POLITECNICO DI MILANO

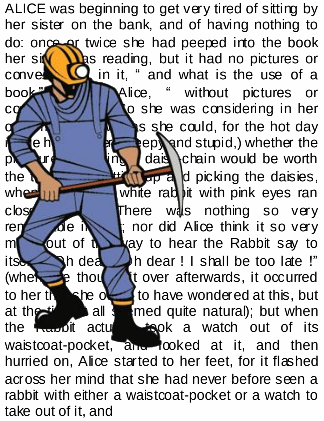
Natural Language Processing

Applications Of Transformers

Natural Language Processing

Some slide content based on textbooks:
Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition *by Daniel Jurafsky and James H. Martin*
Machine Learning and Security: **Protecting Systems with Data and Algorithms** by Clarence Chio & David Freeman

Mark Carman



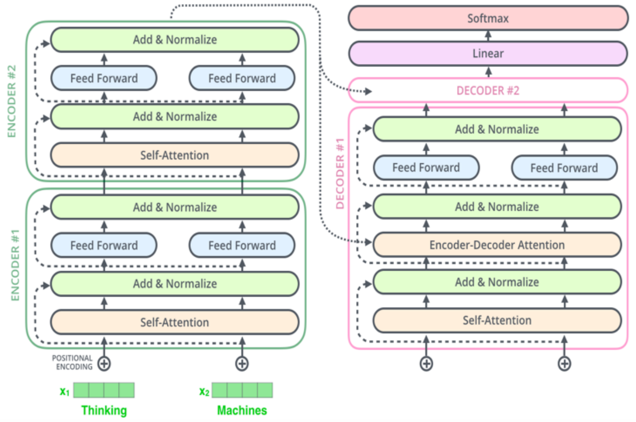
Miner Image source: https://freesvg.org/miner-1574424884

Lecture Contents:

- Fine-tuning BERT and GPT-2
- Zero-shot Learning
- Document Embeddings
- Vector Databases
- Multi-task Learning
- Multi-modal Learning

Fine-tuning Transformers to perform other Tasks

Reminder: Three Possible Architectures

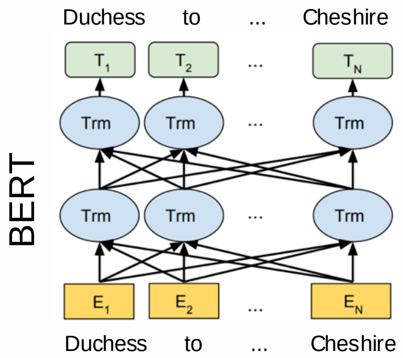


Original Transformer

- was **designed for translation**, so
- contains both **encoder** and **decoder**

BERT = **B**idirectional **E**ncoder **R**epresentations from **T**ransformers

- **encoder-only** model
- pretrained as a noisy **autoencoder** -- must **recover** potentially **masked input** at top of each column
- great for **representing text** (e.g. for building classifiers)



GPT = **G**enerative **P**retrained **T**ransformer

- **decoder-only** model
- pretrained as **autoregressive** model – must **predict next token** at top of each column
- great for **generating text**

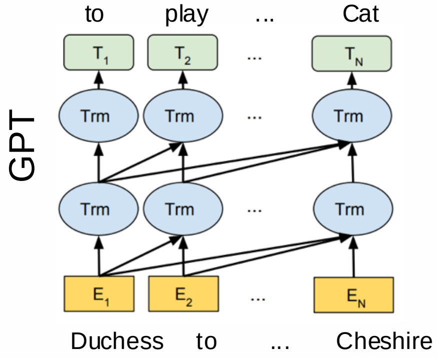


Image Source: http://jalammar.github.io/illustrated-transformer/

Fine-tuning BERT for …

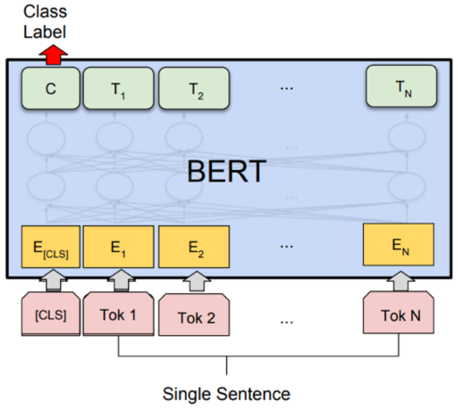
Bidirectional language models like BERT are **very flexible**!

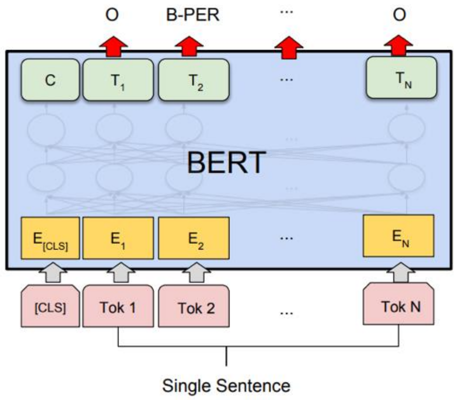
BERT usually trained to perform **text classification**
- by fine-tuning to replace “[CLS]” token by class

But can also be fine-tuned to perform **sequence labelling**
- by simply replacing the output text during fine-tuning
- with sequences of begin/inside/outside labels

Moreover, can be fine-tuned to perform **text pair classification**
- by adding a special “[SEP]” token to separate 2 pieces of text
- comparing texts is massively useful for all sorts of applications
- such as determining whether they agree or discuss same topic

BERT can even be used for question-answering tasks
- but GPT is more more appropriate model for that task





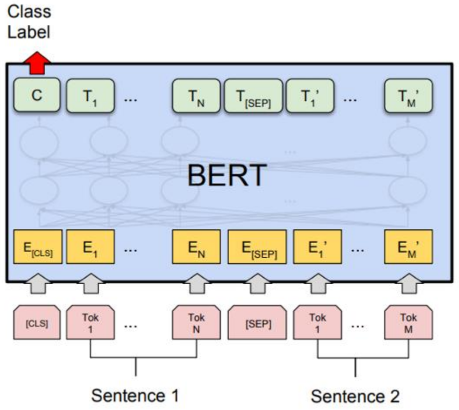


Image source: https://arxiv.org/pdf/1810.04805.pdf

Fine-tuning GPT-2 for …

GPT-2 can also be used as text encoder for classification tasks, but strength of GPT-2 is **text generation**

- so makes sense to use it for tasks such as **translation**, **summarization**, **dialog**, etc.

During **fine-tuning**

- introduce special **tokens (or text prompts)** to separate input from output
- and to **indicate type** of output desired

Fine-tuning dataset for translation

I	am	a	student	<to-fr>	je	suis	étudiant
let	them	eat	cake	<to-fr>	Qu'ils	mangent	de
good	morning	<to-fr>	Bonjour				

Fine-tuning dataset for summarization

Article #1 tokens	<summarize>	Article #1 Summary	
Article #2 tokens	<summarize>	Article #2 Summary	padding
Article #3 tokens	<summarize>	Article #3 Summary	

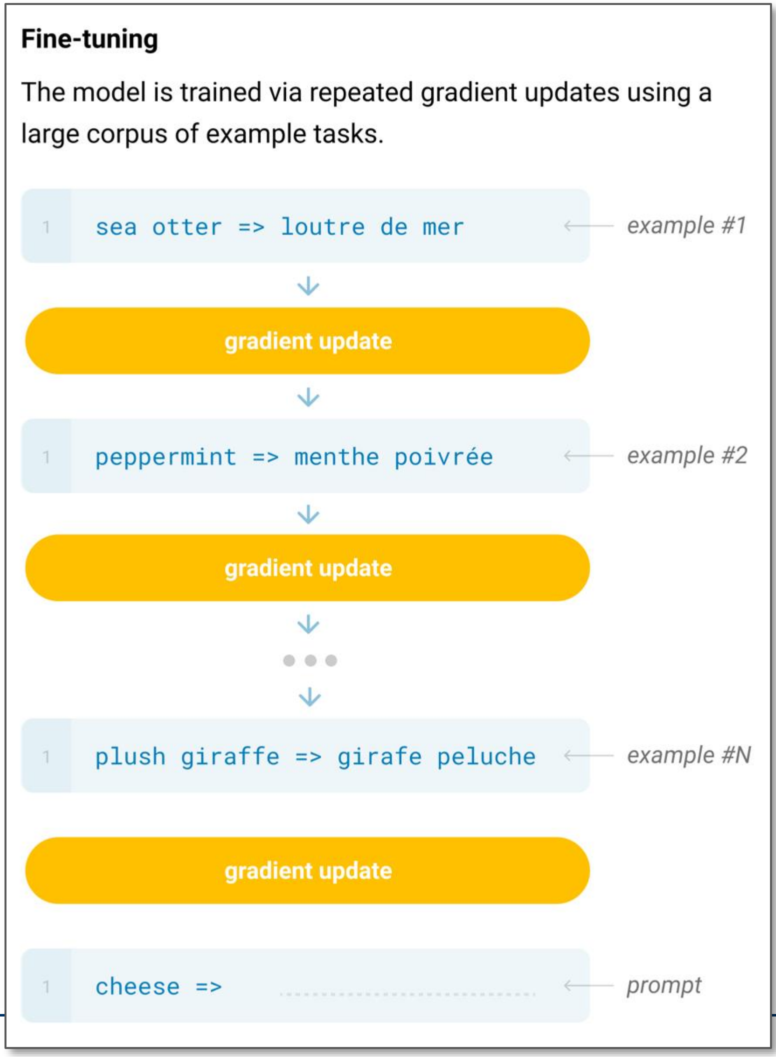
Source: http://jalammar.github.io/illustrated-gpt2/

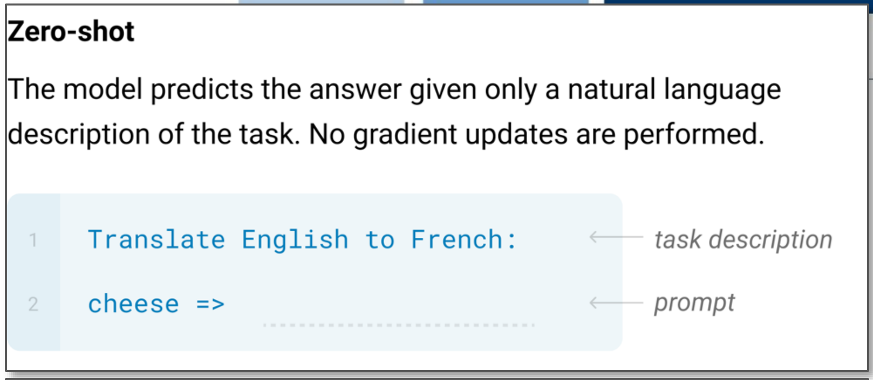
Further uses of GPT:

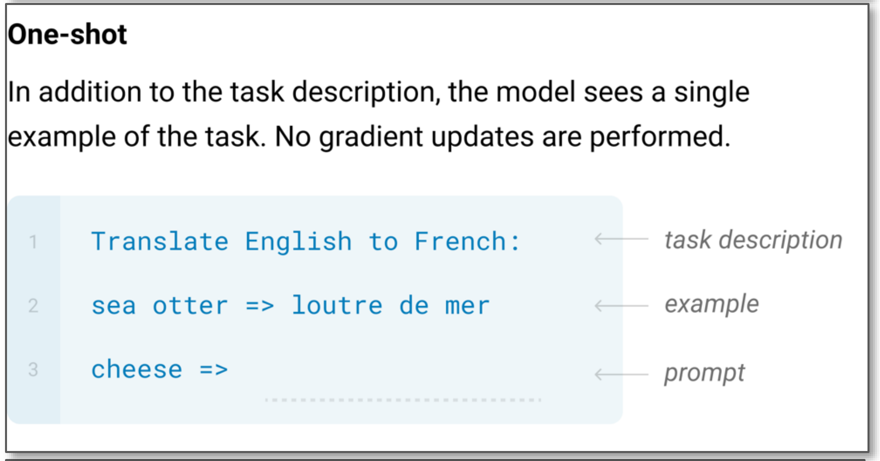
Zero, One and Few-shot Learning with Generative (GPT) models

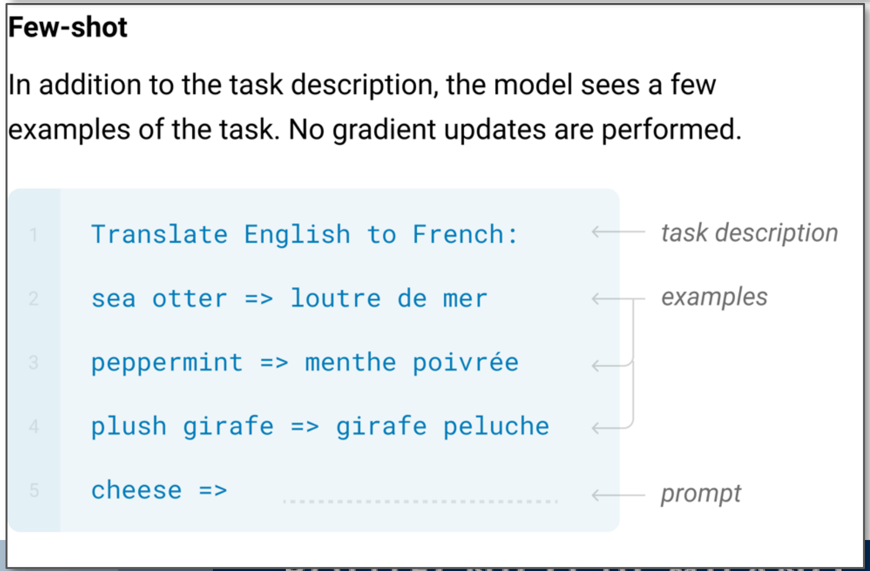
GPT without fine-tuning

Language Models are universal learners that can be used **with** or **without fine-tuning**





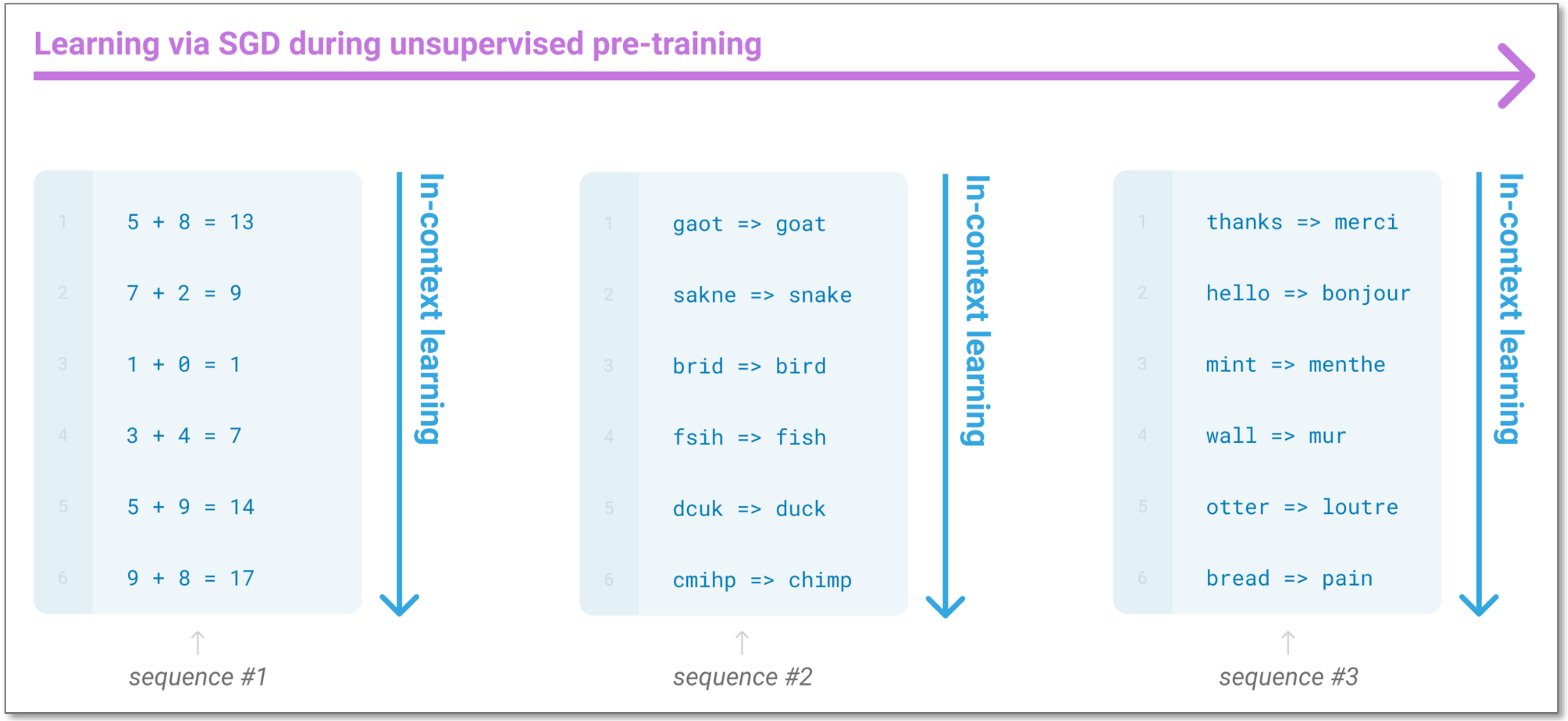




From: Brown et al. "Language Models are Few Shot Learners" https://arxiv.org/abs/2005.14165

How is few-shot learning even possible?

Model has seen **lots of examples** of few-shot learning during pretraining!



From: Brown et al. "Language Models are Few Shot Learners"
https://arxiv.org/abs/2005.14165

Many tasks handled by zero/few-shot learning

"swiss army knife" image created from:
https://www.bing.com/images/create/swiss-army-knife/1-6628d0dca2a54ff0a82b6606e63b7a62

Language models are universal learners!
Predicting text is flexible method for providing all sorts of functionality:

Translation:

- in the context, provide multiple strings of the form:
  text in source language **=** *text in target language*
- then prompt with: *sentence to translate* **=**

Question answering:

- simply prompt the model with the question, possibly formulated as a statement:
- *The height of the Eiffel Tower in metres is*

Reading comprehension:

- give text and examples of questions with answers,
- then prompt with unanswered question

Summarization:

- Provide content to be summarised and prefix response with "**tl;dr:**"

GPT-2 examples: question answering

Language model can **learn facts**, and answer questions!

- stores facts in its “**parametric knowledge**”

Confident predictions from GPT-2 were usually correct:

- although not as reliable as other forms of question answering (at the time)
- note: system **had not been trained** to do this!

Question	Generated Answer	Correct	Probability
Who wrote the book the origin of species?	Charles Darwin	✓	83.4%
Who is the founder of the ubuntu project?	Mark Shuttleworth	✓	82.0%
Who is the quarterback for the green bay packers?	Aaron Rodgers	✓	81.1%
Panda is a national animal of which country?	China	✓	76.8%
Who came up with the theory of relativity?	Albert Einstein	✓	76.4%
When was the first star wars film released?	1977	✓	71.4%
What is the most common blood type in sweden?	A	✗	70.6%
Who is regarded as the founder of psychoanalysis?	Sigmund Freud	✓	69.3%
Who took the first steps on the moon in 1969?	Neil Armstrong	✓	66.8%
Who is the largest supermarket chain in the uk?	Tesco	✓	65.3%
What is the meaning of shalom in english?	peace	✓	64.0%
Who was the author of the art of war?	Sun Tzu	✓	59.6%
Largest state in the us by land mass?	California	✗	59.2%
Green algae is an example of which type of reproduction?	parthenogenesis	✗	56.5%
Vikram samvat calender is official in which country?	India	✓	55.6%
Who is mostly responsible for writing the declaration of independence?	Thomas Jefferson	✓	53.3%
What us state forms the western boundary of montana?	Montana	✗	52.3%

Image source: “Language Models are Unsupervised Multitask Learners” by Radford et al.
https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

GPT-2 examples: reading comprehension

Context (passage and previous question/answer pairs)

The 2008 Summer Olympics torch relay was run from March 24 until August 8, 2008, prior to the 2008 Summer Olympics, with the theme of "one world, one dream". Plans for the relay were announced on April 26, 2007, in Beijing, China. The relay, also called by the organizers as the "Journey of Harmony", lasted 129 days and carried the torch 137,000 km (85,000 mi) – the longest distance of any Olympic torch relay since the tradition was started ahead of the 1936 Summer Olympics.

After being lit at the birthplace of the Olympic Games in Olympia, Greece on March 24, the torch traveled to the Panathinaiko Stadium in Athens, and then to Beijing, arriving on March 31. From Beijing, the torch was following a route passing through six continents. The torch has visited cities along the Silk Road, symbolizing ancient links between China and the rest of the world. The relay also included an ascent with the flame to the top of Mount Everest on the border of Nepal and Tibet, China from the Chinese side, which was closed specially for the event.

Q: What was the theme
A: "one world, one dream".

Q: What was the length of the race?
A: 137,000 km

Q: Was it larger than previous ones?
A: No

Q: Where did the race begin?
A: Olympia, Greece

Q: Is there anything notable about that place?
A: birthplace of Olympic Games

Q: Where did they go after?
A: Athens

Q: How many days was the race?
A: seven

Q: Did they visit any notable landmarks?
A: Panathinaiko Stadium

Q: And did they climb any mountains?
A:

Model answer: Everest

Provide source **context**:

- document, which contains answers

For **few shot** learning provide also:

- examples of questions and answers

Provide **new question**:

- and prompt model for answer

Note:

- general scheme can be used for all sorts of problems
- e.g. **fact checking**, where potential evidence supporting/refuting claim is first retrieved as context

GPT-2 examples: translation

GPT-2 even worked out of the box as a machine translation system
- was not the best translator out there ;-)
- but the system was not trained to do translation!

Moreover, it was only trained on an ENGLISH corpus
- so how could it learn to “speak” French?
- fragments of French were hidden in the training data …

English reference	GPT-2 French translation
One man explained that the free hernia surgery he’d received will allow him to work again.	Un homme expliquait que le fonctionnement de la hernia fonctionnelle qu’il avait reconnaît avant de faire, le fonctionnement de la hernia fonctionnelle que j’ai réussi, j’ai réussi.
French reference	**GPT-2 English translation**
Un homme a expliqué que l’opération gratuite qu’il avait subie pour soigner une hernie lui permettrait de travailler à nouveau.	A man told me that the operation gratuity he had been promised would not allow him to travel.

”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: ”**Mentez mentez, il en restera toujours quelque chose**,” which translates as, ”**Lie lie and something will always remain.**”

“I hate the word ‘**perfume**,”’ Burr says. ‘It’s somewhat better in French: ‘**parfum**.’

If listened carefully at 29:55, a conversation can be heard between two guys in French: “**-Comment on fait pour aller de l’autre coté? -Quel autre coté?**”, which means “**- How do you get to the other side? - What side?**”.

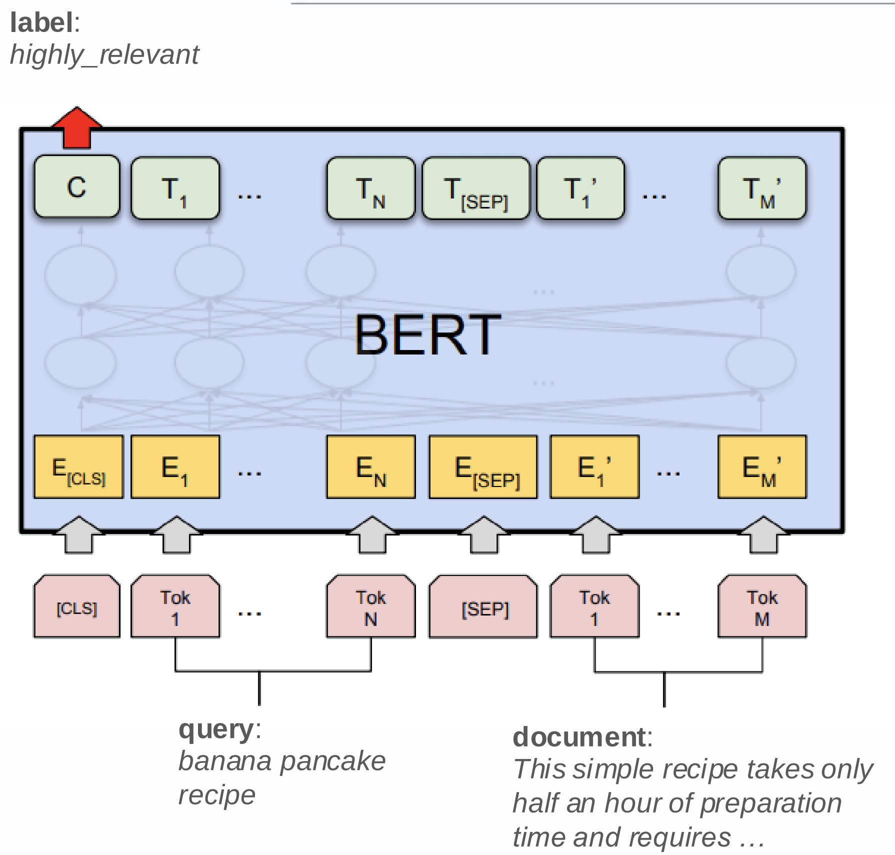
If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

“**Brevet Sans Garantie Du Gouvernement**”, translated to English: “**Patented without government warranty**”.

Further uses of BERT:

Estimating Similarity between Documents

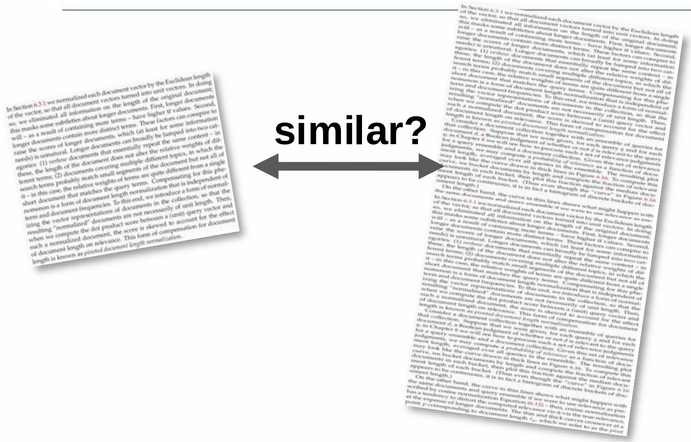
Learning to Rerank Documents



BERT-based can be used to **rerank documents** in web search

- simply fine-tune BERT model to predict the **relevance label** (e.g. “highly relevant”, ”relevant”, “not relevant”, ”spam”)
- for a set of **<query, document>** pairs
- use “[SEP]” token to separate query and document on input

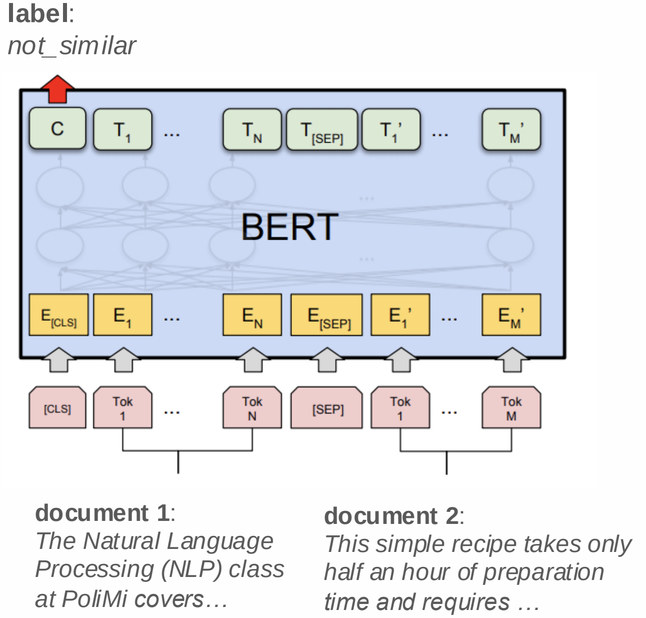
Document Similarity



What if we need to calculate similarity between documents?
- say in order to cluster them

Fine-tune BERT to estimate **semantic similarity** between documents:
- starting from pairs of **similar documents**
- and randomly chosen pairs of (probably) **dissimilar documents**
- fine-tune **pairwise classifier** to identify similar documents
- use logits (or probability of similar class label) as **similarity score**

If ground truth similarity/distance information is available
- could also fine-tune model on **regression task**
- to predict the similarity value)



Further uses of BERT:

Sentence Transformers

Problem: Computational Overhead

Using **pairwise BERT classifier** to estimate **relevance** of each document for a given query:

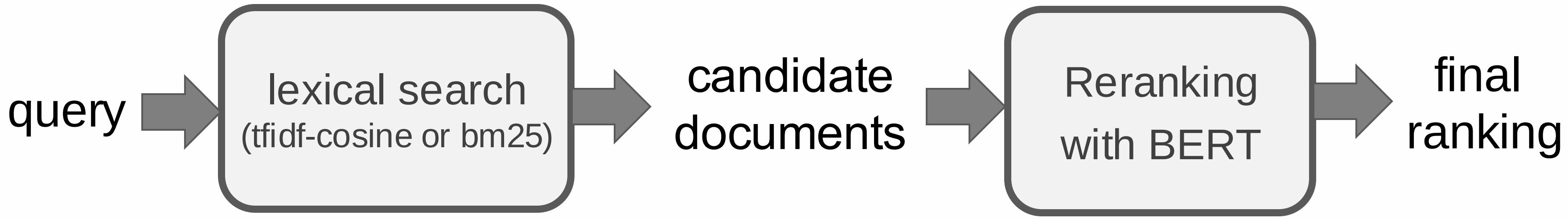
- is **very powerful** since the BERT model can:
  - leverage the **order of words** in the document and the query
  - take into account the **semantics of words** using embeddings

- but also **very costly** since BERT
  - performs many matrix multiplications during inference
  - **needs GPU** to run fast but still takes considerable time to run
  - will need to compute a score for **every document** in the collection
  - example: if it takes 1ms to compute score per document and there are a million documents, will wait over 15 minutes to run query

Can something be done to speed up the computation?

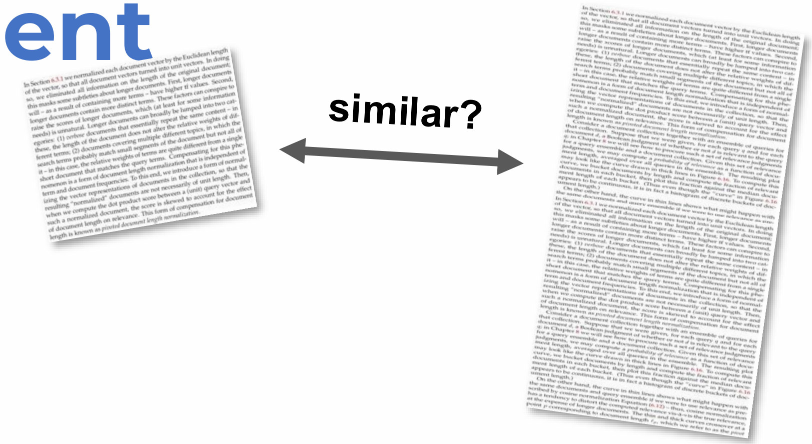
- ideally, like to perform as much **precomputation** as possible
- but can't precompute similarity until we've seen the query …

Solution 1: use lexical search to find candidates



- use lexical search engine to find candidate set of documents quickly
- use fine-tuned pairwise BERT classifier only to **rerank** candidate documents

Solution 2: pre-compute document embeddings



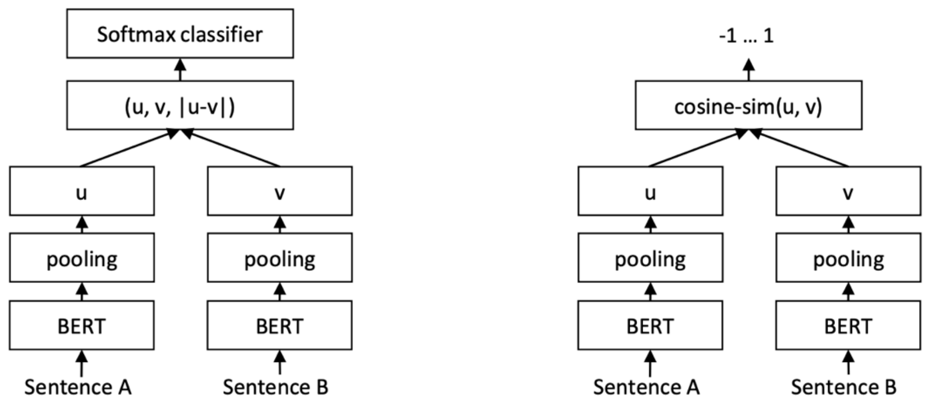
BERT can also be trained to compute embeddings for documents

- use **output** (contextual) **embedding** for **[CLS] token** in BERT as the **representation of each document**
- and **dot-product** between embeddings of different documents **as their similarity**
- train model on pairs of similar and dissimilar documents using "**contrastive loss**"
  - i.e. to produce high similarity scores for documents labelled similar and low similarity scores for documents labelled as dissimilar.
- given the context length restriction (500 tokens) might need to compare sections of the document and aggregate

Sentence BERT (SBERT)

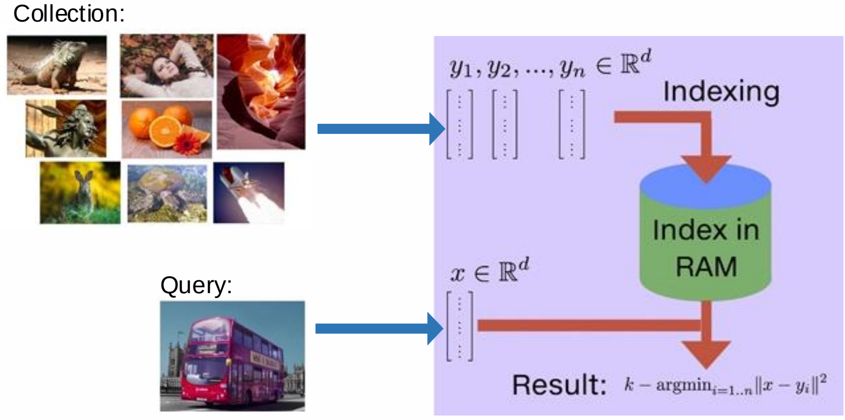
Sentence BERT (SBERT) uses:

- BERT (or RoBERTa) model to learn vector representation for entire documents
- **contrastive learning** to produce an embedding space where similar documents produce similar embeddings
  - in practice: train two BERT models (one for query and one for document)
  - use dot-product between output embeddings in [CLS] token position to represent documents
  - train model on pairs of similar and not similar documents to produce high values for similarity if documents are similar and vice versa



Vector Databases

Vector Search



FAISS applied to image embeddings

Vector Databases

- index objects (text or images) based on their embedding
- provide **fast nearest neighbor search** in embedding space
  - example FAISS (Facebook AI Similarity Search)

Nearest Neighbour Search

- finding nearest neighbours in **high-dimensions** is difficult
- since vectors are all at 90 degrees and approximately equidistant
- clever algorithms effectively **partition space** into **clusters**
  - example HNSW (Hierarchical Navigable Small Worlds)

Approximate Nearest Neighbour search

Finding Nearest Neighbours in high-dimensional data is hard!

- indexes like k-d trees can provide O(log(n)) search for nearest neighbours in low dimensional spaces, https://en.wikipedia.org/wiki/K-d_tree
- but break down in high dimensions, leading to O(n) behaviour

Approximate nearest neighbour search

- make use of Hierarchical Navigable Small Worlds (HNSW) https://en.wikipedia.org/wiki/Hierarchical_navigable_small_world
- Navigable Small World graphs:
  - nodes are connected to their nearest neighbours, allowing for quick search
- Hierarchy of layers
  - nodes are connected between layers
  - with all nodes on bottom layer and iteratively fewer nodes on higher layers

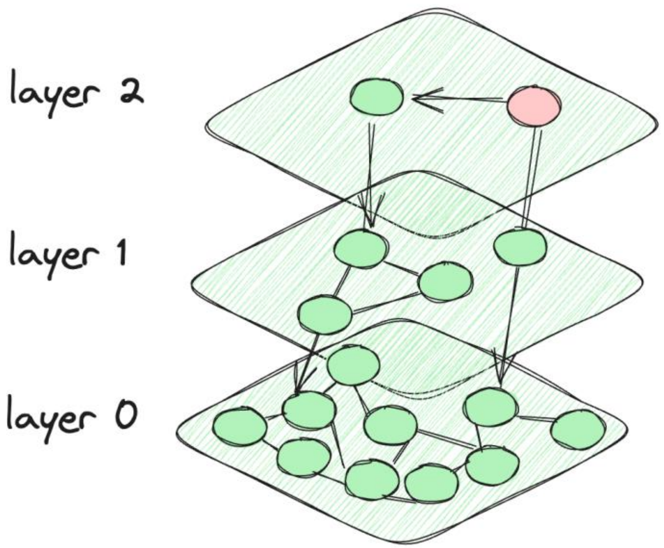


Image source https://medium.com/@myscale/understanding-vector-indexing-a-comprehensive-guide-d1abe36ccd3c

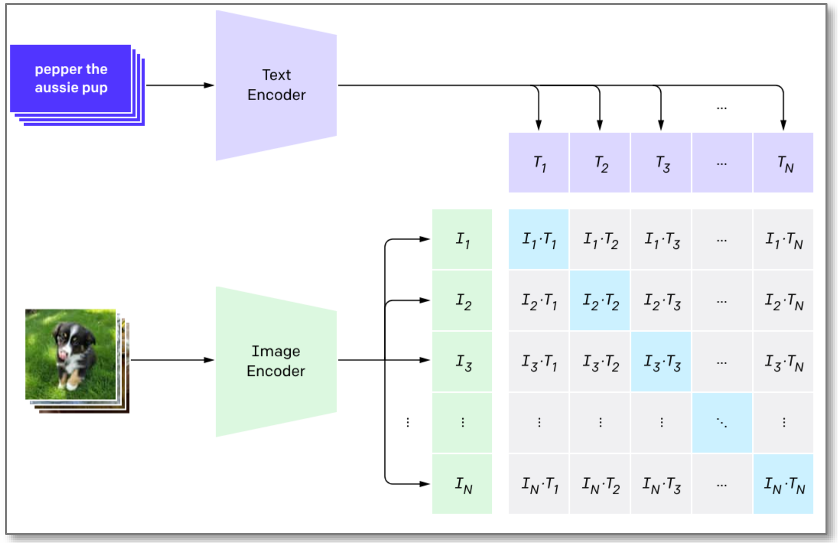
- Algorithm searchers for nearest neighbour in
  - Implemented in FAISS (Facebook AI Similarity Search) https://en.wikipedia.org/wiki/FAISS
  - For more information see: https://medium.com/@myscale/understanding-vector-indexing-a-comprehensive-guide-d1abe36ccd3c

Multimodal embeddings

CLIP (Contrastive Language–Image Pre-training)

Align the **embedding spaces!**

- embeddings can be generated for **text**
  - e.g. using SentenceTransformer
- and **also for images**
  - e.g. using ResNet or VisionTransformer
- using **contrastive learning** we can force the two spaces to agree



How?

- simply take a set of **<image, text> pairs**, e.g. images and their text captions
- and then for a **batch** of such pairs:
  - training classifier to identify
  - which piece of text describes which image
  - and which image describes which piece of text

Query:
"soccer fan elephant riding a bicycle"

Why would we want an aligned embedding space?

- allows for semantic **image search** using a **text queries**

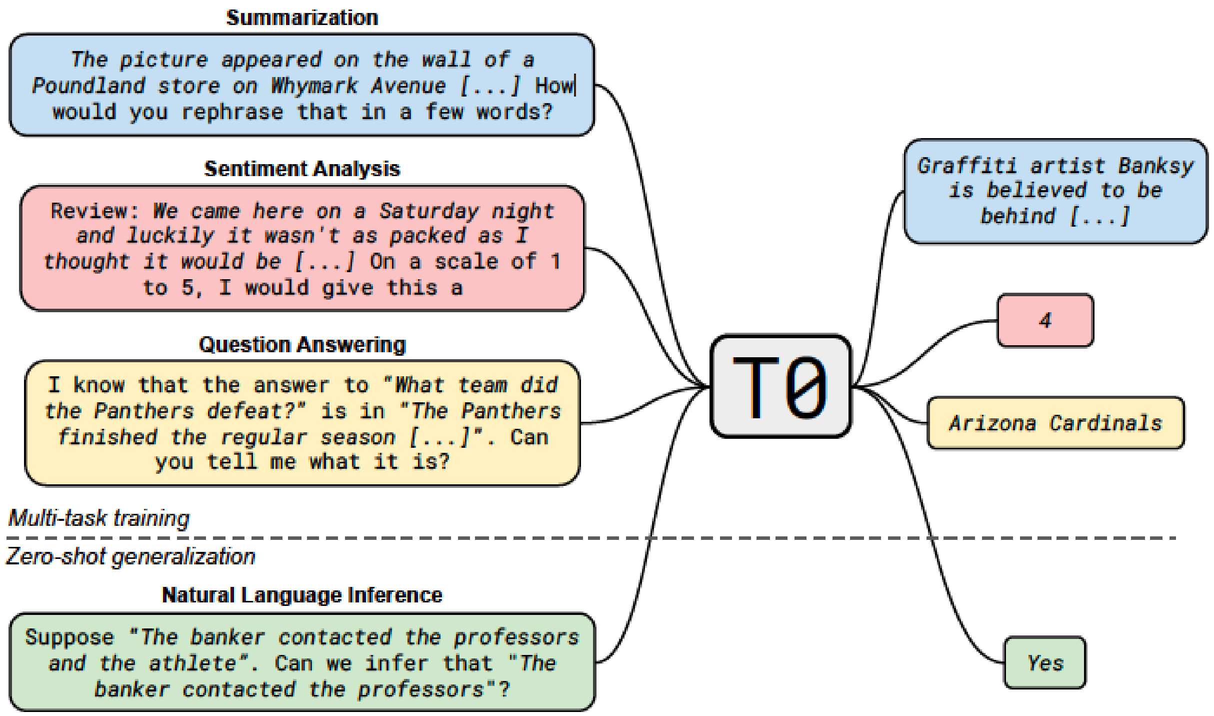
"inter-milan supporting elephant riding bicycle", Images generated by:
https://www.bing.com/images/create/inter-milan-supporting-elephant-riding-bicycle/1-6628ca5a038c4bd99594c90df8ef4bd8

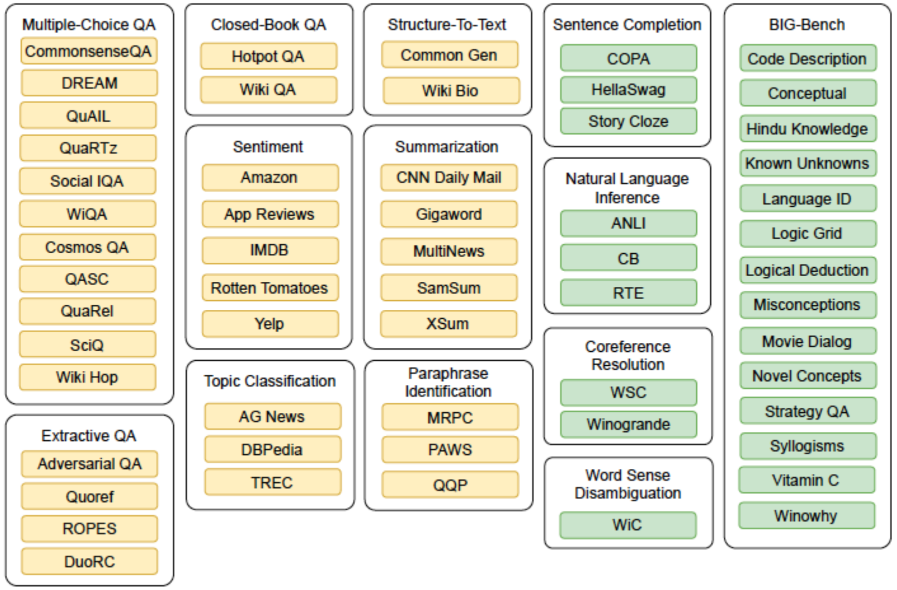
Multi-task Learning

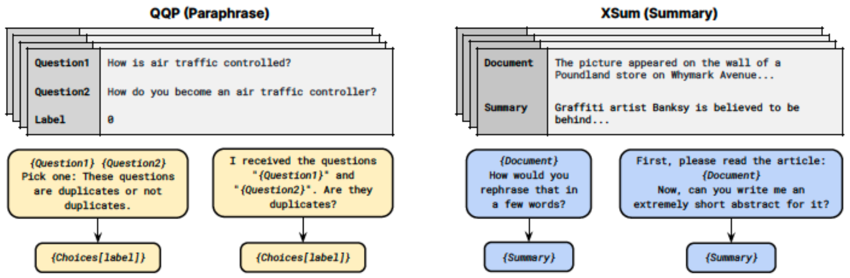
LMs are general purpose models …

So people have trained them to be multi-task

- and found that multi-task models often **outperform** models trained to perform a single task
- some even try to learn the best prompt for each task







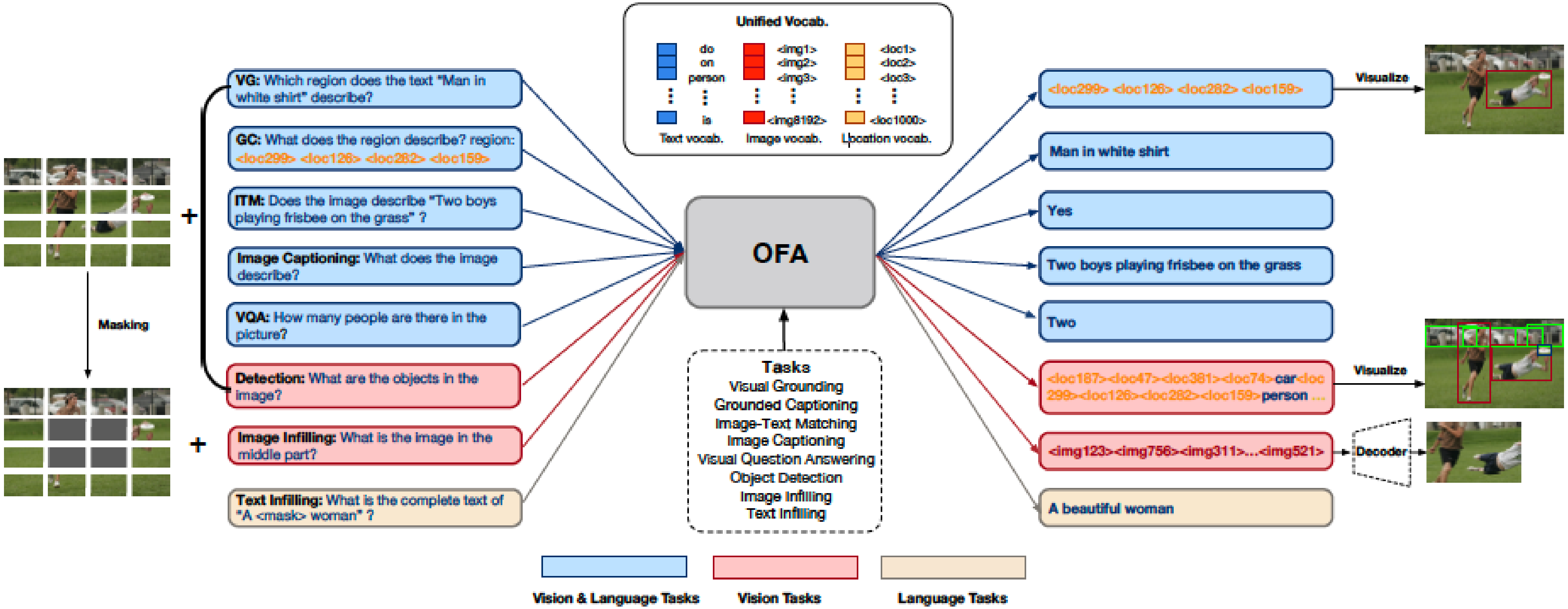
Source: Sanh et al. "MULTITASK PROMPTED TRAINING ENABLES ZERO-SHOT TASK GENERALIZATION" https://arxiv.org/pdf/2110.08207.pdf

Multi-modal Models

Multimodal learning…

Transformer architecture is very flexible:

- relatively easy to extend text-to-text models to multimodal (text+image) settings
- allows for learning of tasks across all media …



Source: Wang et al. "OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework" https://arxiv.org/abs/2202.03052

Conclusions

Conclusions

"transforming robot", image source:
https://www.bing.com/images/create/transforming-robot/1-6628c4a3b05d455e937ddbeaa24ab625

MANY applications of Transformer Architecture

- pairwise text classification
- document translation and summarization
- document similarity estimation
- semantic search
- image search
- multi-task learning