UNIVERSITY OF PORTSMOUTH

Intelligent Data and Text Analytics

Text Mining – Deep Learning and Large Language Models

Text mining – DL and LLMs

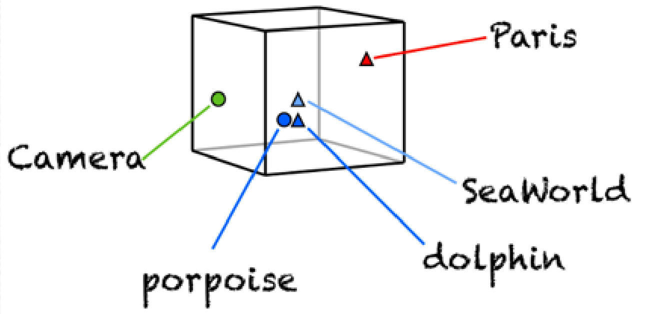
- Text representations with DL
- LLMs
- Bias and Limitations
- Lab work: LLMs for classification and topic modelling

Deep Learning and Text representations with DL

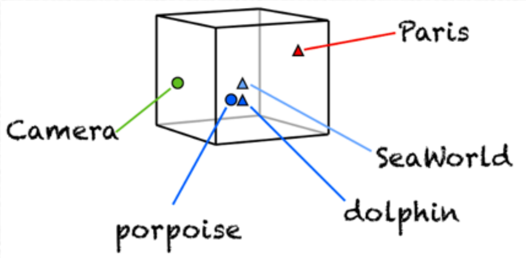
Big Data Challenge: The Curse of High-Dimensionality

- Text: Word co-occurrence statistics matrix

- High-dimensionality:
  - There are over **171k** words in English language
- Redundancy:
  - Many words share similar semantic meanings
    - Sea, ocean, marine..



Solution to Data Challenge: Dimension Reduction

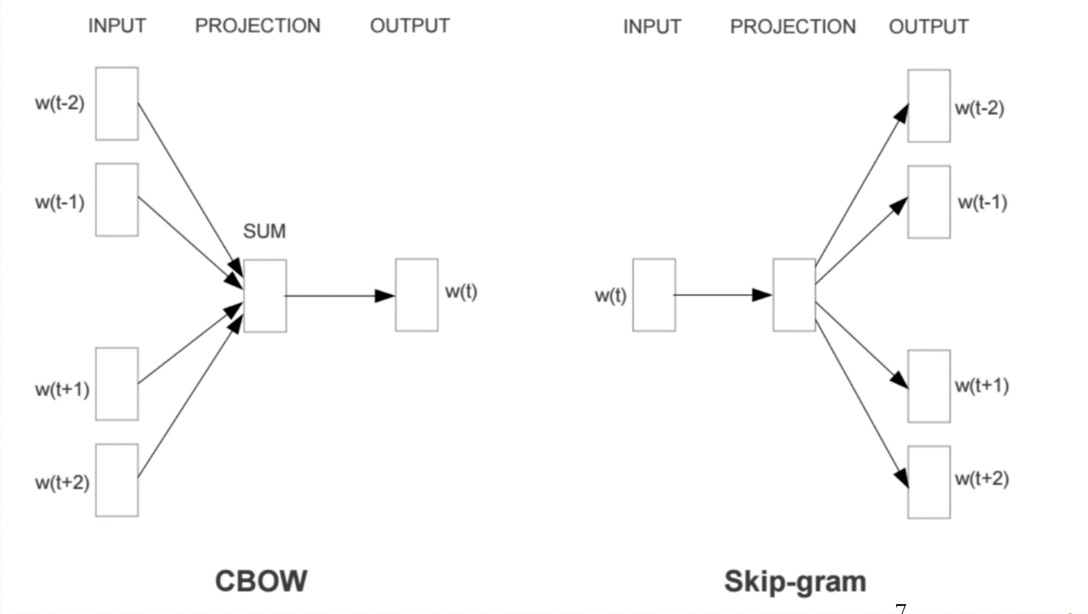


- Why Low-dimensional Space?
  - ❑ Visualization
  - ❑ Compression
  - ❑ Explanatory data analysis
  - ❑ Fill in (impute) missing entries (link/node prediction)
  - ❑ Classification and clustering
  - ❑ Identify / point
- How to automatically identify the lower-dimensional space that the high-dimensional data (approximately) lie in

Word2Vec: Word Embeddings

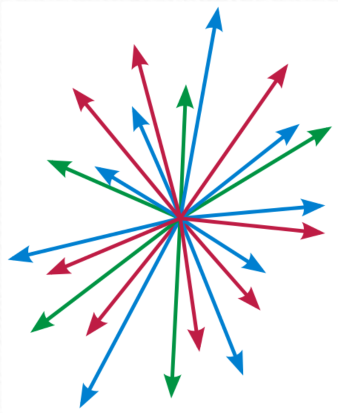
- Word2vec: created by T. Mikolov at Google (2013)
  - Input: a large corpus; output: a vector space, of 10^2 dimensions
  - Words sharing common contexts in close proximity in the vector space
- Embedding vectors created by Word2vec: better than LSA (Latent Semantic Analysis)

- Models: shallow, two-layer neural networks
- Two model architectures:
  - Continuous bag-of-words (CBOW)
    - Order does not matter, faster
  - Continuous skip-gram
    - Weigh nearby context words more heavily than more distant context words
    - Slower but better job for infrequent words



Predictive Text Embedding

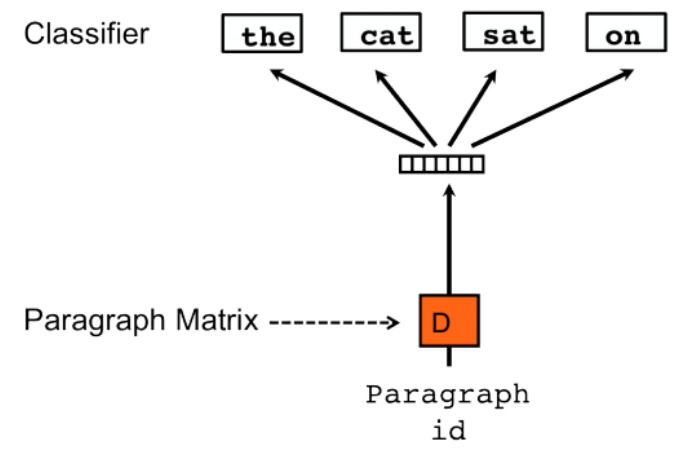
- Text Representation
  - Learning meaningful text representations is important for various machine learning tasks



Text Classification
Text Clustering
Retrieval

  - Bag of words:
    - Sparsity
    - Ignore the relatedness between different words

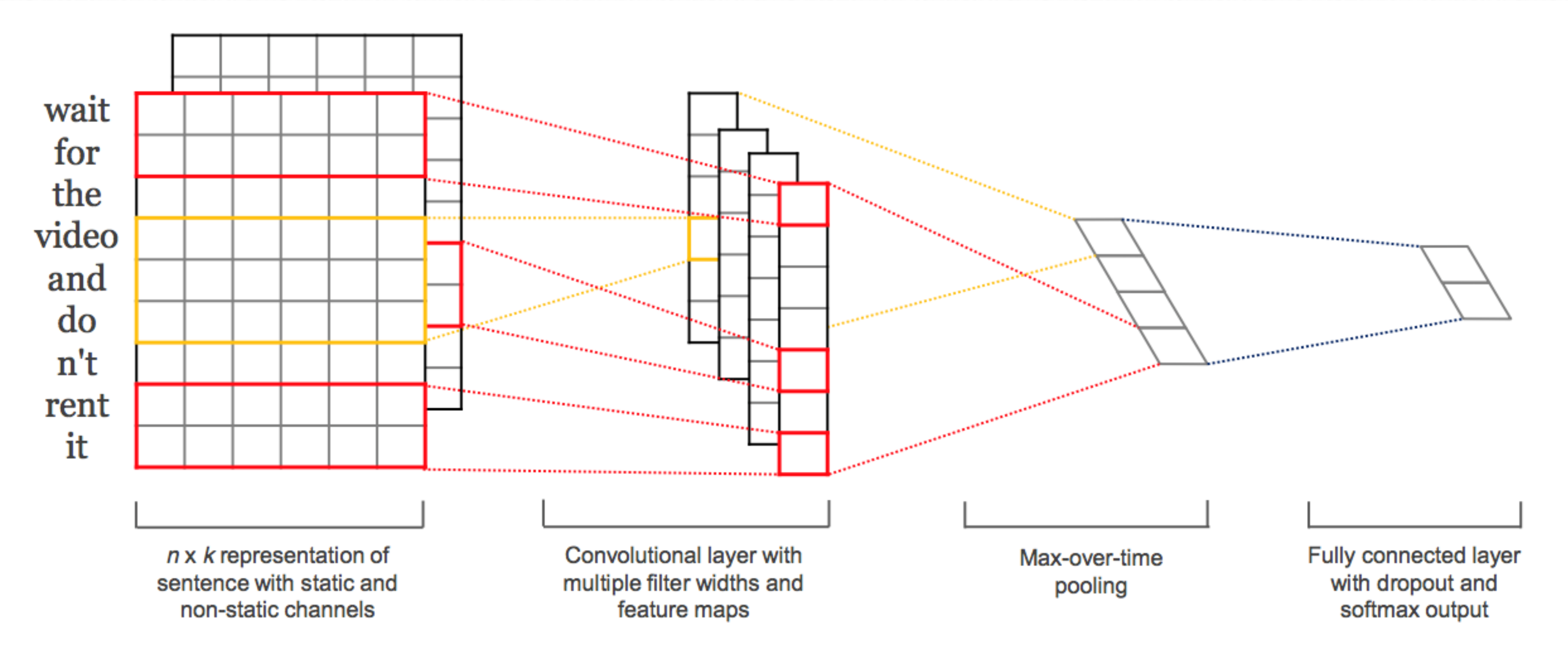
Doc2Vec: document embeddings



- **Distributed Representations**
  - Embed text into a low-dimensional space
  - Word2vec, Paragraph Vector (Doc2Vec)
    - word2Vec represents words as vectors
    - doc2Vec represents paragraphs/documents as vectors
- Strength:
  - Low-dimensional vectors; similar texts have similar vectors; efficient
- Weakness:
  - Totally unsupervised; Can't guide the training

CNNs

- **Convolutional Neural Networks**
  - Used for text classification
  - The hidden layer can be used for text representation
- Strength: High accuracy
- Weakness: Totally supervised;
  - Slow to train; Training is very tricky



- **Recurrent Neural Networks**
  - Memory-intensive
  - Slow to train

Large Language Models

What are LLMs?

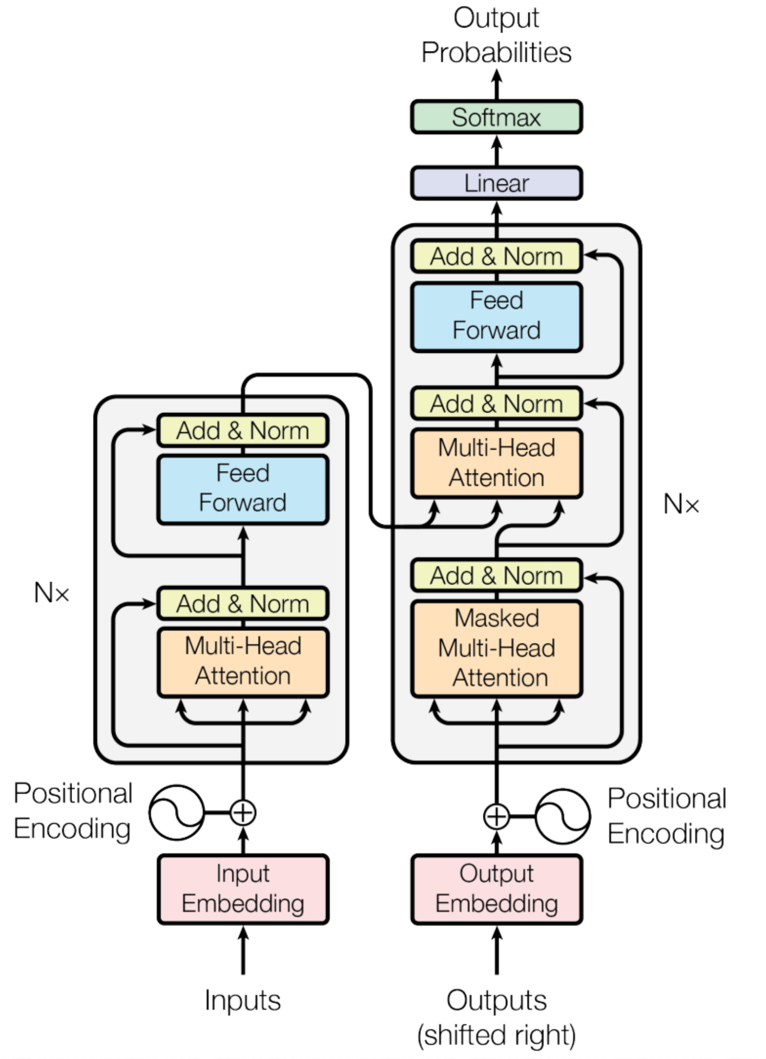
- LLMs are a subset of Deep Learning
- Generative AI is also a subset of Deep Learning
- LLMs key characteristics:
    - General-purpose language models that can be pre-trained and then
    - Fine-tuned for specific purposes

LLMs history overview

- 2017: Transformer LLM (Google)
- 2018: GPT (OpenAI) and BERT (Google)
- 2019: GPT2 and BART (Facebook/Meta)
- 2020: GPT3
- 2023: GPT4

Text Transformers

- 2017: Attention is all you need (initial focus was translation)
  - Previously: the need for complex recurrent or convolutional neural networks in an encoder-decoder configuration
  - What transformers changed:
    - new architecture based on just attention mechanisms
    - No need for recurrence and convolutions



Transformer architecture (from the original paper mentioned above)

Types of Transformers

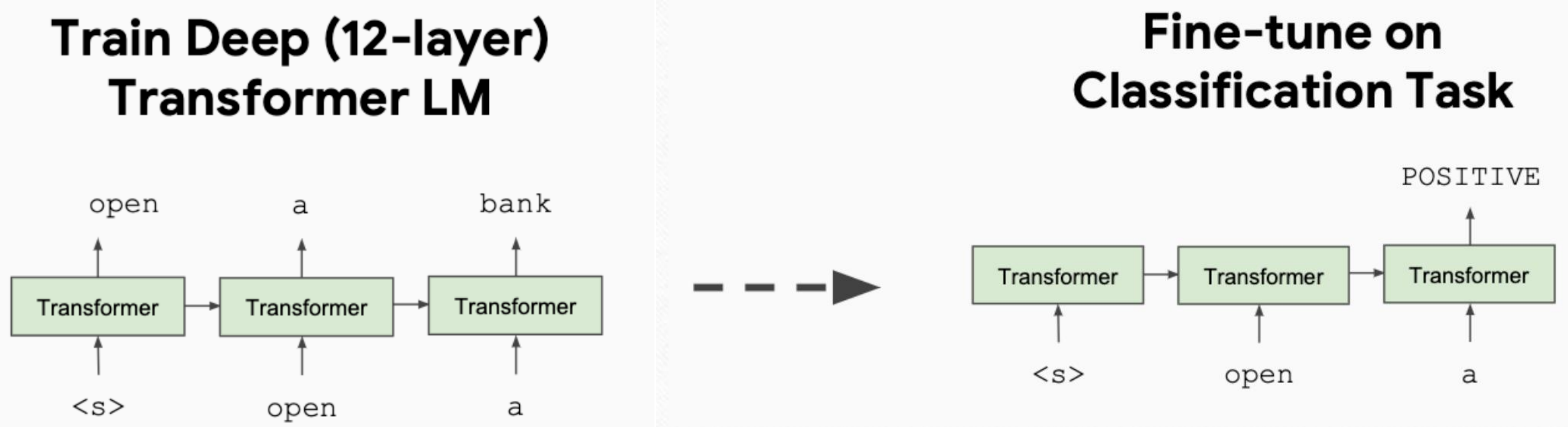
- Generic architecture: 2 parts – encoder and decoder
- Encoder-only models:
  - Good for tasks that require understanding of the input, such as sentence classification and named entity recognition
- Decoder-only models
  - Good for generative tasks such as text generation
- Encoder-decoder models or sequence-to-sequence models
  - Good for generative tasks that require an input, such as translation or summarization

Transformers and Language Models

- Transformers = language models
  - Trained on large volumes of raw text using self-supervisor learning (i.e., no need for labelled/annotated data)
  - due to the training, this is a generic model, but not very useful for a specific task
  - To be useful such a mode needs to be *fine-tuned* in a supervised way with annotated data
- Most models are Large Language Models
  - They are trained on huge amounts of data and have very large numbers of parameters; e.g., ChatGPT3 vas trained on 500 million text sources and has 175 billion parameters

OpenAI GPT (2018)

- Train one unidirectional LM (left-to-right) based on a deep Transformer decoder
- Fine-tuning approach: all pre-trained parameters are re-used & updated on downstream tasks
- Trained on 512-token segments on BooksCorpus — much longer context!



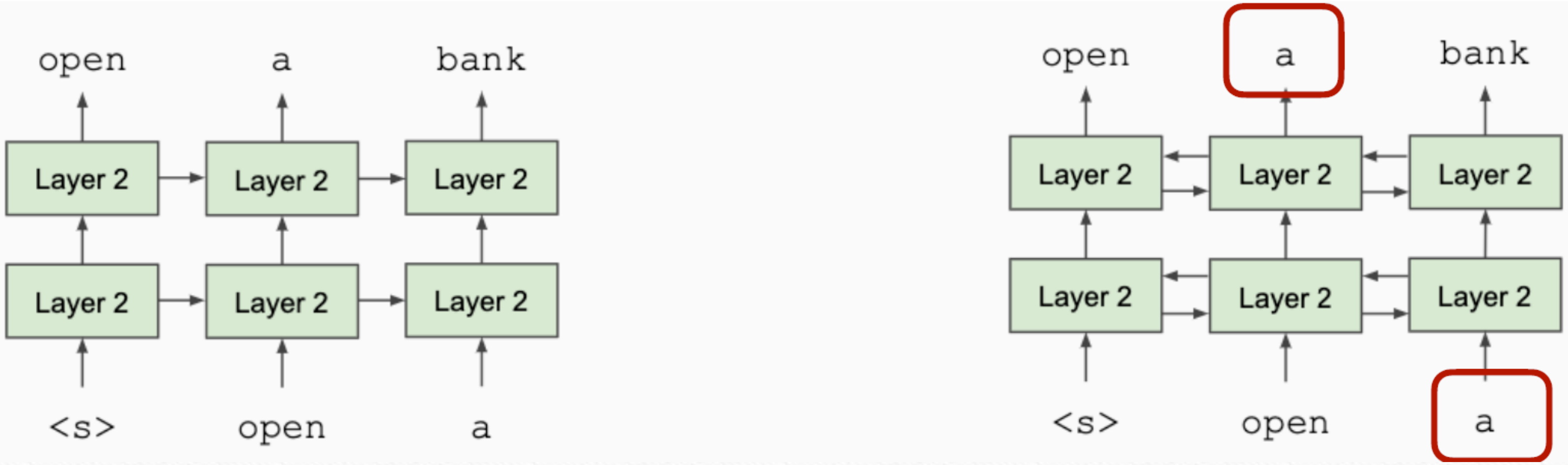
Images by Danqi Chen (https://www.cs.princeton.edu/~danqic/)

Google BERT (2018)

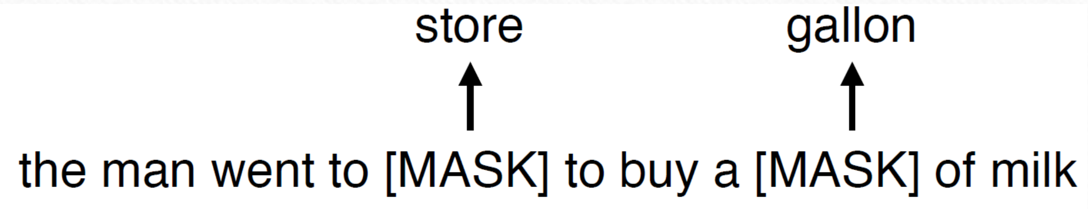
- It is a fine-tuning approach based on a deep Transformer encoder
- Learn representations based on <u>bidirectional</u> context
  - Both <u>left</u> and <u>right contexts</u> are important to understand the meaning of words.
    - Example1: we went to the river **<u>bank</u>**.
    - Example2: I need to go to **<u>bank</u>** to make a deposit.
- Pre-training objectives: masked language modelling + next sentence prediction
- State-of-the-art performance on a large set of sentence-level and token-level tasks

Masked Language Modelling (MLM)

- How to enable bidirectional learning, not just unidirectional (left-to-right as in GPT)



- Solution: Mask out k% of the input words, and then predict the masked words



MLM: masking rate and strategy

- What is the value of k?
  - They always use k = 15%.
  - little masking: computationally expensive
  - Too much masking: not enough context
- How are masked tokens selected?
  - 15% tokens are uniformly sampled
  - Is it optimal? Other strategies proposed later: span masking (Joshi et al., 2020) and PMI masking (Levine et al., 2021)

Next Sentence Prediction (NSP)

- Motivation: many NLP downstream tasks require understanding the relationship between two sentences (natural language inference, paraphrase detection, Question-Answering)
- NSP is designed to reduce the gap between pre-training and fine-tuning

[CLS]: a special token always at the beginning

[SEP]: a special token used to separate two segments

Input = [CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]

Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP] penguin [MASK] are flight ##less birds [SEP]

Label = NotNext

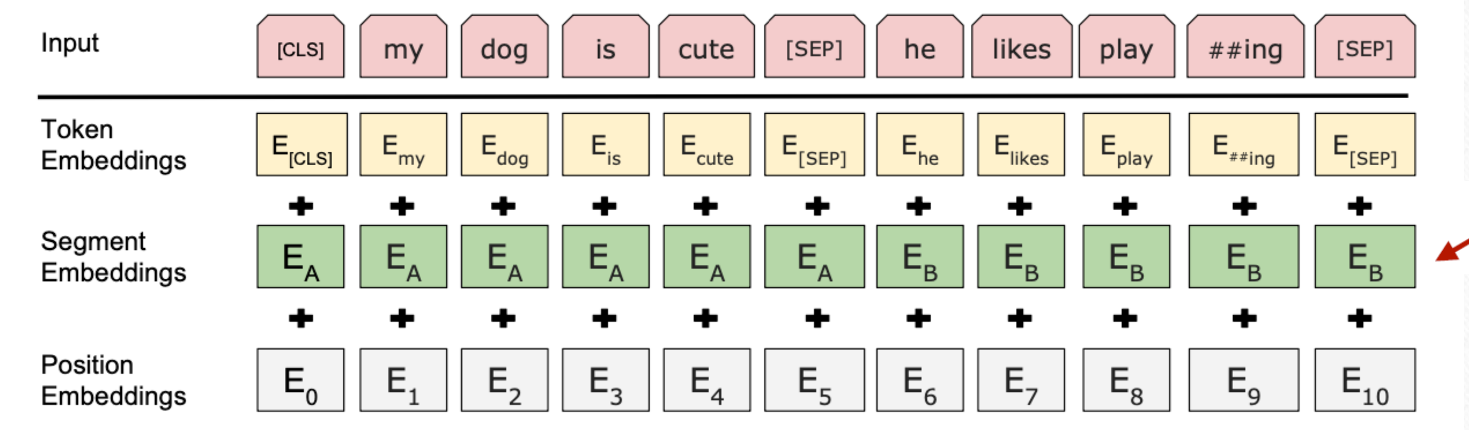
- They sample two contiguous segments for 50% of the time and another random segment from the corpus for 50% of the time

BERT pre-training (1)

- Vocabulary size: 30,000 workpieces (common sub-word units) (Wu et al., 2016)

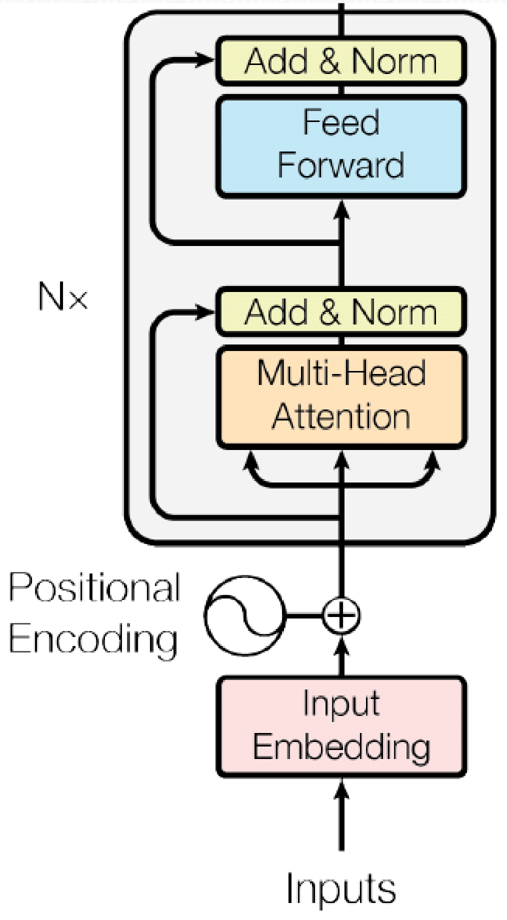
(Image: Stanford CS224N)

- Input embeddings:



Separate two segments

BERT pre-training (2)



- BERT-base: 12 layers, 768 hidden size, 12 attention heads, 110M parameters

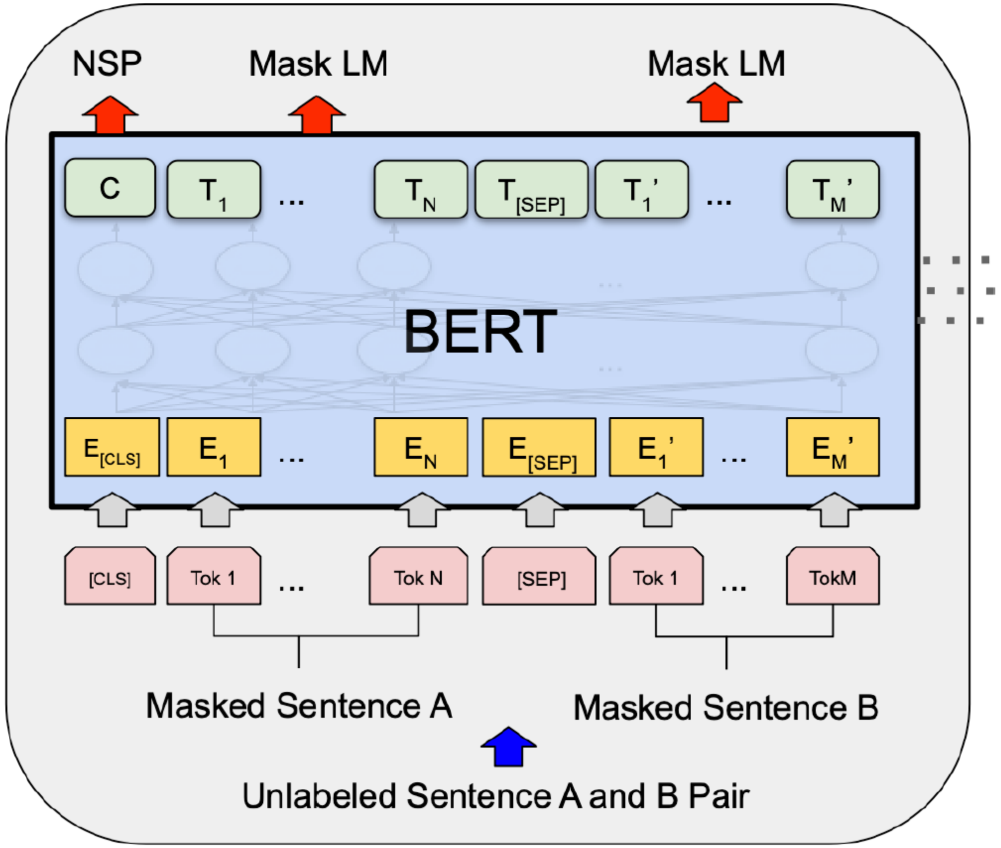
Same as OpenAI GPT

- BERT-large: 24 layers, 1024 hidden size, 16 attention heads, 340M parameters

OpenAI GPT was trained on BooksCorpus only!

- Training corpus: Wikipedia (2.5B) + BooksCorpus (0.8B)
- Max sequence size: 512 word pieces (roughly 256 and 256 for two non-contiguous sequences)#
- Trained for 1M steps, batch size 128k

BERT pre-training (3)



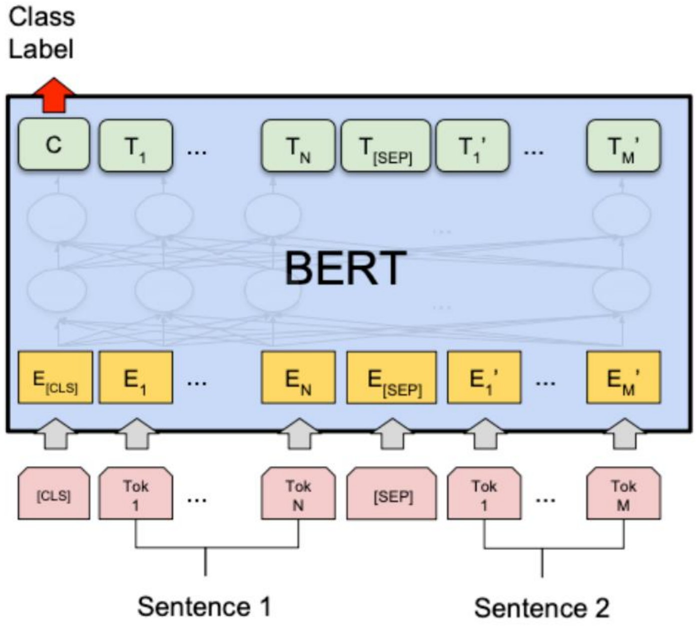
Pre-training

- MLM and NSP are trained together
- [CLS] is pre-trained for NSP
- Other token representations are trained for MLM

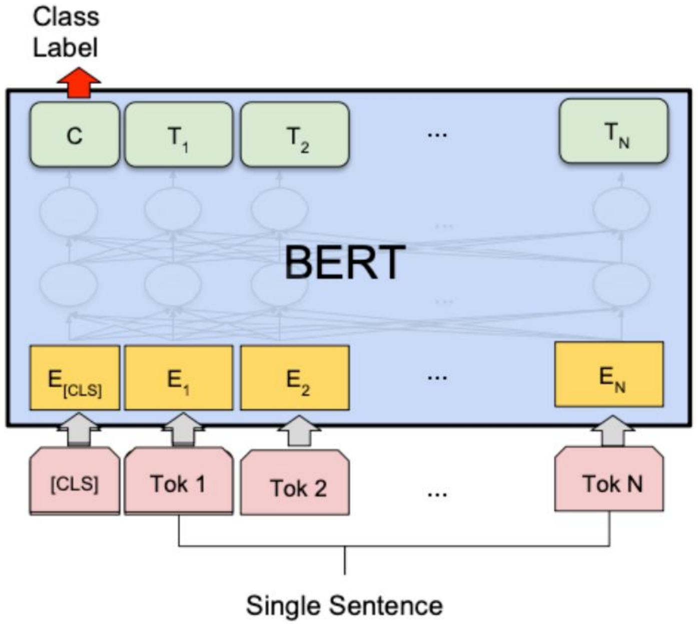
Fine-tuning BERT (1)

“Pretrain once, finetune many times.”

sentence-level tasks



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

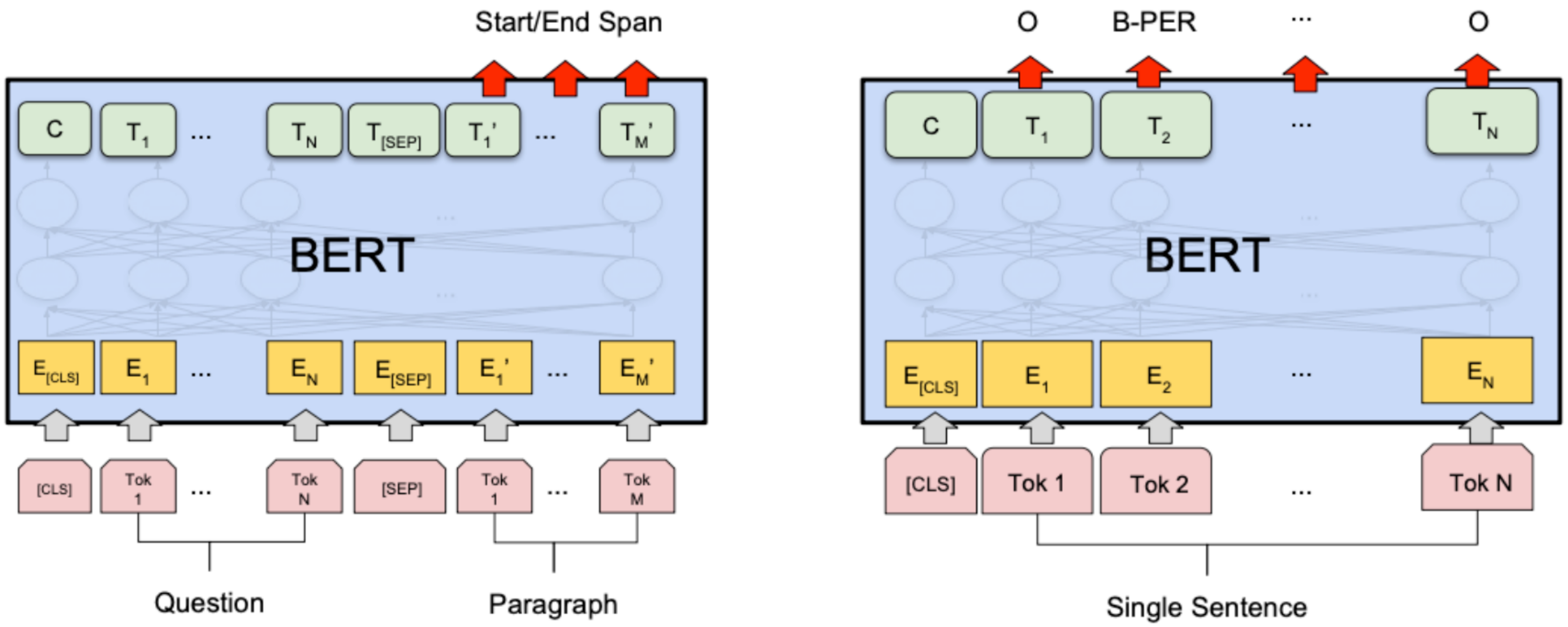


(b) Single Sentence Classification Tasks: SST-2, CoLA

Fine-tuning BERT (2)

"Pretrain once, finetune many times."

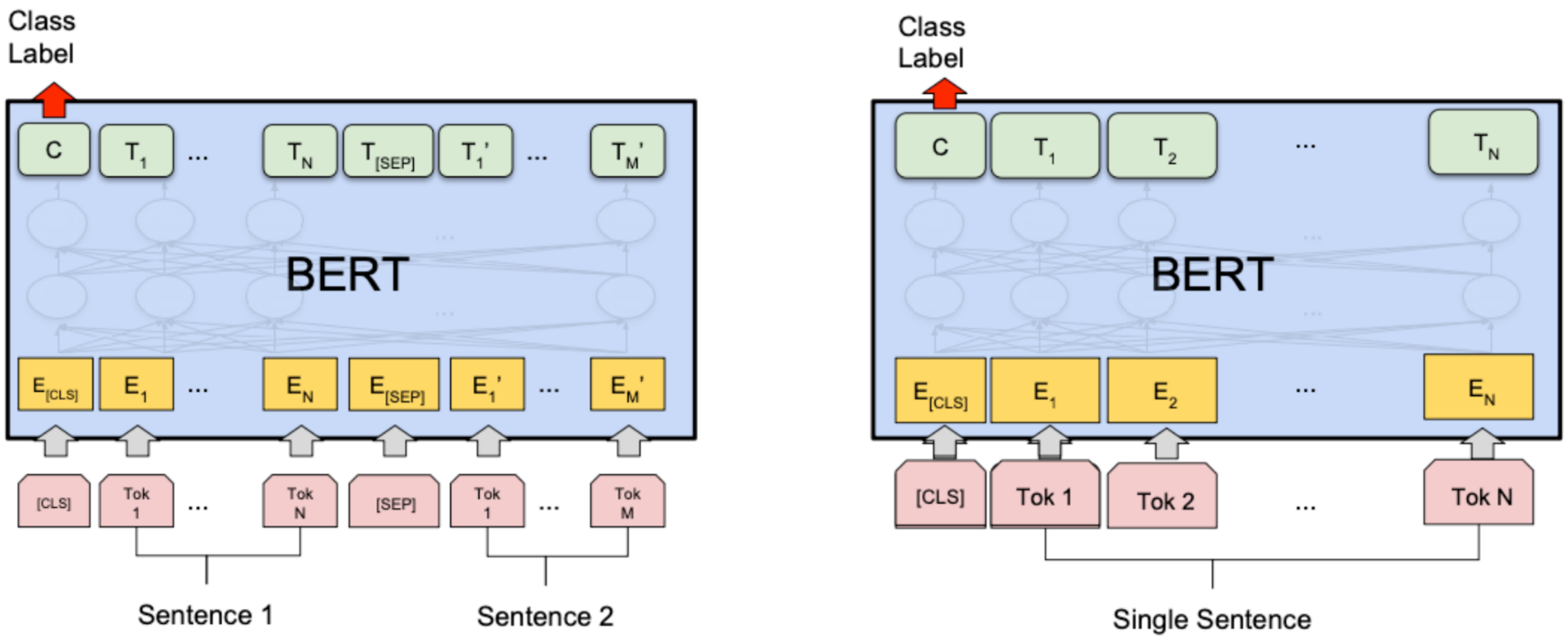
token-level tasks



(c) Question Answering Tasks:
SQuAD v1.1

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

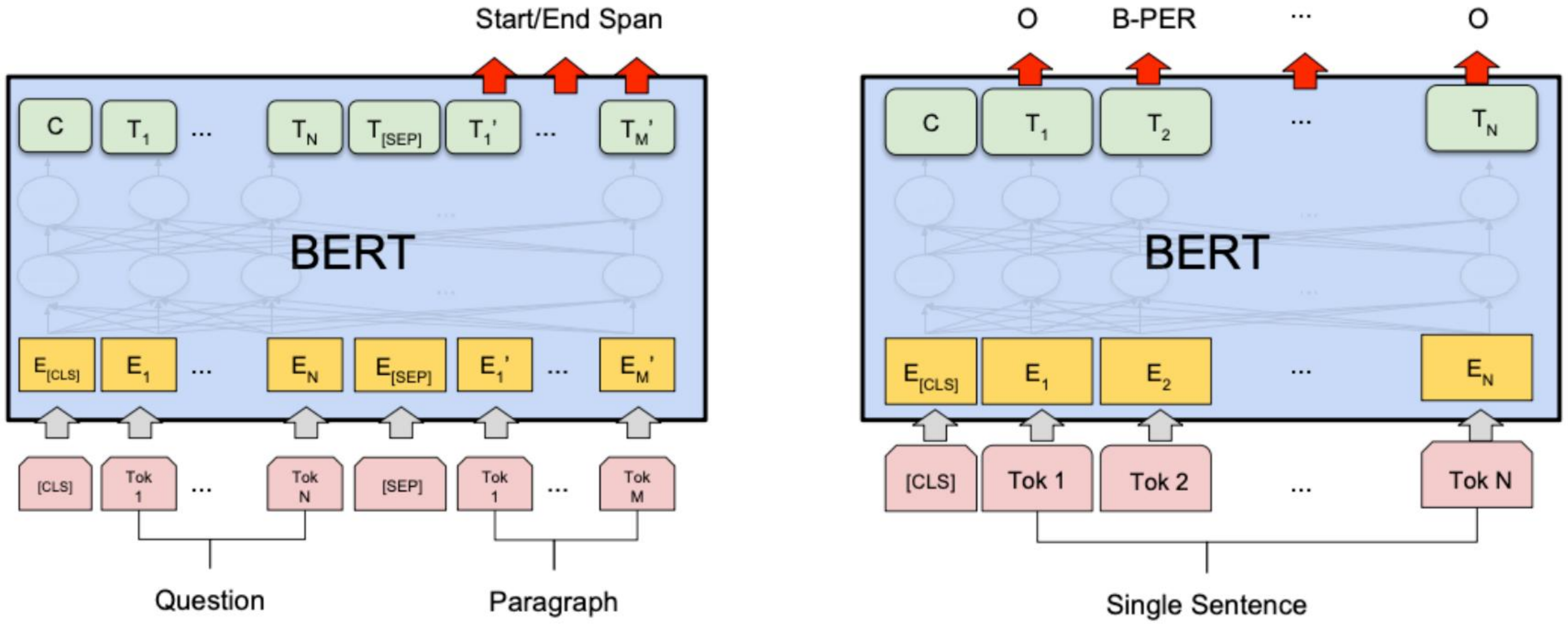
Fine-tuning BERT (3)



- For sentence pair tasks, use [SEP] to separate the two segments with segment embeddings
- Add a linear classifier on top of [CLS] representation and introduce $C \times h$ new parameters

C: # of classes, h: hidden size

Fine-tuning BERT (4)



- For token-level prediction tasks, add linear classifier on top of hidden representations

Q: How many new parameters?

LLMs after BERT

- RoBERTa (Liu et al., 2019)
  - Trained on 10x data & longer, no NSP
  - Much stronger performance than BERT (e.g., 94.6 vs 90.9 on SQuAD)
  - Still one of the most popular models to date
- ALBERT (Lan et al., 2020)
  - Increasing model sizes by sharing model parameters across layers
  - Less storage, much stronger performance but runs slower.
- ELECTRA (Clark et al., 2020)
  - It provides a more efficient training method by predicting 100% of tokens instead of 15% of tokens

- Models that handle long contexts (512 tokens)
  - Longformer, Big Bird, …
- Multilingual BERT
  - Trained single model on 104 languages from Wikipedia. Shared 110k WordPiece vocabulary
- BERT extended to different domains
  - SciBERT, BioBERT, FinBERT, ClinicalBERT, …
- Making BERT smaller to use
  - DistillBERT, TinyBERT, …

Bias and Limitations

Bias and Limitations

- LLMs are trained on available data regardless of quality or safeguards
- The pretrained models can generate sexist, racist, homophobic, offensive or harmful content
- The fine-tuning of the model on annotated data will not address this bias
- Other limitations
  - Toxicity
  - Disinformation

What is bias and why it matters?

- **Performance Disparities**
  - A system is more accurate for some demographic groups than others
- **Social Bias/Stereotypes**
  - A system's predictions contain associations between target concepts and demographic groups, and this effect is bigger for some demographic groups than for others
- Language models have new powerful capabilities
- This leads to increased adoption
- This leads to increased harms

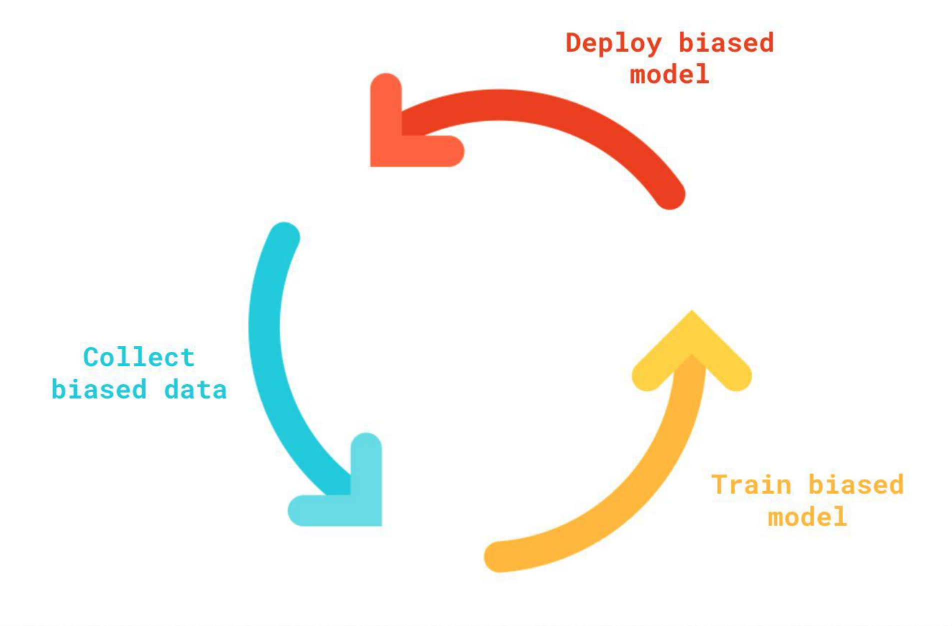


Image: Richard Zhu & Maxine Perroni-Scharf, Princeton University

What is Toxicity?

- Generation of rude, disrespectful, or unreasonable text that would make someone want to leave a conversation.
- In neural LLM's, causal phenomenon known as neural toxic degeneration
- The definition of what constitutes toxicity varies
- Why do we care about toxicity?
  - Downstream users may include younger or more vulnerable audiences
  - Unintended outputs for given task

Disinformation

- Generating misleading content
- Misinformation: false or misleading information, regardless of intention
- Disinformation: false or misleading information to **intentionally** deceive a target population
- Excludes: fictional literature, satire

Image Source: Zellers et al., 2020

Disinformation

- Motivation
  - Language models are steadily increasing in size
  - This has resulted in an increase in number of training tokens to maintain performance improvements
  - This demand for larger datasets has meant drawing from lower quality sources
  - Large language models may act as stochastic parrots, repeating potentially dangerous text

Acknowledgments

- Slides have been compiled from several sources:
- Danqi Chen, Lecture slides on Large Language Models, Princeton University
- Richard Zhu & Maxine Perroni-Scharf, Princeton University
- David Wolfe Corne, Lecture slides on Deep Learning, Heriot Watt University
- Li Deng, Deep Learning Technology Center (DLTC), Microsoft Research