# Modern Natural Language Processing: Where do we go from here?

Antoine Bosselut

#### **Section Outline**

- Advances: NLP Successes, Pretraining, Scale
- Robustness, Multimodality

New Problems: Prompting, Knowledge & Reasoning, Retrieval-Augmentation,

# Core Methods

## Word Embeddings

- Words and other tokens become vectors; no longer discrete symbols!
  - assign to a vector
- Word embeddings can be learned in a self-supervised manner from large quantities of raw text
  - Learning word embeddings from scratch using labeled data for a task is datainefficient!
- Three main algorithms: Continuous Bag of Words (CBOW), Skip-gram, and GloVe

# Need to define a vocabulary of words (or token types) V that our system can

## LMs & RNNs

- context
- Recurrent neural networks can theoretically learn to model an unbounded context length
  - no increase in model size because weights are shared across time steps
- long-range dependencies
- problem
  - used for for many sequence-to-sequence tasks

• Language models learn to estimate the distribution over the next word given a

Early neural LMs (and n-gram models) suffered from fixed context windows

• Practically, however, vanishing gradients stop vanilla RNNs from learning useful

• LSTMs are variants of recurrent networks that mitigate the vanishing gradient

## Transformers

- learning long-range dependencies
- state, so must be **computed in series**
- Attention: Direct connections between output states and inputs (solves temporal bottleneck)
  - for encoding
- embeddings to capture sequence order

• Temporal Bottleneck: Vanishing gradients stop many RNN architectures from

• Parallelisation Bottleneck: RNN states depend on previous time step hidden

• Self-Attention: Remove recurrence over input, allowing parallel computation

• Transformers use self-attention to encode sequences, but now require position

#### **Text Generation**

- Text generation is the foundation of many useful NLP applications (e.g., translation, summarisation, dialogue systems)
- **Autoregressive**: models generate one token a time, using the context and previously generated tokens as inputs to generate the next token
- Teacher forcing is the premier algorithm for training text generators
- A variety of decoding algorithms can be used to generate text from models, each trading off expected quality vs. diversity in different ways.
- Automatic evaluation of NLG systems (content overlap, model-based, human) is difficult as most metrics fall short of reliable estimates of output quality



## **Deep Learning Successes in NLP**

#### The New York Times

FEATURE

#### The Great A.I. Awakening

How Google used artificial intelligence to transform Google Translate, one of its more popular services — and how machine learning is poised to reinvent computing itself.

> **The New York Times** A Breakthrough for A.I. Technology: Passing an 8th-Grade Science Test

#### The New York Times

#### Finally, a Machine That **Can Finish Your Sentence**

Completing someone else's thought is not an easy trick for A.I. But new systems are starting to crack the code of natural language.

#### THE **NEW YORKER**

## The Next Word

Where will predictive text take us?

Text by John Seabrook





#### What did these ingredients propel ?

## Pretraining

#### Massive Text Corpus

dia III da 1 Co

Critics say that current voting systems used in the United States are inefficient and often lead to the inaccurate counting of votes. Miscounts can be especially damaging if an election is closely contested. Those critics would like the traditional systems to be replaced with far more efficient and trustworthy computerized voting systems.

In traditional voting, one major source of inaccuracy is that people accidentally vote for the wrong candidate. Voters usually have to find the name of their candidate on a large sheet of paper containing many names-the ballot-and make a small mark next to that name. People with poor eyesight can easily mark the wrong name. The computerized voting machines have an easy-to-use touch-screen technology: to cast a vote, a voter needs only to touch the candidate's name on the screen to record a vote for that candidate; voters can even have the computer magnify the name for easier viewing.

Another major problem with old voting systems is that they rely heavily on people to count the votes. Officials must often count up the votes one by one, going through every ballot and recording the vote. Since they have to deal with thousands of ballots, it is almost inevitable that they will make mistakes. If an error is detected, a long and expensive recount has to take place. In contrast, computerized systems remove the possibility of human error, since all the vote counting is done quickly and automatically by the computers.

Finally some people say it is too risky to implement complicated voting technology nationwide. But without giving it a thought, governments and individuals alike trust other complex computer technology every day to be perfectly accurate in banking transactions as well as in the communication of highly sensitive information.

#### Transformer Language Model



(Radford et al., 2018, 2019, many others)























## **Pretraining: Two Approaches**

#### (Causal, Left-to-right) Language Modeling

I really enjoyed the movie we watched on



(Radford et al., 2018, 2019, many others)



#### Masked Language Modeling

#### I really enjoyed the \_\_\_\_\_ we watched on Saturday!





(Devlin et al., 2018; Liu et al., 2020)

## Fine-tuning a single model



RTE, SWAG

#### • Prepend special token [CLS]: Classify output embedding for this token

#### Devlin et al. (2019)

## Fine-tuning a single model



- Prepend special token [CLS]: Classify output embedding for this token
- labelling tasks, and many more!

• Can use same model for classification tasks, sentence pair tasks, sequence

#### Devlin et al. (2019)

## Pretraining Improvements!

	Rank	Name	Model	URL	Score	BoolQ	СВ	COPA	MultiRC	ReCoRD	RTE	WiC	WSC	AX-b	AX-g
	1	Liam Fedus	SS-MoE		91.0	92.3	96.9/98.0	99.2	89.2/65.2	95.0/94.2	93.5	77.4	96.6	72.3	96.1/94.1
	2	Microsoft Alexander v-team	Turing NLR v5		90.9	92.0	95.9/97.6	98.2	88.4/63.0	96.4/95.9	94.1	77.1	97.3	67.8	93.3/95.5
	3	ERNIE Team - Baidu	ERNIE 3.0		90.6	91.0	98.6/99.2	97.4	88.6/63.2	94.7/94.2	92.6	77.4	97.3	68.6	92.7/94.7
+	4	Zirui Wang	T5 + UDG, Single Model (Google Brain)		90.4	91.4	95.8/97.6	98.0	88.3/63.0	94.2/93.5	93.0	77.9	96.6	69.1	92.7/91.9
+	5	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.3	90.4	95.7/97.6	98.4	88.2/63.7	94.5/94.1	93.2	77.5	95.9	66.7	93.3/93.8
	6	SuperGLUE Human Baseline	s SuperGLUE Human Baselines		89.8	89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0	76.6	99.3/99.7
+	7	T5 Team - Google	T5		89.3	91.2	93.9/96.8	94.8	88.1/63.3	94.1/93.4	92.5	76.9	93.8	65.6	92.7/91.9

# Superhuman results on benchmark datasets!

#### All top models use pretrained transformers!

#### Scale: Parameters



#### Time

#### Scale: Data

ELMo: 1B training tokens BERT: 3.3B training tokens RoBERTa: ~30B training tokens

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

![](_page_15_Picture_5.jpeg)

### Scale: Flops

#### **Total Compute Used During Training**

![](_page_16_Figure_2.jpeg)

Training Petaflop/s-days

![](_page_16_Picture_6.jpeg)

## Scaling Laws

![](_page_17_Figure_1.jpeg)

**Figure 1** Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute<sup>2</sup> used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

![](_page_17_Figure_5.jpeg)

![](_page_18_Picture_0.jpeg)

## Why do we want to make these models as big as possible?

## Fine-tuning a single model

![](_page_19_Figure_1.jpeg)

- Prepend special token [CLS]: Classify output embedding for this token
- labelling tasks, and many more!

• Can use same model for classification tasks, sentence pair tasks, sequence

#### Devlin et al. (2019)

## **Efficient tuning: LoRA**

![](_page_20_Figure_1.jpeg)

- During fine-tuning:
  - Keep all pretrained parameters frozen
  - LoRA: Initialise new Feedforward Net (FFN) \_ alongside components of transformer blocks
  - Keep these FFN layers limited in number of \_ parameters
    - # parameters in FFN layers is 2 \* d \* r, so keep r small
    - r is hidden dimension of FFN
  - Only update these **FFN** layers —

![](_page_20_Picture_11.jpeg)

Hu et al., (2021)

#### In-context Learning: A new paradigm!

Prompts in Red. GPT3 responses in Blue

Q. How do you draw a bicycle with shapes? A. You draw a bicycle with two circles connected by a line.

Q. How do you draw a ball with shapes? A. You draw a ball with one circle.

Q. How do you draw a house with shapes? A. You draw a house with a trapezium on top of a square.

Q. How do you draw a star with shapes? A. You draw a star with five triangles.

Q. How do you draw a clock with shapes? A. You draw a clock with two circles connected by a line.

Q. How do you draw a chair with shapes? A. You draw a chair with a square on top of a diamond.

Q. How do you draw a telephone with shapes? A. You draw a telephone with two ovals connected by a line.

- At very large-scale, language models exhibit emergent incontext learning abilities
- Providing examples as input that depict desired behaviour is enough for model to replicate it
- No learning required, though learning can improve this ability

## **Chain-of-thought Reasoning**

#### **Standard Prompting**

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Input

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

![](_page_22_Picture_5.jpeg)

## Model self-rationalizes through text generation

#### **Chain of Thought Prompting**

#### Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

#### Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9.

# What do those two abilities remind you of?

#### ChatGPT!

AN

answers?

![](_page_24_Picture_3.jpeg)

such as instructions or questions.

at following instructions and providing accurate explanations.

#### What language model is capable of following instructions and providing explanations for its

As an Al language model, I am an example of a system that is capable of following instructions and providing explanations for my answers. I am based on the GPT-4 architecture, developed by OpenAI. The GPT series of models, including GPT-3 and GPT-4, are designed to generate coherent and contextually relevant text based on given prompts,

These models can understand and respond to a wide range of requests, including providing explanations for their responses. However, it is important to note that AI language models like GPT-4 are not perfect and can sometimes generate incorrect or incomplete information. As AI continues to evolve, newer and more advanced models will likely become even better

## Alignment

Step 1

Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

We give treats and punishments to teach...

 $\bigcirc$ 

Explain reinforcement

learning to a 6 year old.

This data is used to fine-tune GPT-3.5 with supervised learning.

# BBB

Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

## Large Models can be Aligned to new Behaviors

![](_page_25_Figure_15.jpeg)

#### Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

![](_page_25_Figure_23.jpeg)

## Outcome: Many Tasks

#### AN

Can you give me a two-day itinerary for a weekend trip to Paris ? Please don't include more than three activities per day.

#### Day 1:

- 1. Visit the Eiffel Tower Start your day with the iconic symbol of Paris, the Eiffel Tower Take a stroll around the Champ de Mars and enjoy the views from the tower.
- 2. Explore the Louvre Museum After lunch, head to the Louvre Museum and admi world-famous artwork such as the Mona Lisa and the Venus de Milo.
- 3. Stroll along the Seine River In the evening, take a romantic stroll along the Seine and enjoy the Parisian atmosphere.

#### Day 2:

- 1. Visit Notre-Dame Cathedral Start your day by visiting one of the most famous cathedrals in the world, Notre-Dame Cathedral.
- 2. Explore the Marais neighborhood After lunch, explore the Marais neighborhood known for its historical architecture, trendy boutiques, and restaurants.
- 3. Enjoy a Cabaret Show In the evening, enjoy a cabaret show at the famous Paris venue, the Moulin Rouge, and end your trip with a memorable experience.

	AN	Can you generate the Python code for an attention mechanism ?	ľ
de	\$	Here is an example of a simple attention mechanism implemented in Python:	ம்
		Ô	Copy code
		import numpy as np	
		<pre>def attention(query, key, value, mask=None, dropout=None): "Compute 'Scaled Dot Product Attention'"</pre>	
		$d_k = query.shape[-1]$	
ower.		<pre>scores = np.matmul(query, key.transpose()) / np.sqrt(d_k)</pre>	
		if mask is not None:	
ire the		scores = scores - mask * 1e9	
		<pre>scores = np.exp(scores)</pre>	
ne River		if dropout is not None:	
		<pre>scores = dropout(scores)</pre>	
		<pre># Normalize the scores to weights</pre>	
		<pre>weights = scores / np.sum(scores, axis=-1, keepdims=True)</pre>	
		<pre># Apply the weights to the values</pre>	
		output = np.matmul(weights, value)	
Gothic		return output, weights	
d,		This function computes the attention weights and outputs using the cooled dat	product
,		ettention method, as described in the transformer probite stars. It takes a include	
		attention method, as described in the transformer architecture. It takes as input	t the query,
sian		key, and value vectors, as well as optional arguments for masking padded value	es and
		applying dropout to the attention scores.	
		Note that this is just a simple example, and there are many variations and impro	vements

Note that this is just a simple example, and there are many variations and improvements that can be made to the attention mechanism, such as using other methods for calculating the attention scores or incorporating more sophisticated masking schemes.

![](_page_26_Picture_14.jpeg)

#### **Outcome:** Personalization

- Explain the causes of World War I in a three paragraph essay. Try to use as many AN complicated words as possible.
- \$ The causes of World War I can be traced back to several factors, including imperialism, nationalism, and the complex network of alliances between countries. Imperialism, or the pursuit of territorial expansion and political power, was a driving force behind the conflict as various nations sought to increase their influence and control over regions and resources. Nationalism, or a strong sense of pride and loyalty to one's country, was also a major contributing factor, as many people believed that their nation was superior to others and sought to assert its dominance.

The complex network of treaties and alliances between countries created an intricate web of obligations that ultimately led to the outbreak of war. The Triple Entente, comprising of France, Russia, and Britain, was formed in response to the formation of the Central Powers, including Germany, Austria-Hungary, and the Ottoman Empire. The system of alliances meant that, when one country was attacked, its allies were obligated to come to its aid, thereby drawing multiple nations into the conflict.

Additionally, the powder keg of Europe was primed for an explosion with the proliferation of military build-up, as nations sought to increase their military might. Tensions were further exacerbated by the assassination of Archduke Franz Ferdinand of Austria-Hungary by a member of the Serbian nationalist group, the Black Hand, in 1914. This event, known as the Sarajevo Incident, set off a chain of events that eventually led to the declaration of war. The complex interplay of these various factors ultimately culminated in the outbreak of World War I, a conflict that would shape the course of history for generations to come.

Explain the causes of world war 1 to a 10 year old

\$

World War 1 was a big war that started in 1914 and lasted until 1918. It was fought between many countries, including Britain, France, and Russia on one side, and Germany, Austria-Hungary, and the Ottoman Empire on the other.

The main cause of World War 1 was a disagreement between countries over who was in charge of certain territories and the assassination of Archduke Franz Ferdinand of Austria-Hungary by a member of a Serbian nationalist group. This event led to a chain of events that eventually led to the start of the war.

Another factor that contributed to the start of the war was the military alliances between countries. Some countries had made agreements to support each other if one of them was attacked, which meant that if one country went to war, others would join too.

In short, World War 1 was caused by a combination of factors, including political and military tensions, territorial disputes, and alliances between countries that made it more likely for a small conflict to escalate into a large war.

![](_page_27_Picture_15.jpeg)

![](_page_28_Picture_0.jpeg)

#### Why are these language models so effective at scale?

## Encoded Knowledge

World knowledge is *implicitly* encoded in LM parameters! (e.g., that barbershops are places to get buzz cuts)

Bob went to the <MASK> to get a buzz cut BERT (**teacher**): 24 layer Transformer barbershop: 54% barber: 20% salon: 6% stylist: 4%

. . .

![](_page_29_Picture_7.jpeg)

![](_page_30_Picture_0.jpeg)

## Why might this be a bad idea?

In these language models, the learned world knowledge is stored *implicitly* in the parameters of the underlying neural network. This makes it difficult to determine what knowledge is stored in the network and where. Furthermore, storage space is limited by the size of the network—to capture more world knowledge, one must train ever-larger networks, which can be prohibitively slow or expensive.

![](_page_31_Picture_3.jpeg)

![](_page_32_Picture_0.jpeg)

#### What could we do instead?

## **Retrieval-Augmented LLMs**

![](_page_33_Figure_1.jpeg)

![](_page_33_Figure_2.jpeg)

![](_page_33_Figure_5.jpeg)

![](_page_33_Picture_7.jpeg)

Unlabeled text, from pre-training corpus  $(\mathcal{X})$  The [MASK] at the top of the pyramid (x)

![](_page_34_Picture_3.jpeg)

![](_page_35_Picture_0.jpeg)

corpus  $(\mathcal{Z})$ 

![](_page_35_Picture_1.jpeg)

Unlabeled text, from pre-training corpus  $(\mathcal{X})$ The [MASK] at the top of the pyramid (x)

knowledge - - - - - Neural Knowledge Retriever  $\sim p_{\theta}(z|x)$ 

![](_page_35_Picture_6.jpeg)

![](_page_36_Figure_0.jpeg)

![](_page_36_Picture_3.jpeg)

![](_page_37_Figure_0.jpeg)

[CLS] The [MASK] at [SEP] The pyramidio material higher

om pre-training corpus (X)
le top of the pyramid $(x)$
nowledge Retriever $\sim p_{ heta}(z x)$
nent on top allows for less
er up the pyramid. $(z)$
t
t the top of the pyramid on on top allows for less up the pyramid. $(x, z)$

![](_page_37_Picture_5.jpeg)

![](_page_38_Figure_0.jpeg)

![](_page_38_Picture_3.jpeg)

![](_page_38_Picture_6.jpeg)

## **Retrieval-Augmented LLMs**

- Readily available at scale, requires no processing
- We have powerful methods for encoding text (e.g., BERT)

![](_page_39_Picture_7.jpeg)

## **Retrieval-Augmented LLMs**

- Readily available at scale, requires no processing
- We have powerful methods for encoding text (e.g., BERT)
- However, these methods don't really work yet with larger units of text (e.g., books)
- "Long-context" NLP is an active area of research!

![](_page_40_Picture_7.jpeg)

![](_page_41_Picture_0.jpeg)

#### Why might we want to integrate retrieval components?

## **Remaining Problems!**

![](_page_42_Figure_1.jpeg)

#### Robustness

# Deep learning models exploit biases (Bolukbasi et al., 2016), annotation artifacts (Gururangan et al., 2018), surface patterns (Li & Gauthier, 2017), etc.

## They struggle to learn robust understanding abilities

![](_page_43_Picture_3.jpeg)

## Multimodality

![](_page_44_Picture_1.jpeg)

A woman is throwing a <u>frisbee</u> in a park.

![](_page_44_Picture_3.jpeg)

A dog is standing on a hardwood floor.

![](_page_44_Picture_5.jpeg)

A little girl sitting on a bed with a teddy bear.

A group of <u>people</u> sitting on a boat in the water.

## Image captioning with attention

![](_page_44_Picture_10.jpeg)

A <u>stop</u> sign is on a road with a mountain in the background.

![](_page_44_Picture_12.jpeg)

![](_page_44_Picture_13.jpeg)

A giraffe standing in a forest with trees in the background.

![](_page_44_Picture_16.jpeg)

## Multimodality

#### Masked Language Modeling

## I really enjoyed the \_\_\_\_\_ we watched on Saturday!

![](_page_45_Figure_4.jpeg)

(b) Multi-modal alignment prediction

![](_page_45_Picture_7.jpeg)

## Multimodality

![](_page_46_Picture_1.jpeg)

Using natural language training to improve computer vision

![](_page_46_Figure_3.jpeg)

![](_page_46_Picture_6.jpeg)

#### Learning to generate images from natural language descriptions

TEXT PROMPT

an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

![](_page_46_Picture_11.jpeg)

Edit prompt or view more images↓

#### TEXT PROMPT

an armchair in the shape of an avocado....

AI-GENERATED IMAGES

![](_page_46_Picture_16.jpeg)

Edit prompt or view more images↓

https://openai.com/blog/dall-e/

![](_page_46_Picture_19.jpeg)

#### Thanks for a great semester!