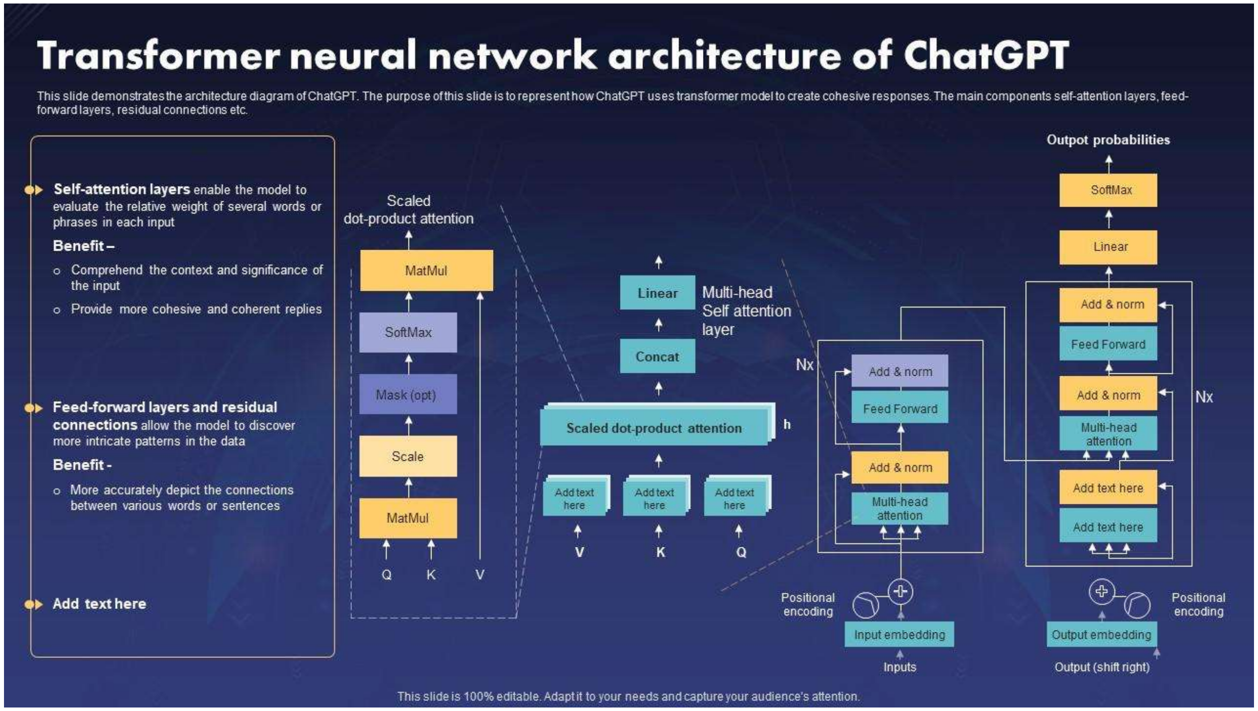
Certainly! Here's a summarized explanation of my algorithmic architecture in 9 points or less:

1. Transformer-Based: I'm built on the Transformer architecture, which is a deep learning model designed for natural language processing tasks.
2. Pretrained Model: I'm pretrained on a massive amount of text data to learn language patterns and knowledge.
3. Attention Mechanism: I use self-attention mechanisms to weigh the importance of different words in a sentence, allowing me to understand context.
4. Multi-Head Attention: I employ multiple attention heads to capture different aspects of relationships between words.
5. Stacked Layers: I consist of multiple stacked Transformer layers, enabling me to learn hierarchical representations of text.
6. Bidirectional: I can process text in both directions, which helps me understand context from both past and future words.
7. Fine-Tuning: My capabilities can be fine-tuned on specific tasks, such as text generation, question answering, or language translation.
8. Contextual Understanding: I generate responses based on the context of the entire input, considering all preceding words.
9. Open-Domain: I can generate human-like text on a wide range of topics, making me versatile for various natural language understanding and generation tasks.

What did you notice?

Did ChatGPT bore you with details about feed forward layers, activation functions etc.?

ChatGPT Architecture



Transformer neural network architecture of ChatGPT

This slide demonstrates the architecture diagram of ChatGPT. The purpose of this slide is to represent how ChatGPT uses transformer model to create cohesive responses. The main components self-attention layers, feed-forward layers, residual connections etc.

Self-attention layers enable the model to evaluate the relative weight of several words or phrases in each input

Benefit –

- Comprehend the context and significance of the input
- Provide more cohesive and coherent replies

Feed-forward layers and residual connections allow the model to discover more intricate patterns in the data

Benefit -

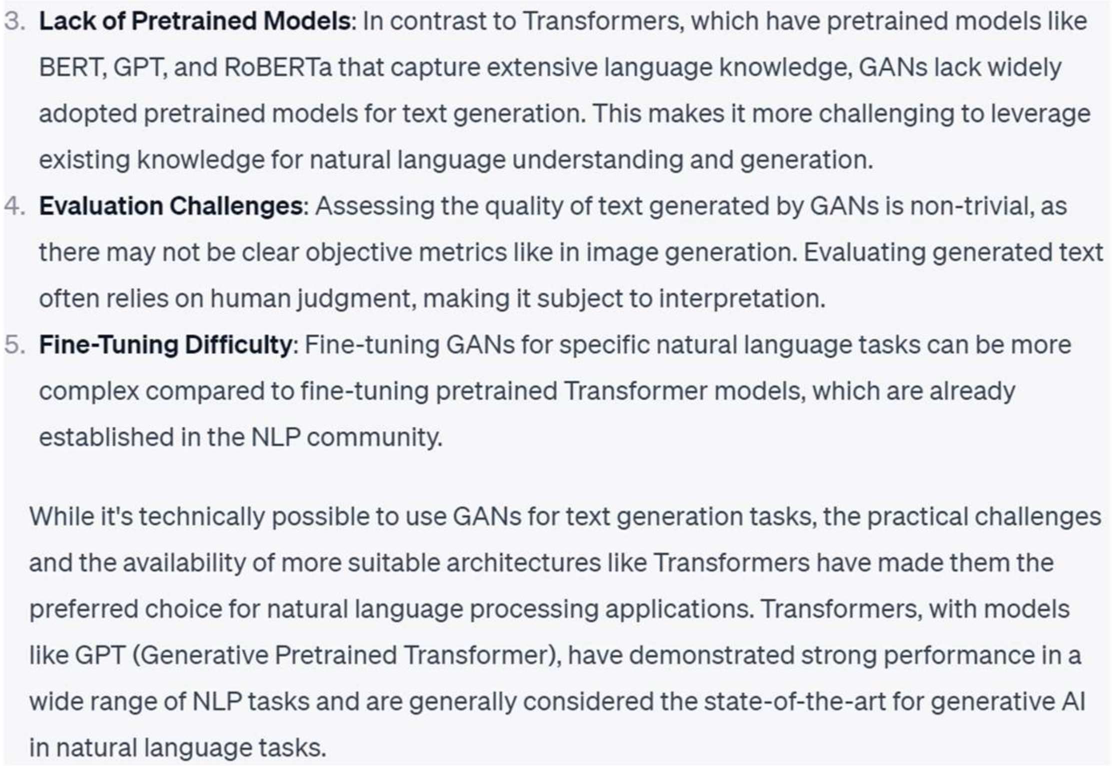
- More accurately depict the connections between various words or sentences

Add text here

This slide is 100% editable. Adapt it to your needs and capture your audience's attention.

Why GPT models?

- Alternatively, why choose GPT over BERT/GAN language models?



3. **Lack of Pretrained Models**: In contrast to Transformers, which have pretrained models like BERT, GPT, and RoBERTa that capture extensive language knowledge, GANs lack widely adopted pretrained models for text generation. This makes it more challenging to leverage existing knowledge for natural language understanding and generation.
4. **Evaluation Challenges**: Assessing the quality of text generated by GANs is non-trivial, as there may not be clear objective metrics like in image generation. Evaluating generated text often relies on human judgment, making it subject to interpretation.
5. **Fine-Tuning Difficulty**: Fine-tuning GANs for specific natural language tasks can be more complex compared to fine-tuning pretrained Transformer models, which are already established in the NLP community.

While it's technically possible to use GANs for text generation tasks, the practical challenges and the availability of more suitable architectures like Transformers have made them the preferred choice for natural language processing applications. Transformers, with models like GPT (Generative Pretrained Transformer), have demonstrated strong performance in a wide range of NLP tasks and are generally considered the state-of-the-art for generative AI in natural language tasks.

ChatGPT (architecture and use in trading)

- https://www.youtube.com/watch?app=desktop&v=bSvTVREwSNw
- https://www.youtube.com/watch?v=Z348tIeEqIk
- ChatGPT has been banned by major established institutions like Citibank and JPMorgan on its trading floors.
- Generative AI for trading ideas as a technique is mostly used by retail investors or start-ups.

https://theconversation.com/chatgpt-powered-wall-street-the-benefits-and-perils-of-using-artificial-intelligence-to-trade-stocks-and-other-financial-instruments-201436

References

- https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html
- https://huggingface.co/docs/transformers/tokenizer_summary
- https://www.ibm.com/cloud/learn/natural-language-processing#:~:text=Natural%20language%20processing%20(NLP)%20refers,same%20way%20human%20beings%20can.
- https://towardsdatascience.com/all-you-need-to-know-about-attention-and-transformers-in-depth-understanding-part-1-552f0b41d021
- https://arxiv.org/pdf/1910.01108.pdf
- https://ai.stackexchange.com/questions/28818/in-computer-vision-what-is-the-difference-between-a-transformer-and-attention
- https://www.analyticsvidhya.com/blog/2021/06/part-5-step-by-step-guide-to-master-nlp-text-vectorization-approaches/

Stay in Touch!

- My Telegram: @KC
- My school email: e0960154@u.nus.edu

Lab 5

NLP stock market sentiment analysis

Sentiment analysis

- https://github.com/avhirupc/Stock-Market-Prediction-Challenge/blob/master/Predicting%20Stock%20Prices%20Challenge.ipynb
- https://github.com/NUS-Fintech-Society/ML_FOMC-Project/blob/main/experiment.ipynb (experiment)

Example code for financial sentiment analysis [VADER]

Sentiment analysis model evaluation

Investment advice from GPT chatbot

- What prompts? How complex an interface to implement?
- https://github.com/KhanhHua2102/Monetize.ai
- https://github.com/marcelloinfante/finance-gpt/blob/main/app.py