Deep Learning For NLP
Past, Current, and Future

Clara Vania
Amazon Alexa AI

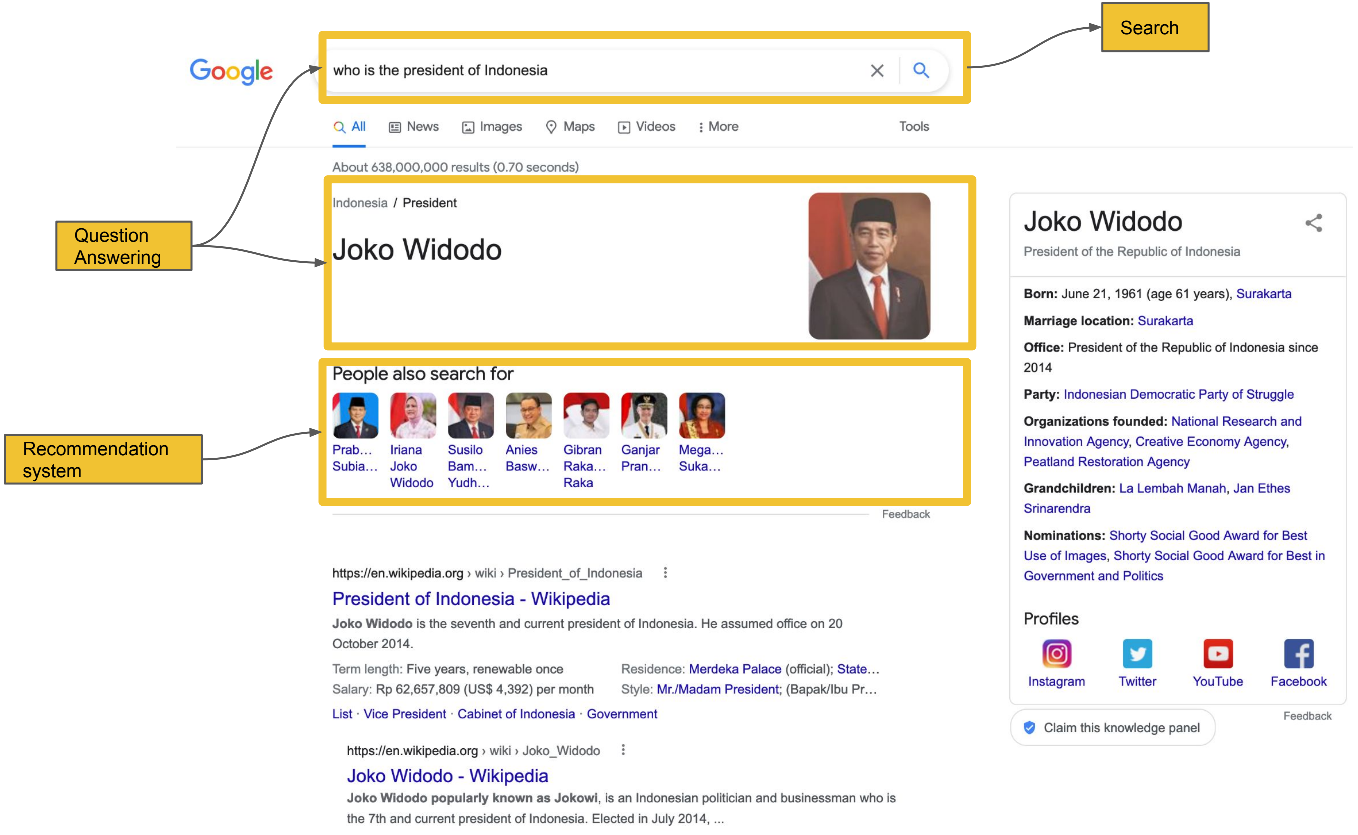
https://claravania.github.io/

Natural Language Processing (NLP)

A field at the intersection of *computer science, linguistics, artificial intelligence*, and many more.

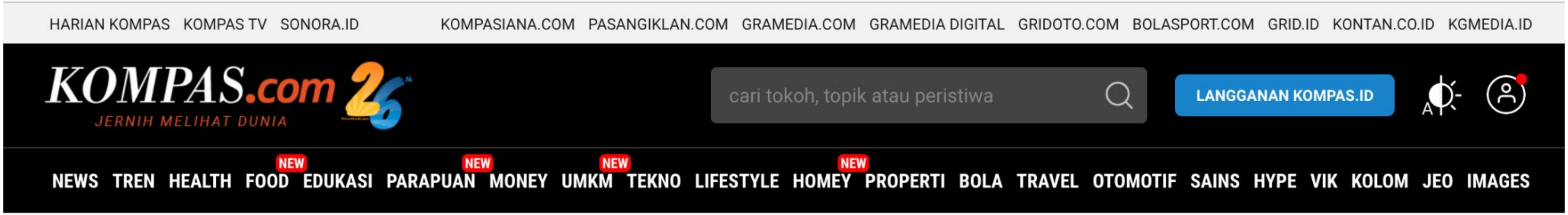
Goal: To build a system that can understand human languages.

Applications

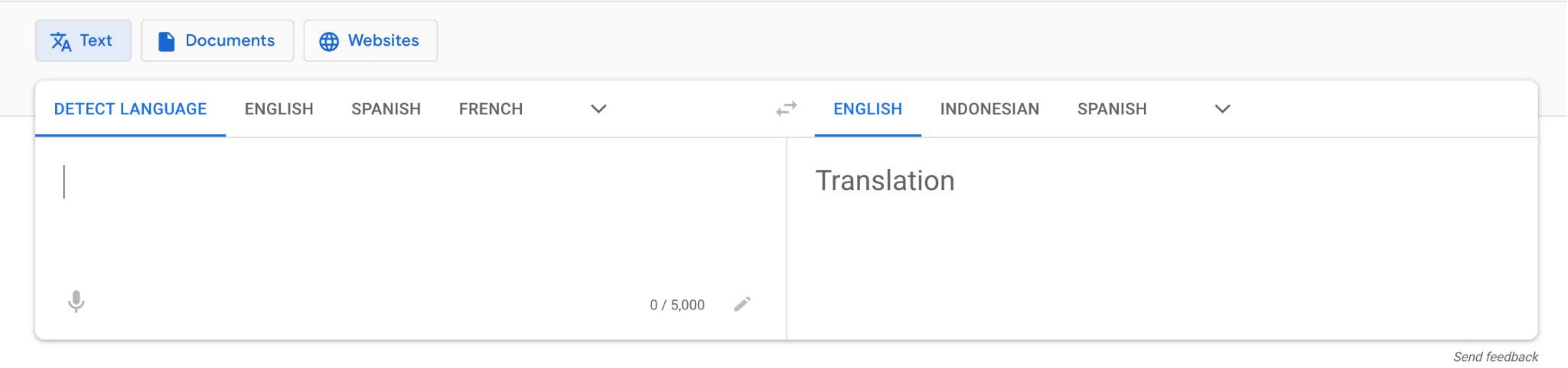


Applications

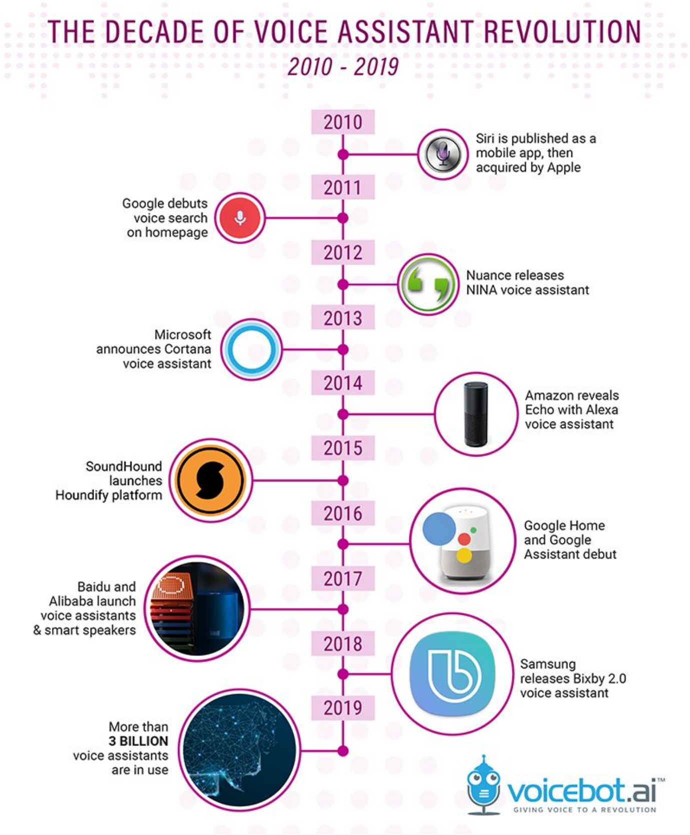
Text classification



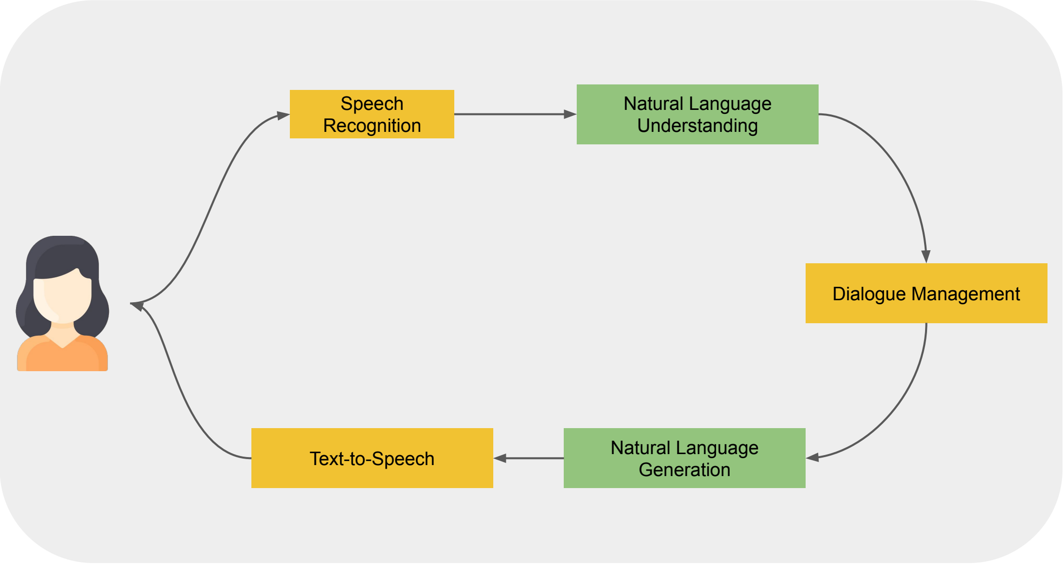
Translation



Applications



Spoken Language Understanding



Why NLP is hard?

Ambiguity: a word or a sentence can have multiple meanings.

Lexical (word):

“beruang”, “genting”, “orang tua”, “tahu”, “hati”, etc.

Sentence-level:

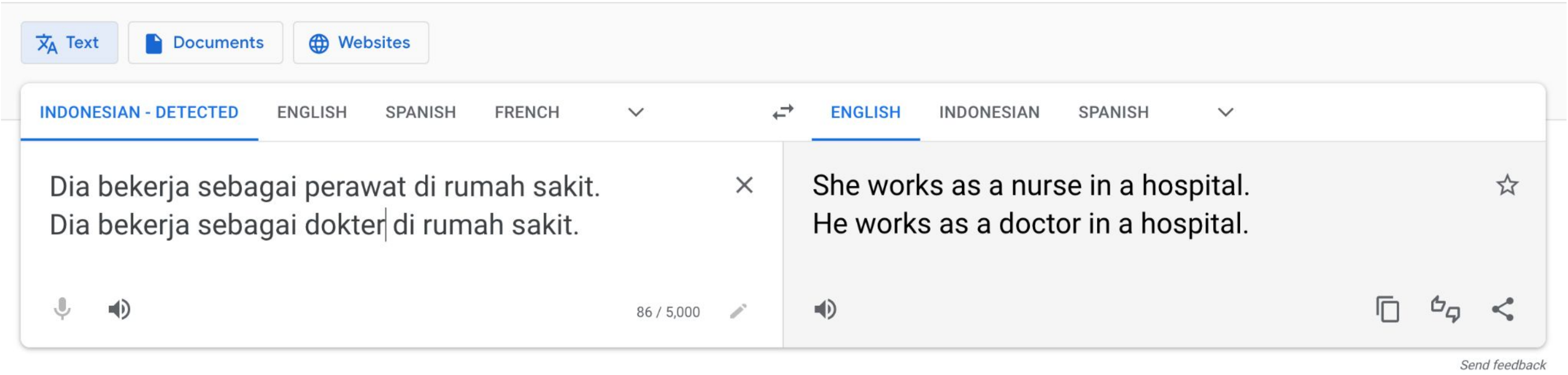
- “Saya makan nasi kemarin.”
- “Anak dokter yang baru masuk itu sering datang kemari.”
- “*I saw a man on the hill with a telescope.*”

Why NLP is hard?

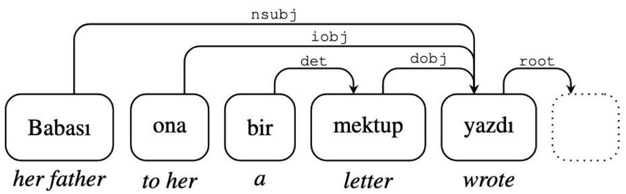
Variability: the same meaning can be expressed in multiple ways.

- “Saya berkunjung ke rumah teman saya kemarin.”
- “Kemarin saya berkunjung ke rumah teman saya.”
- “Saya pergi ke rumah teman saya kemarin.”

… also across languages



Why NLP is hard?



(1) Babası yazdı bir mektup ona (SVOIO)
(2) Yazdı babası ona bir mektup (VSIOO)
(3) Bir mektup yazdı babası ona (OVSIO)
(4) Ona bir mektup yazdı babası (IOOVS)

(Sahin and Steedman, 2019)

Turkish	English
Muvaffak	Successful
Muvaffak**iyet**	Success ('successful**ness**')
Muvaffakiyet**siz**	**Un**successful ('**without** success')
Muvaffakiyetsiz**leş***(-mek)*	*(To)* **become** unsuccessful
Muvaffakiyetsizleş**tir***(-mek)*	*(To)* **make one** unsuccessful
Muvaffakiyetsizleştiri**ci**	**Maker of** unsuccessful ones
Muvaffakiyetsizleştirici**leş***(-mek)*	*(To)* **become** a maker of unsuccessful ones
Muvaffakiyetsizleştiricileş**tir***(-mek)*	*(To)* **make one** a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiri**ver***(-mek)*	*(To)* **easily/quickly** make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştirivere**bil***(-mek)*	*(To)* **be able to make** one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştirivere**meye**bil*(-mek)*	**Not** *(to)* be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebil**ecek**	**(He/she who) will** not be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebilecek**ler**	**Those** who will not be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebilecekleri**miz**	Those who **we** will not be able to make easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimiz**den**	**Among/From** those **whom** we will not be able to easily/quickly make a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizden**miş**	**(He/she) happens to be have been** from among those whom we will not be able to easily/quickly make a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizdenmiş**siniz**	**You** happen to have been from among those whom we will not be able to easily/quickly make a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizdenmişsiniz**cesine**	**As though** you happen to have been from among those whom we will not be able to easily/quickly make a maker of unsuccessful ones

https://en.wikipedia.org/wiki/Longest_word_in_Turkish

Text is naturally sequential

- A word is a sequence of characters.
- A sentence is a sequence of words.
- A document is a sequence of sentences.

How should we model language?

Language Modeling (LM)

A central task in NLP:

- Machine translation
- Summarization
- Spell checker
- Dialogue systems
- …

“Saya harus belajar untuk mempersiapkan ujian ______________”

esok

akhir

bahasa

Language Modeling (LM)

The task of assigning the probability of a sentence.

Let $w = w_0, \ldots, w_T$ be a sequence of words in a sentence. A language model computes the probability of w as:

$$P(w) = \prod_{t=0}^{T} P(w_t \mid w_0, \ldots, w_{t-1})$$

The probability of a sentence is a product of probabilities of individual words, each conditioned on the history of previous words in the sequence.

Feedforward Neural Network LM (Bengio et al., 2003)

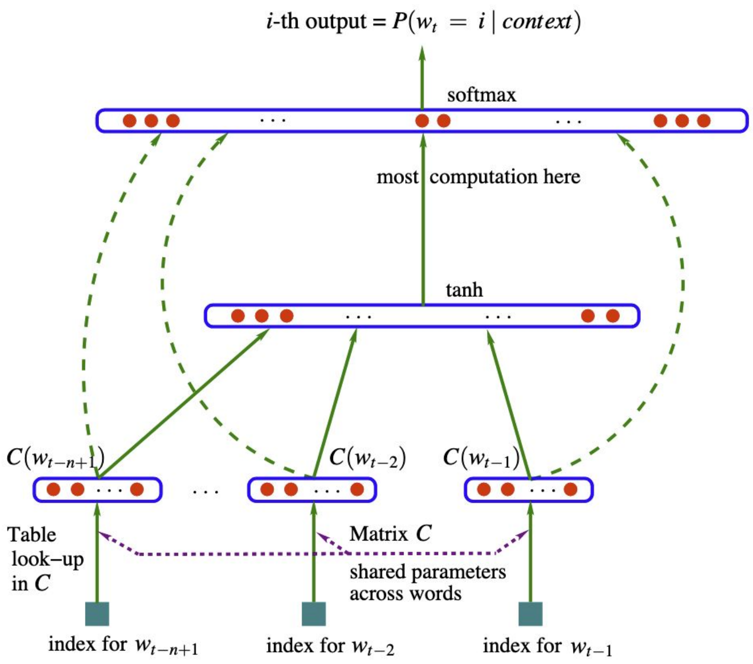


Figure 1: Neural architecture: $f(i, w_{t-1}, \cdots, w_{t-n+1}) = g(i, C(w_{t-1}), \cdots, C(w_{t-n+1}))$ where g is the neural network and $C(i)$ is the i-th word feature vector.

word2vec (Mikolov et al., 2013)

Approximate softmax:

- **Negative sampling**

  Only select a small number of "negatives" to update parameters.

- **Hierarchical softmax layers**

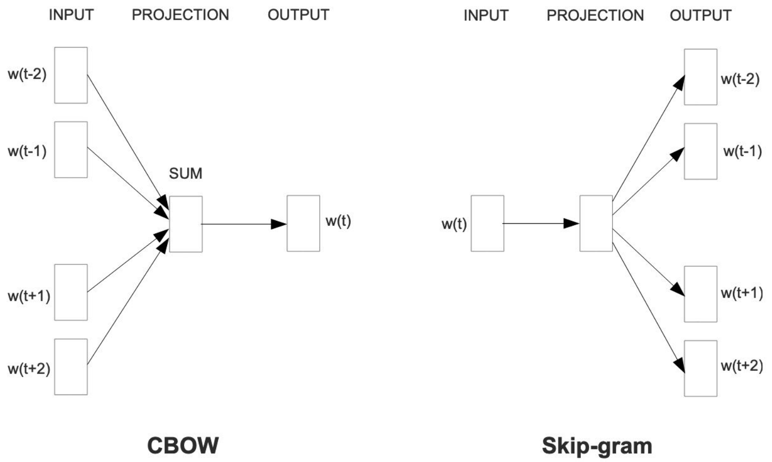


Figure 1: New model architectures. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

Recurrent Neural Network LM (Mikolov et al., 2010)

Use **infinite amount of context**. At each time step *t*, an RNN computes the following:

hidden states

embedding of word w_t

$$\mathbf{h}_t = g(\mathbf{U}^T \cdot \mathbf{w}_t + \mathbf{H}^T \cdot \mathbf{h}_{t-1})$$

$$\hat{w}_{t+1} \sim \text{softmax}(\mathbf{V}^T \cdot \mathbf{h}_t)$$

Variants of RNN: LSTM, GRU

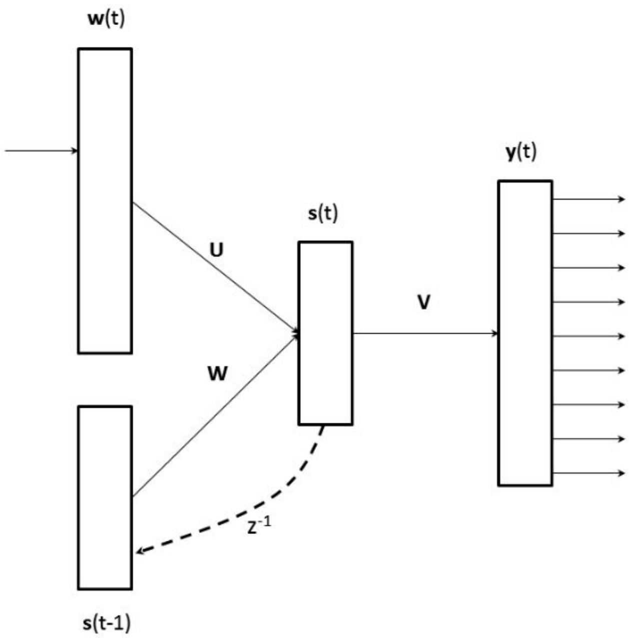


Figure 1: Recurrent Neural Network Language Model.

Sequence-to-Sequence Model

Commonly used for NLP task that generate text, e.g., machine translation or summarization.

- **Encoder:** Transform raw input to a hidden representation
- **Decoder:** Generate output from a hidden representation

Attention mechanism:

Give different weights ("attention") to different inputs

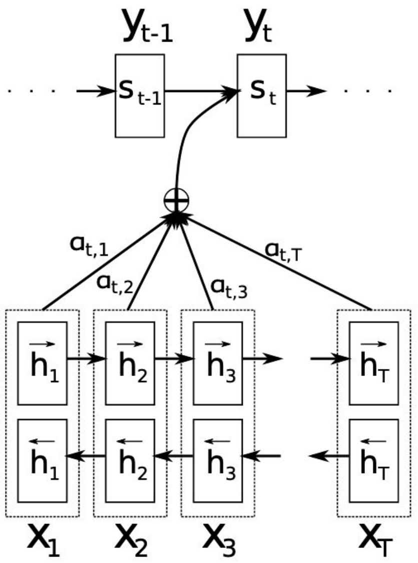


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Bahdanau et al., 2014

Transformer Model

- “Attention is all you need” ([Vaswani, et al., 2017](#))
- **Non-recurrent** encoder-decoder model
  - Long-distance context has “equal opportunity”
  - Allows parallelization
- Components:
  - Multi-headed self attention
  - Feed-forward layers
  - Layer norm and residuals
  - Positional encoding
- More complete explanations:
  - https://nlp.seas.harvard.edu/2018/04/03/attention.html

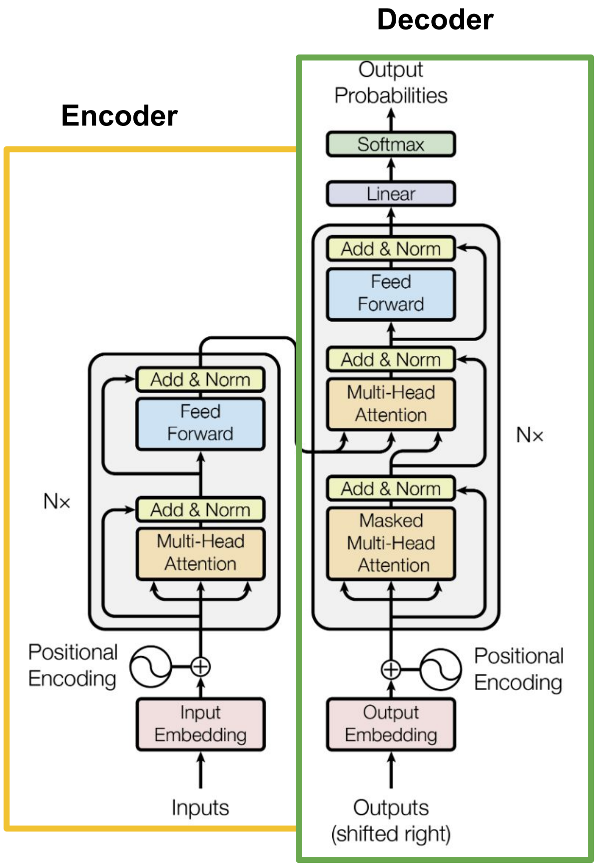


Figure 1: The Transformer - model architecture.

Attention Visualizations

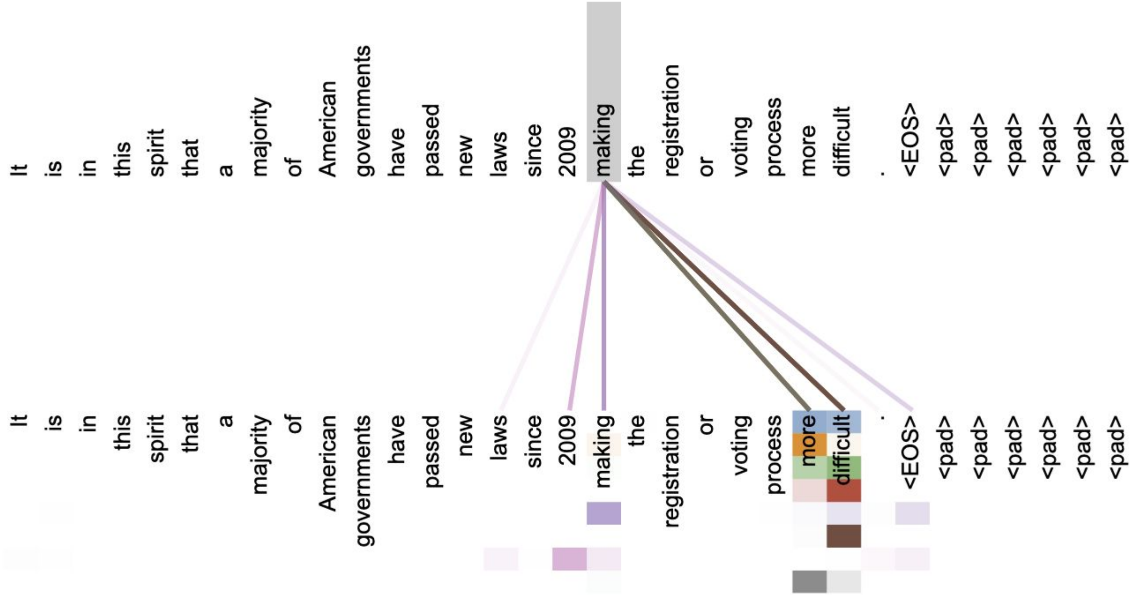


Figure 3: An example of the attention mechanism following long-distance dependencies in the encoder self-attention in layer 5 of 6. Many of the attention heads attend to a distant dependency of the verb 'making', completing the phrase 'making...more difficult'. Attentions here shown only for the word 'making'. Different colors represent different heads. Best viewed in color.

Bidirectional Encoder Representations from Transformers (BERT; Devlin et al., 2018)

Language *understanding* is bidirectional (requires a full context)

Pre-training task #1: Masked LM

Mask out *k%* (k=15) of the input words and train a bidirectional encoder to predict the mask words.

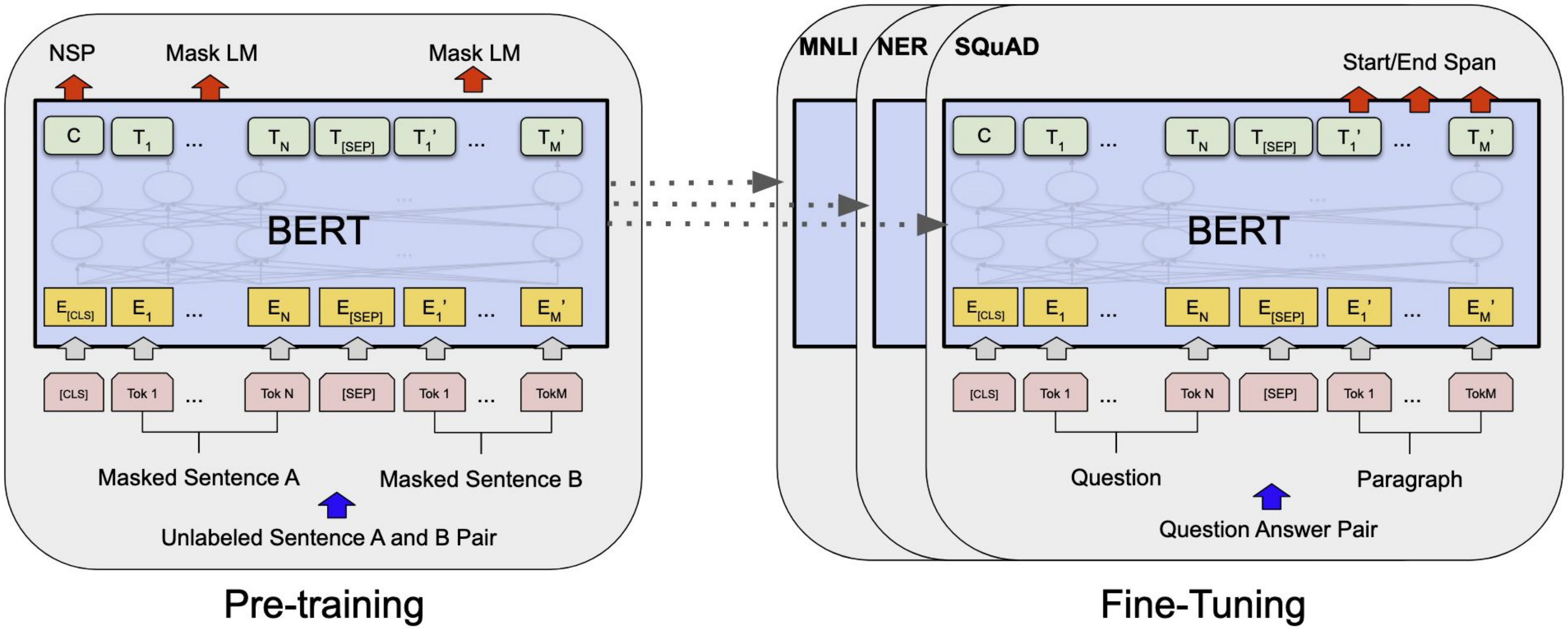
The dog wants to go to `[MASK]` to `[MASK]` his friends .

$\downarrow$ London $\quad\quad$ $\downarrow$ meet

Pre-training task #2: Next Sentence Prediction

Predict whether sentence B is the actual sentence that follows sentence A.

Bidirectional Encoder Representations from Transformers (BERT; Devlin et al., 2018)



Generative Pretraining Transformer (GPT; Radford et al., 2018)

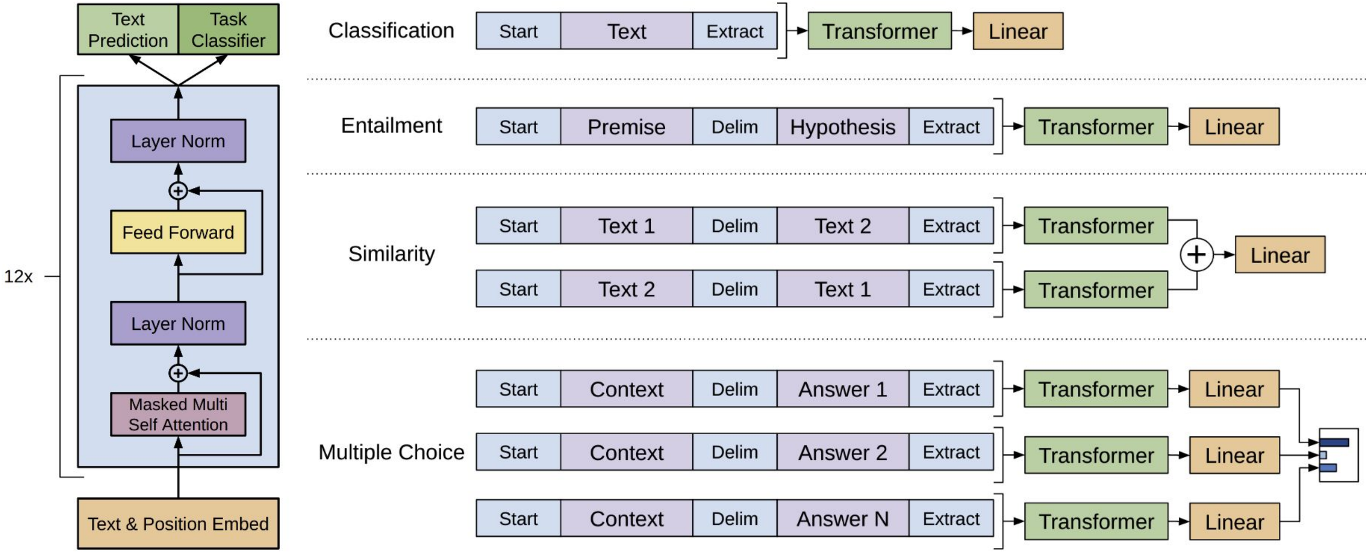
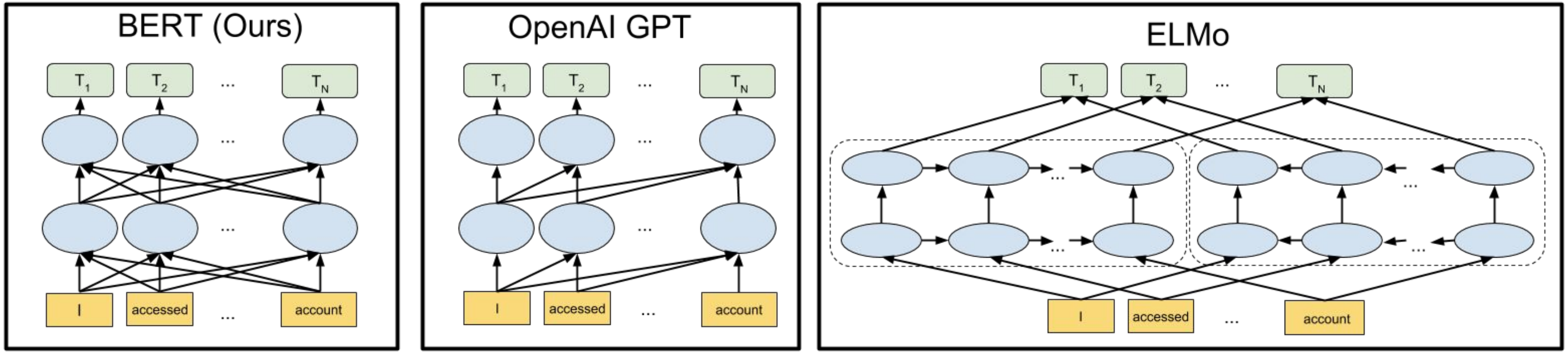
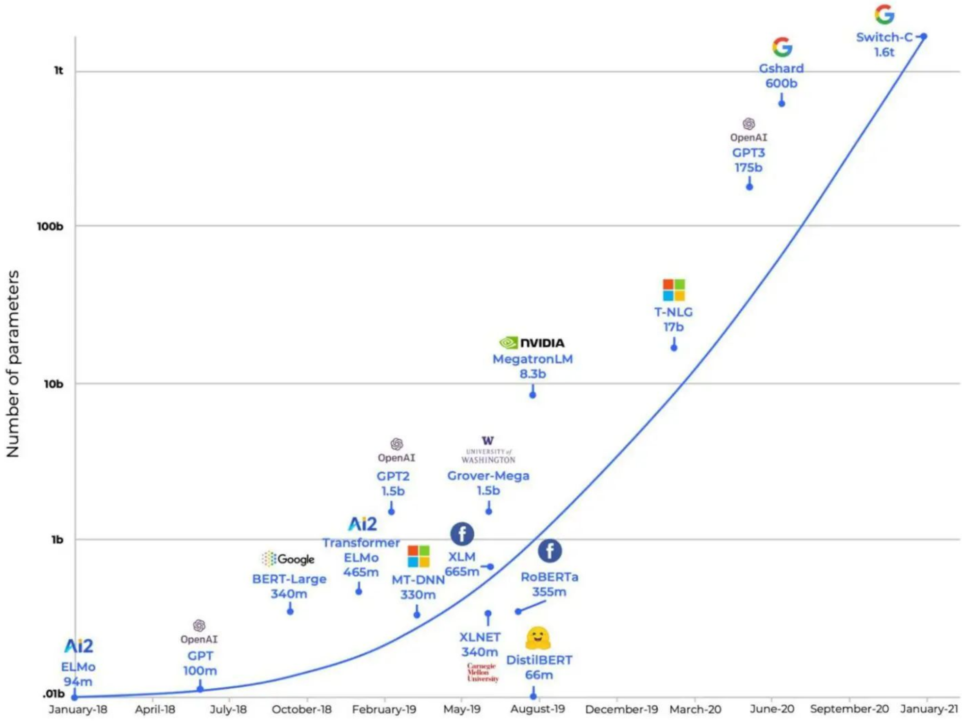


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Language Modeling with Transformer



https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html



https://deci.ai/blog/sota-dnns-overview/

Increasing number of parameters and model size improve performance, but also increase latency and make model deployment difficult.

Transfer Learning with Large Language Models (LLMs)

“Storing knowledge gained solving one problem and applying it to a different but related problem” – Ruder, 2019.

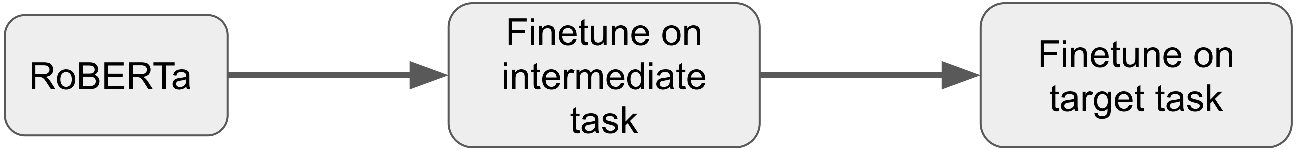
Since LLMs were trained on massive amount of data, we can utilize their knowledge for specific *target* tasks.

Especially useful for target tasks where we have limited or zero labeled data.

Intermediate-Task Transfer (STILTS; Phang et al., 2018)

Idea:

1. Pretrain a model on unlabeled data (BERT, RoBERTa, etc.)
2. Finetune the model on a large labeled *intermediate* dataset
3. Finetune it again on a smaller *target* labeled dataset



What kind of tasks make good intermediate tasks?

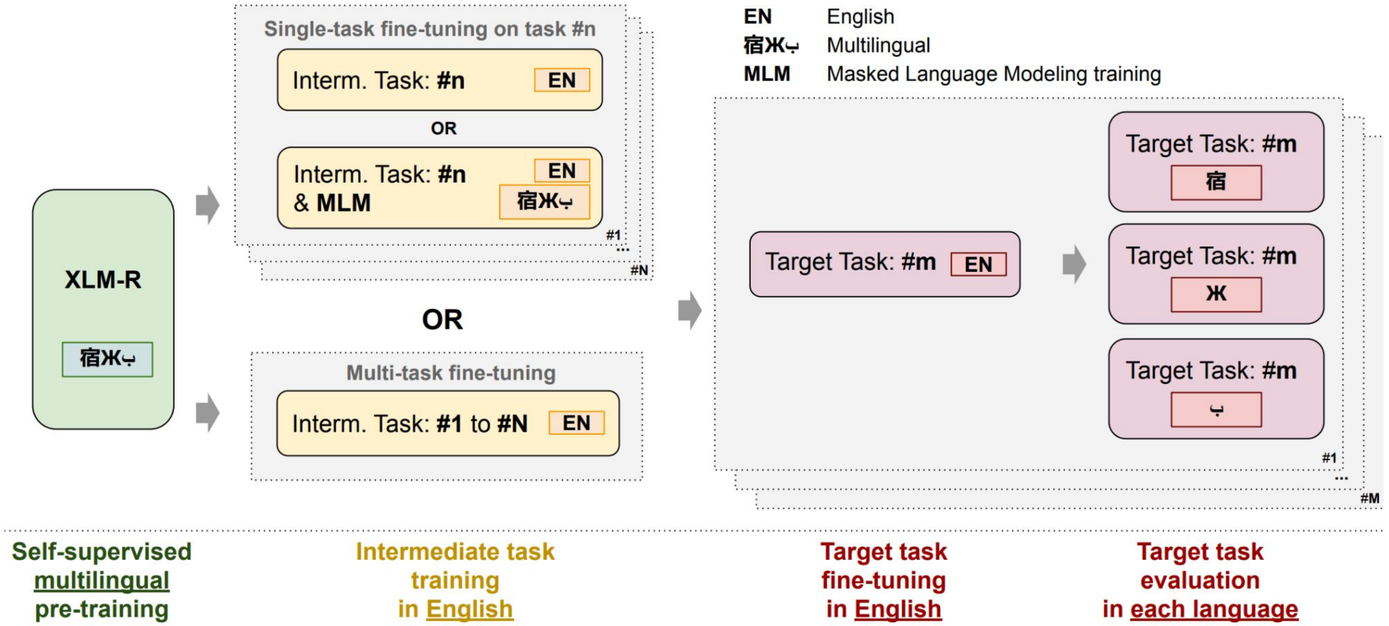
A: The commonsense-related tasks, e.g., CommonsenseQA, CosmosQA, and HellaSwag

Intermediate Tasks

Target	QAMR	CSenseQA	SciTail	CosmosQA	SocialIQA	CCG	HellaSwag	QA-SRL	SST-2	QQP	MNLI	Baseline Performance
CB	-4.0	-0.4	-6.2	-0.4	-21.7	-12.2	-3.1	-7.2	-1.2	-31.0	-0.4	99.1
COPA	-4.0	8.7	4.3	6.0	-3.7	-20.7	6.7	-3.7	-2.0	0.7	-0.7	86.0
WSC	-0.3	0.0	1.3	2.9	-4.8	-3.2	3.6	4.8	2.6	-3.8	0.3	67.3
RTE	0.6	3.4	3.4	5.1	-4.3	-18.2	4.8	1.1	2.6	-2.4	3.1	83.5
MultiRC	2.4	7.9	2.6	10.1	-10.6	-8.1	6.8	2.6	1.1	-4.2	6.5	47.4
WiC	-1.3	0.1	2.5	1.7	-2.0	-1.1	0.1	2.1	-6.4	1.4	0.9	70.5
BoolQ	-0.1	0.9	0.1	1.1	-2.8	-10.6	0.7	0.0	0.9	-4.2	1.4	86.6
CSenseQA	-4.7	-1.6	-2.6	0.1	-7.8	-12.0	0.4	-5.1	-0.9	-7.6	-2.6	74.0
CosmosQA	-2.5	-0.1	-2.1	-0.4	-9.1	-6.9	-0.0	-3.0	-0.0	-8.4	-0.5	81.9
ReCoRD	-4.0	-0.0	-1.5	-0.1	-12.4	-6.1	0.2	-4.7	-0.5	-11.9	-1.6	86.0
Avg. Target	-1.8	1.9	0.2	2.6	-7.9	-9.9	2.0	-1.3	-0.4	-7.1	0.7	78.2

Pruksachatkun et al., ACL 2020

English Intermediate-Task Training for Zero-Shot Cross Lingual Transfer (Phang et al., 2020)



English Intermediate-Task Training for Zero-Shot Cross Lingual Transfer (Phang et al., 2020)

Multi-task on all intermediate tasks obtains best overall average performance

SQuADv1.1 improves performance on QA tasks, XQuAD and MLQA.

		Target tasks									
		XNLI	PAWS-X	POS	NER	XQuAD	MLQA	TyDiQA	BUCC	Tatoeba	Avg.
	Metric	*acc.*	*acc.*	*F1*	*F1*	*F1 / EM*	*F1 / EM*	*F1 / EM*	*F1*	*acc.*	–
	# langs.	15	7	33	40	11	7	9	5	37	–
	XLM-R	**80.1**	86.5	75.7	62.8	76.1 / 60.0	70.1 / 51.5	65.6 / 48.2	71.5	31.0	67.2
Without MLM	**ANLI+**	- 0.8	- 0.0	- 1.4	- 3.5	- 1.1 / - 0.5	- 0.6 / - 0.8	- 0.6 / - 3.0	+19.9	+48.2	+ 6.6
Without MLM	**MNLI**	- 1.2	+ 1.4	- 0.7	+ 0.5	- 0.3 / - 0.1	+ 0.2 / + 0.2	- 1.0 / - 1.6	+20.0	+48.8	+ 7.5
Without MLM	**QQP**	- 4.4	- 4.8	- 6.5	-45.4	- 3.8 / - 3.8	- 3.9 / - 4.4	-11.1 / -10.2	+17.1	+49.5	- 1.5
Without MLM	**SQuADv1.1**	- 1.9	+ 1.2	- 0.8	- 0.4	+ 1.8 / + 2.5	+ 2.2 / + 2.6	+ 9.7 / +10.8	+18.9	+41.3	+ 8.1
Without MLM	**SQuADv2**	- 1.6	+ 1.9	- 1.1	+ 0.8	- 0.5 / + 0.7	- 0.4 / + 0.1	**+10.4 / +11.3**	+19.3	+43.4	+ 8.2
Without MLM	**HellaSwag**	- 7.1	+ 1.8	- 0.7	+ 1.6	- 0.0 / + 0.5	- 0.1 / + 0.2	- 0.0 / - 1.0	+20.3	+47.6	+ 7.0
Without MLM	**CCG**	- 2.6	- 3.4	- 2.0	- 1.5	- 1.5 / - 1.3	- 1.6 / - 1.5	- 2.8 / - 6.2	+11.7	+41.9	+ 4.1
Without MLM	**CosmosQA**	- 2.1	- 0.3	- 1.4	- 1.5	- 0.9 / - 1.3	- 1.5 / - 2.0	+ 0.5 / - 0.6	+19.2	+43.9	+ 6.1
Without MLM	**CSQA**	- 2.9	- 2.8	- 1.7	- 1.6	- 1.0 / - 1.8	- 1.0 / - 0.6	+ 3.5 / + 2.9	+18.1	+48.6	+ 6.5
Without MLM	**Multi-task**	- 0.9	+ 1.7	- 1.0	+ 1.8	+ 0.3 / + 0.9	+ 0.2 / + 0.5	+ 5.8 / + 6.0	+19.6	+49.9	+ 8.7
With MLM	**ANLI+**	- 1.1	+ 1.4	+ 0.0	+ 0.4	- 1.9 / - 1.7	- 0.7 / - 0.6	+ 0.9 / + 0.5	+18.6	+46.2	+ 7.1
With MLM	**MNLI**	- 0.7	+ 1.6	- 1.6	+ 1.0	- 0.7 / + 0.1	+ 0.4 / + 0.8	- 1.8 / - 3.2	+17.1	+44.3	+ 6.6
With MLM	**QQP**	- 1.3	- 1.1	- 2.4	- 0.9	- 0.3 / - 0.2	+ 0.0 / + 0.2	- 1.6 / - 4.2	+14.4	+39.8	+ 5.0
With MLM	**SQuADv1.1**	- 2.6	+ 0.3	- 2.0	- 0.9	+ 0.2 / + 1.6	+ 0.1 / + 1.1	+ 8.5 / + 9.5	+16.0	+40.3	+ 6.8
With MLM	**SQuADv2**	- 1.7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible] 8.9	+15.6	+31.3	+ 6.1
With MLM	**HellaSwag**	- 3.3	+ 2.0	- 0.7	+ 0.8	- 0.8 / - 0.0	+ 0.1 / + 0.6	+ 0.3 / + 1.0	+ 6.3	+22.3	+ 3.1
With MLM	**CCG**	- 1.0	- 1.3	- 1.2	- 1.9	- 1.9 / - 2.2	- 2.1 / - 2.6	- 5.5 / - 6.2	+ 8.8	+36.1	+ 3.3
With MLM	**CosmosQA**	- 1.0	- 1.0	- 1.6	- 3.8	- 3.1 / - 3.3	- 3.7 / - 4.2	- 0.6 / - 3.2	+15.5	+42.7	+ 4.7
With MLM	**CSQA**	- 0.5	+ 0.3	- 1.0	- 0.7	- 0.9 / - 1.0	- 0.7 / - 0.6	+ 2.1 / + 0.4	+11.6	+17.2	+ 2.9
	XTREME Benchmark Scores†										
	XLM-R (Hu et al., 2020)	79.2	86.4	72.6	**65.4**	76.6 / 60.8	71.6 / 53.2	65.1 / 45.0	66.0	57.3	68.1
	XLM-R (Ours)	79.5	86.2	74.0	62.6	76.1 / 60.0	70.2 / 51.2	65.6 / 48.2	64.5	31.0	64.8
	Our Best Models‡	**80.0**	**87.9**	**74.4**	64.0	**78.7 / 63.3**	**72.4 / 53.7**	**76.0 / 59.5**	**71.9**	**81.2**	**73.5**
	Human (Hu et al., 2020)	92.8	97.5	97.0	-	91.2 / 82.3	91.2 / 82.3	90.1 / -	-	-	-

Multilingual MLM < No Multilingual MLM

Multitask Learning

Training multiple tasks together at the same time.

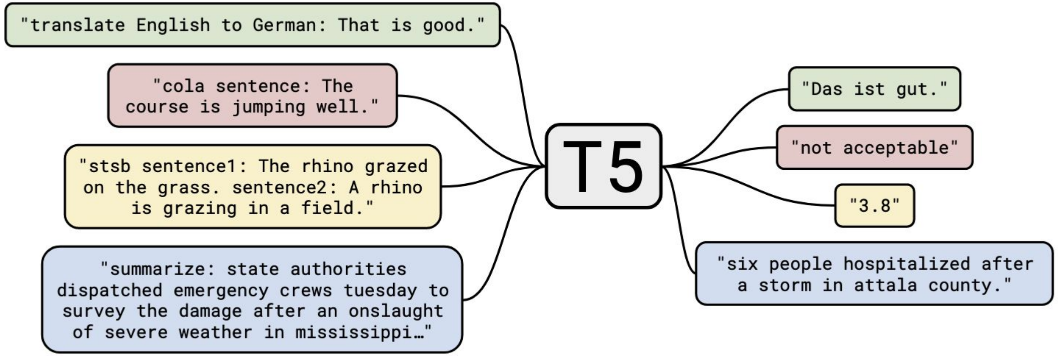


Figure 1: A diagram of our text-to-text framework. Every task we consider—including translation, question answering, and classification—is cast as feeding our model text as input and training it to generate some target text. This allows us to use the same model, loss function, hyperparameters, etc. across our diverse set of tasks. It also provides a standard testbed for the methods included in our empirical survey. "T5" refers to our model, which we dub the "**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer".

Raffel et al., 2020

(Traditional) Few-Shot Learning

N way *K* shot learning

Methods:

- Fine-tuning
- KNN
- Meta-learning

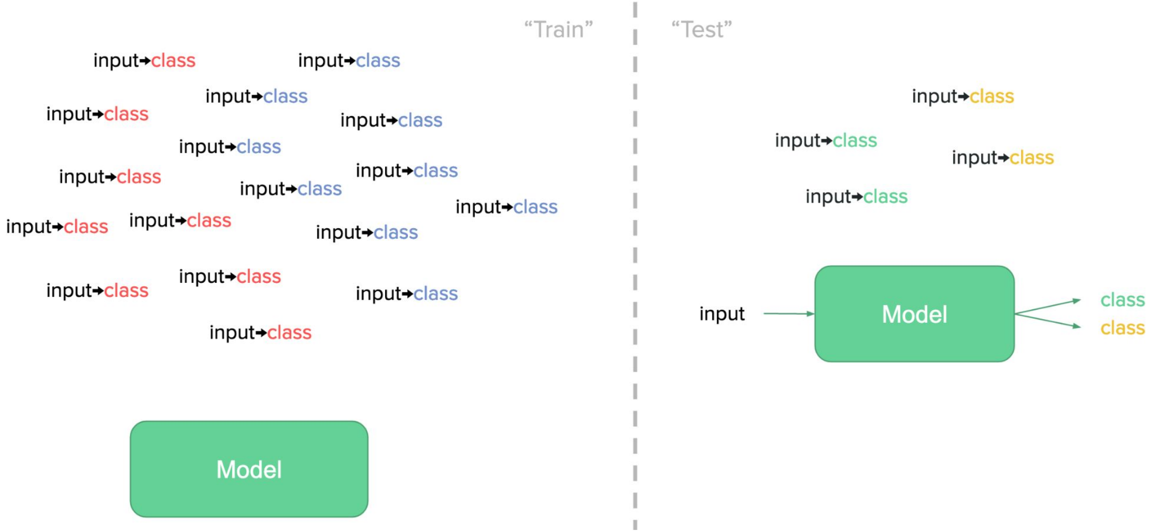


Figure from ACL 2022 tutorial "Zero- and Few-Shot NLP with Pretrained Language Models"

(Modern) Few-Shot Learning

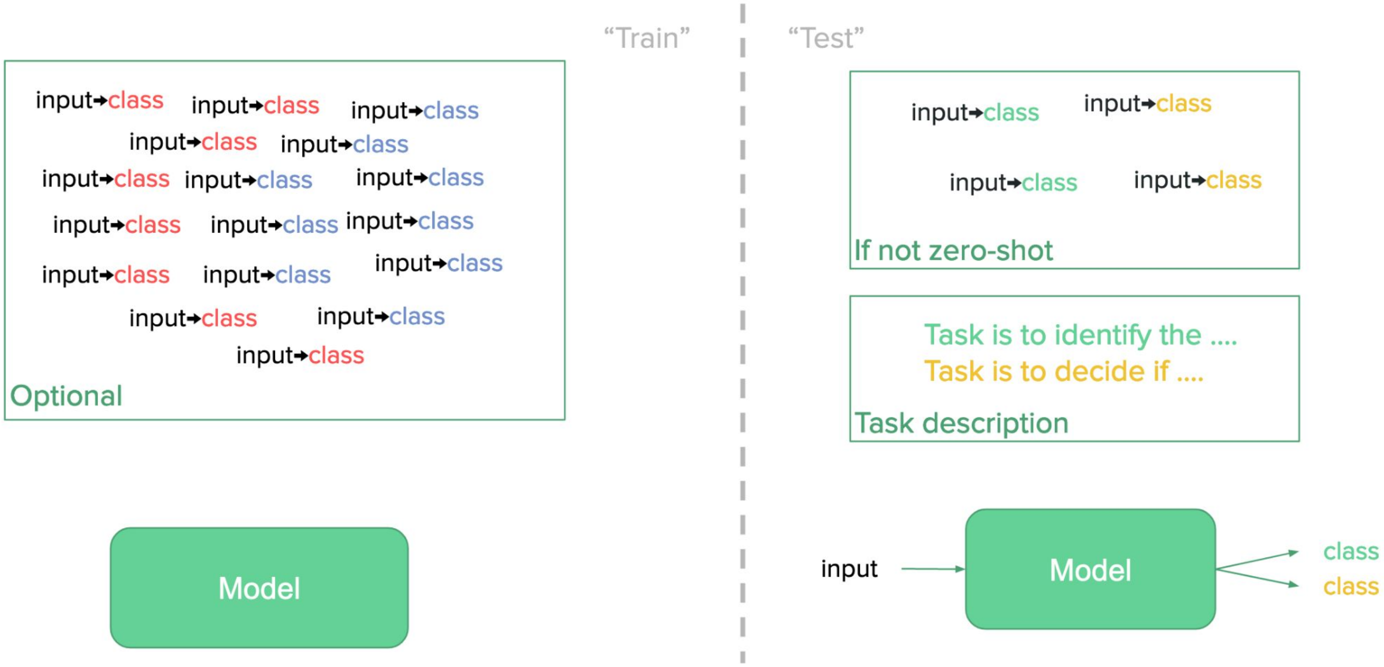
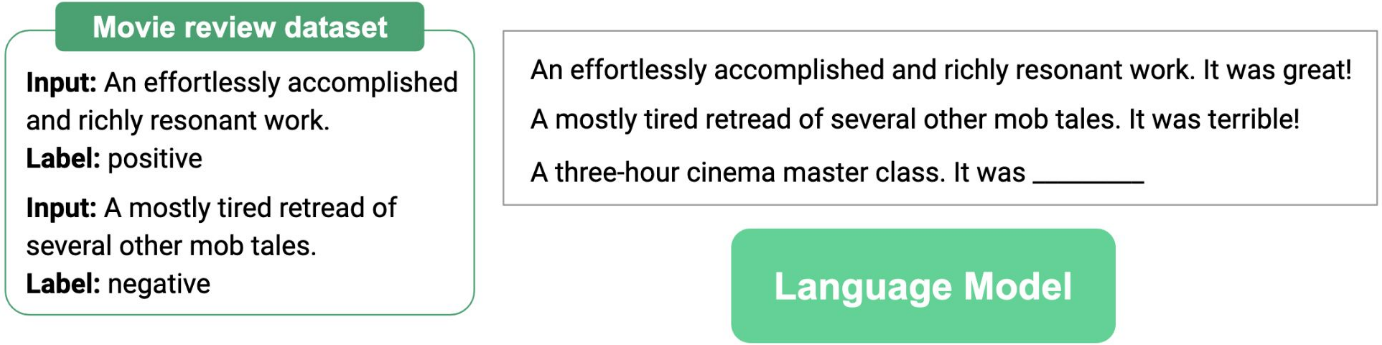


Figure from ACL 2022 tutorial "Zero- and Few-Shot NLP with Pretrained Language Models"

In-context Learning (GPT-3; Brown et al., 2020)

No task-specific parameters



P1 = P(It was **great**! | 1st train input+output \n 2nd train input+output \n A three-hour cinema master class.)

P2 = P(It was **terrible**! | 1st train input+output \n 2nd train input+output \n A three-hour cinema master class.)

P1>P2 "**positive**"

P1<P2 "**negative**"

Figure from ACL 2022 tutorial "Zero- and Few-Shot NLP with Pretrained Language Models"

In-context Learning Results

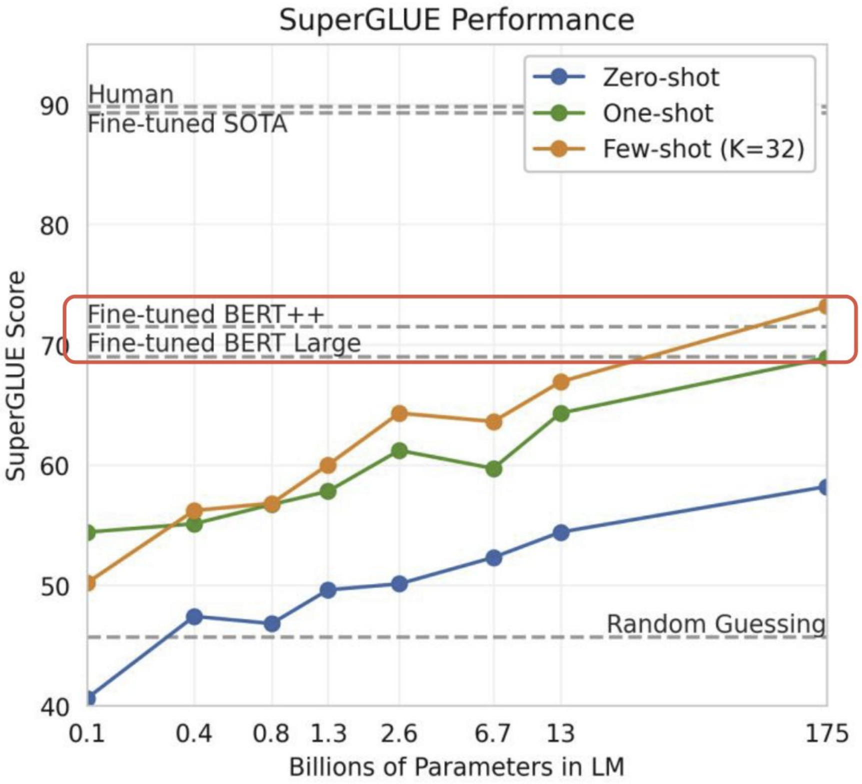


Figure from ACL 2022 tutorial "Zero- and Few-Shot NLP with Pretrained Language Models"

Prompt-based Finetuning

With gradient update, need to learn task-specific parameters.

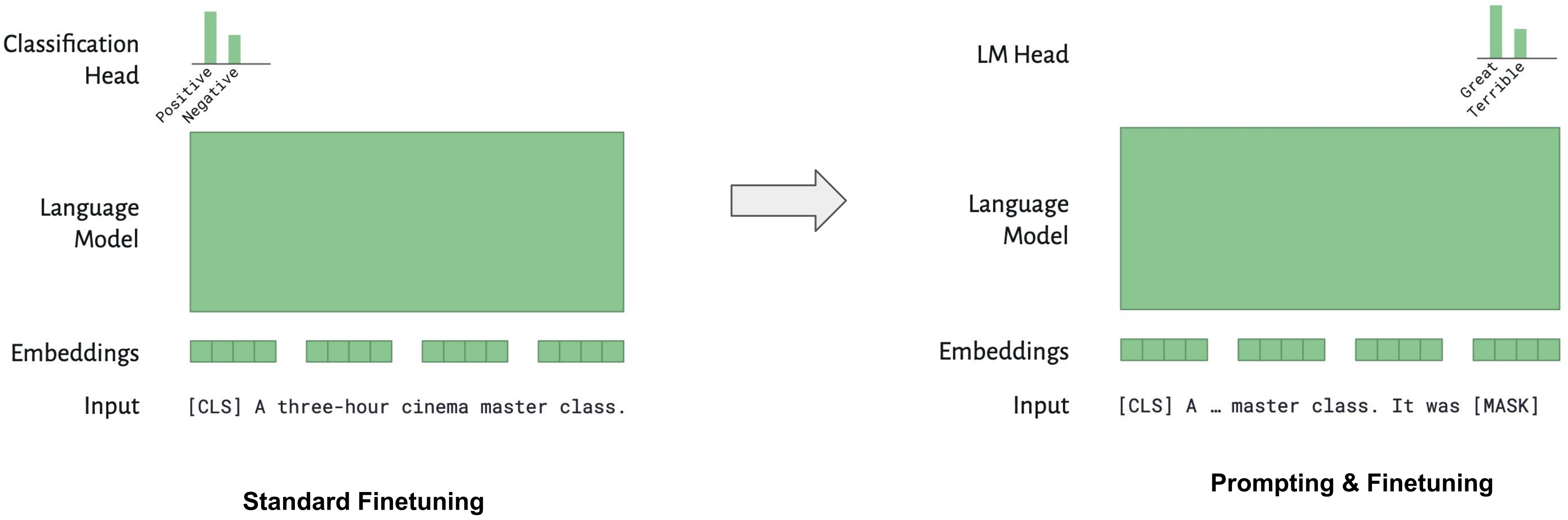


Figure from ACL 2022 tutorial "Zero- and Few-Shot NLP with Pretrained Language Models"

Challenges in Modern Few-Shot Learning

Many things to consider:

- Examples to be used
- Order of examples
- Prompt selection (design)
- …

Zero-Shot Learning

On the **same** language, reformat *target* task as *source* task.

Target task: Relation Extraction (RE)

“Joko Widodo terpilih sebagai Presiden Republik Indonesia pada Pemilihan Presiden (Pilpres) 2014”

Label: **[presiden]**

Source task: Natural Language Inference (NLI)

Premise: original sentence

Hypothesis: *“Joko Widodo adalah presiden negara Indonesia.”*

Label: **entailment**

Sainz et al., 2021

Zero-Shot Learning

Source language **different** with target language

- Use multilingual LM as pretrained model
- Finetuning on high-resource language labeled data
- Evaluate on target language data
- Use machine translation (MT) if needed

Challenges:

- Multilingual LM pretraining data might contain very little to no data in low-resource languages
- Poor performance of MT models

What’s Next?

- Lightweight, more efficient model
  - Knowledge distillation
  - Parameter-efficient model training
- Learning from Limited Labeled Data
  - Few-Shot Learning
  - Efficient Data Collection
- Multilingual
  - Benchmark datasets for non-English languages, esp. the low-resource ones
  - Pretraining methods and data selection

Thank You!