Advanced NLP Tasks

Introduciton

Introduction

Information extraction (IE) is the task of **automatically extracting structured information from unstructured** and/or **semi-structured** machine-readable **documents** and other electronically represented sources.

Introduction

As NLP evolves, so do IE tasks. Traditional tasks evolve, and new ones emerge out of necessity.

What are the most common IE tasks, and what are their related tasks?

Contents

Named Entity Recognition (NER)

Part-of-Speech Tagging (POS)

Named entity recognition (**NER**), aims at **identifying real-world entity mentions from texts**, and **classifying them** into **predefined types**.

Example:
"Suxamethonium infusion rate and observed fasciculations."

"Suxamethonium chloride (Sch) was administred i.v."

Part-of-Speech Tagging (POS)

We wish to predict an output vector $\mathbf{y} = (y_1, y_1, \ldots, y_L)$, of random variables, given an observed characteristic vector $\mathbf{x} = (x_1, x_2, \ldots, x_L)$

$\mathbf{y}$ takes it value from a list of N possible values.

Part-of-Speech Tagging (POS)

POS is the process of mapping words in a text with a label corresponding to their grammatical class.

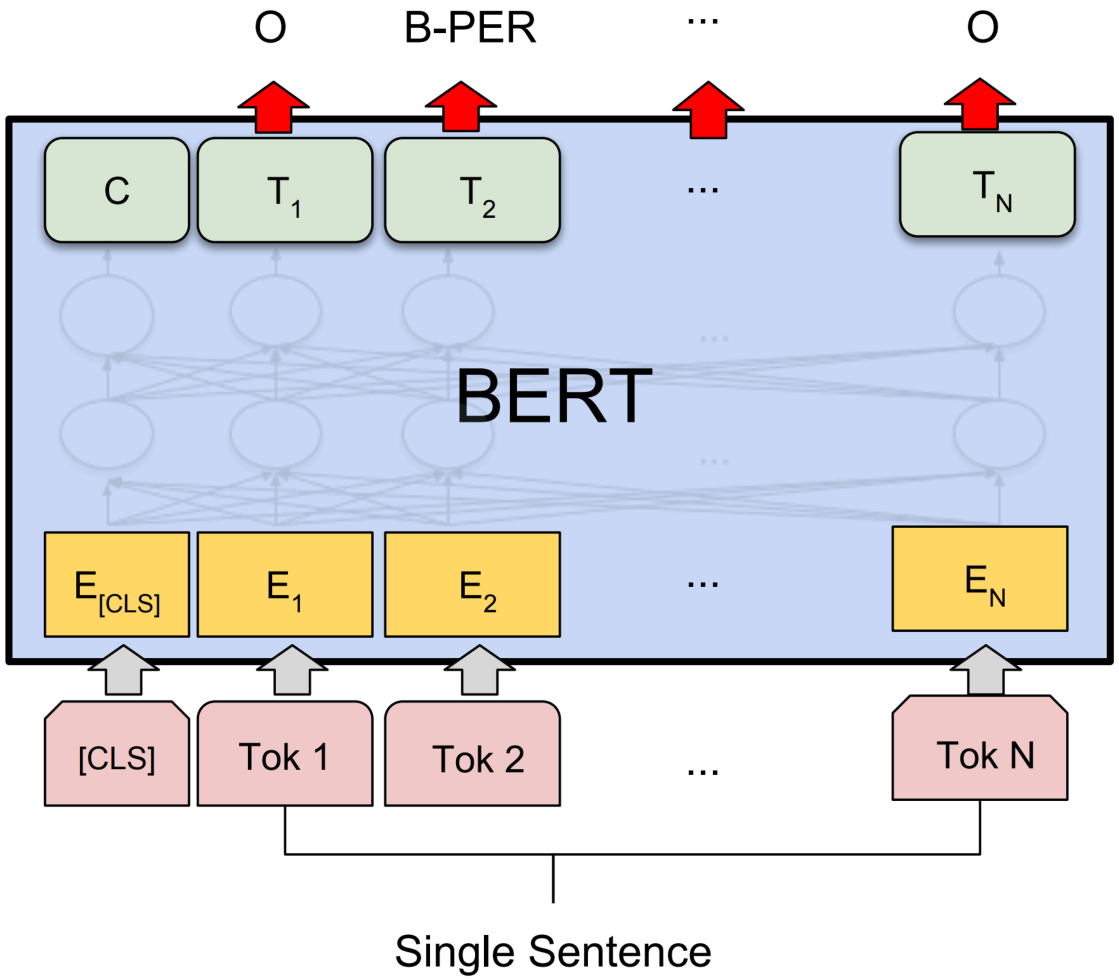
("He", "likes", "to", "drink", "tea"), $\rightarrow$ ("PERSONAL PRONOUN", "VERB", "TO", "VERB", "NOUN").

Part-of-Speech Tagging (POS)

There several levels of granularity.: using [the tag set for english]

("He", "likes", "to", "drink", "tea"), $\rightarrow$ ("PRP", "VBP", "TO", "VB", "NN").

Conditional Random Field (CRF)



Conditional Random Field (CRF)

For each token in a sentence at position l we want to compute a probability p to belong to a class n.

$$p : f(\mathbf{x}, \theta)_l \mapsto ?$$

with $p \in [0, 1]$

Conditional Random Field (CRF)

Using the softmax function?

$$p : f(\mathbf{x}, \theta)_l \mapsto \frac{e^{f(\mathbf{x},\theta)_l^{(n)}}}{\sum_{n'=1}^{N} e^{f(\mathbf{x},\theta)_l^{(n')}}}$$

The probability given by the softmax function will not encode non-local dependencies!

Conditional Random Field (CRF)

We need to take sequential decisions: what if we add transition scores into our softmax?

$$p : f(\mathbf{x}, \theta)_l \mapsto \frac{e^{f(\mathbf{x},\theta)_l^{(n)} + t(y_l^{(n)}, y_{l-1})}}{\sum_{n'=1}^{N} e^{f(\mathbf{x},\theta)_l^{(n')} + t(y_l^{(n')}, y_{l-1})}}$$

But this is the probability for one token to belong to a class, we want to compute the probability of a whole sequence of label at once...

Conditional Random Field (CRF)

$$P(\mathbf{y}|\mathbf{x}) = \prod_{l=2}^{L} p(\mathbf{y}|f(\mathbf{x},\theta)_l)$$

$$= \prod_{l=2}^{L} \frac{e^{f(\mathbf{x},\theta)_l^{(n)} + t(y_l^{(n)}, y_{l-1})}}{\sum_{n'=1}^{N} e^{f(\mathbf{x},\theta)_l^{(n')} + t(y_l^{(n')}, y_{l-1})}}$$

Conditional Random Field (CRF)

$$P(\mathbf{y}|\mathbf{x}) = \frac{exp[\sum_{l=2}^{L}(f(\mathbf{x},\theta)_l^{(n)} + t(y_l^{(n)}, y_{l-1}))]}{\sum_{n'=1}^{N} exp[\sum_{l=2}^{L}(f(\mathbf{x},\theta)_l^{(n')} + t(y_l^{(n')}, y_{l-1}))]}$$

$$= \frac{exp[\sum_{l=2}^{L}(U(\mathbf{x}, y_l^{(n)}) + T(y_l^{(n)}, y_{l-1}))]}{\sum_{n'=1}^{N} exp[\sum_{l=2}^{L}(U(\mathbf{x}, y_l^{(n')}) + T(y_l^{(n')}, y_{l-1}))]}$$

$$= \frac{exp[\sum_{l=2}^{L}(U(\mathbf{x}, y_l^{(n)}) + T(y_l^{(n)}, y_{l-1}))]}{Z(\mathbf{x})}$$

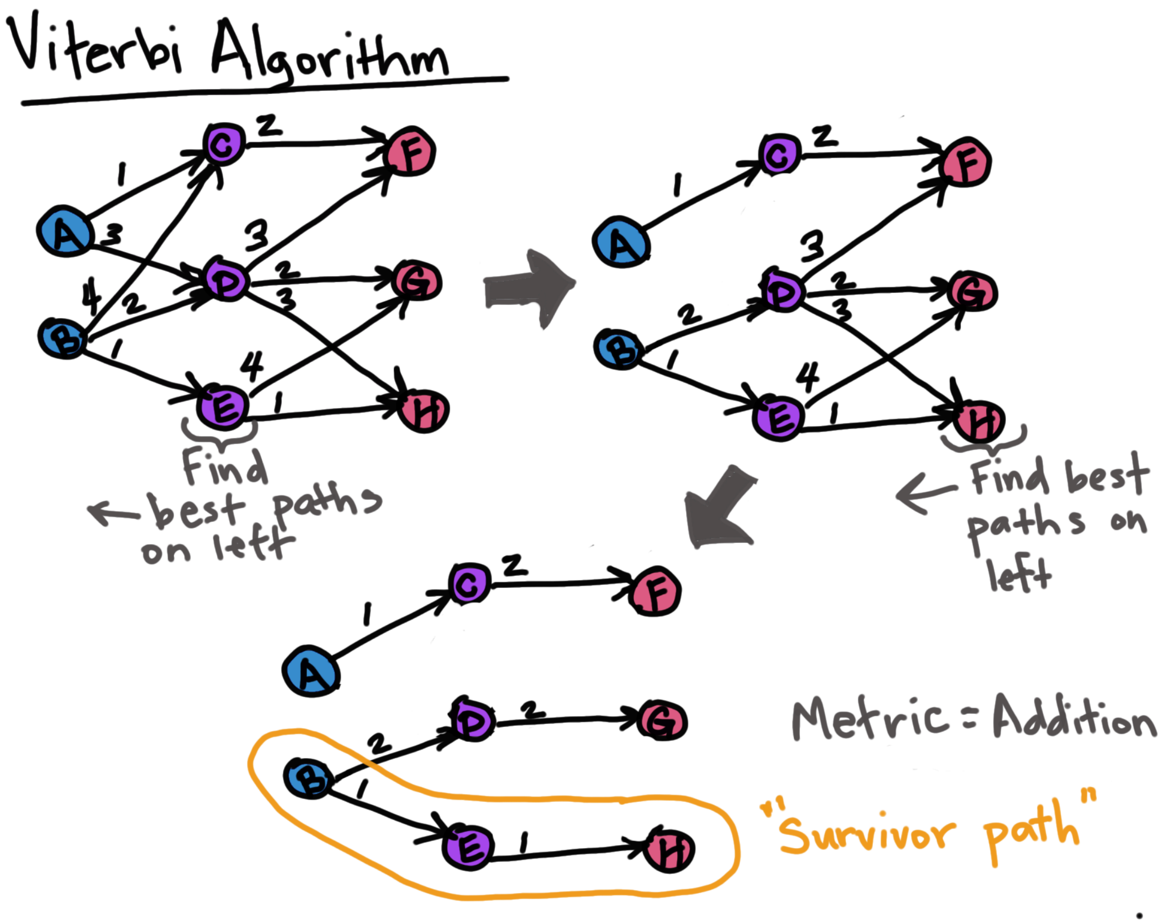
Conditional Random Field (CRF)

$Z(\mathbf{x})$ is commonly referred as the partition function. However, its not trivial to compute: we'll end up with a complexity of $\mathcal{O}(N^L)$.

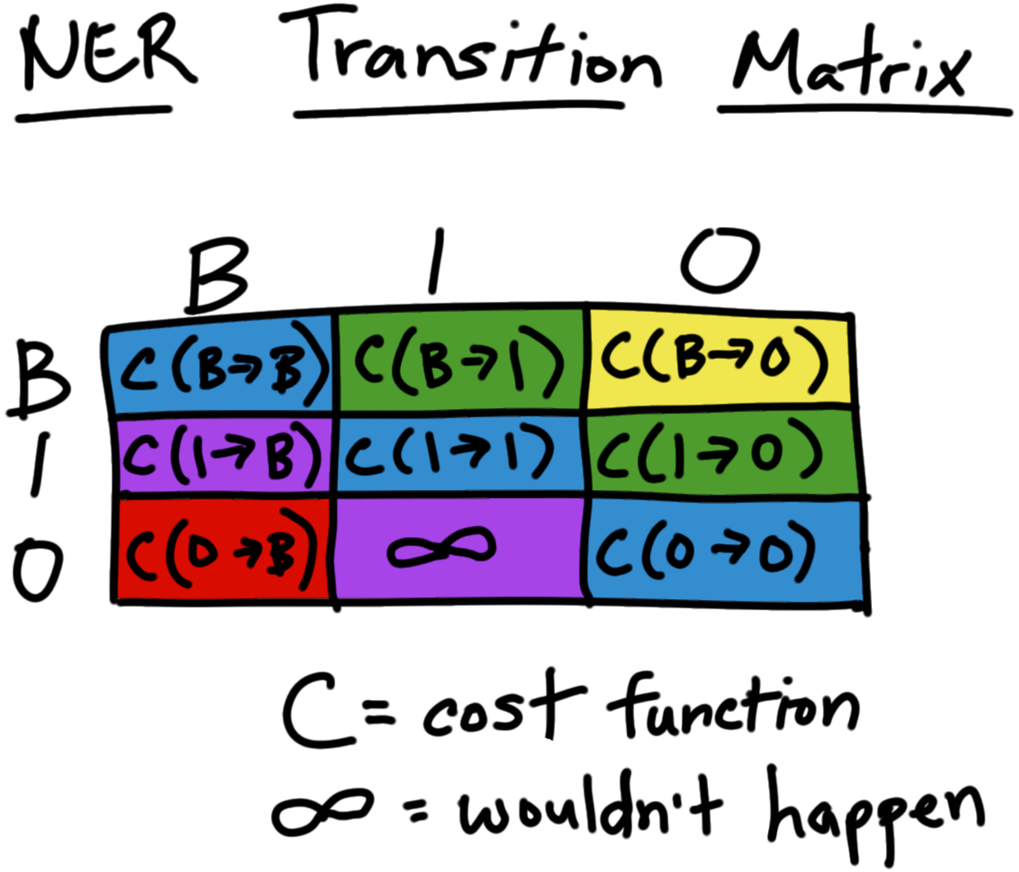
Where N is the number of possible labels and L the sequence length.

How do we proceed?

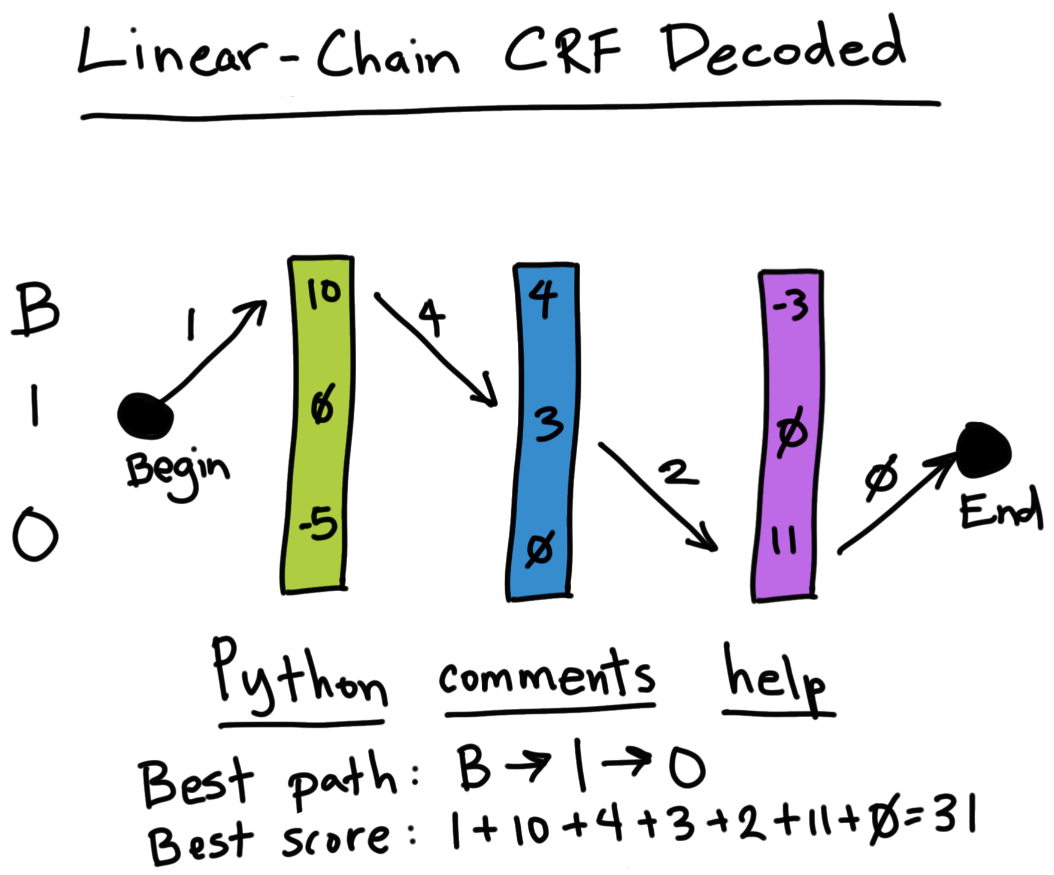
Conditional Random Field (CRF)



Conditional Random Field (CRF)



Conditional Random Field (CRF)



Conditional Random Field (CRF)

Negative log-likelihood:

$$\begin{aligned}
\mathcal{L} &= -log(P(\mathbf{y}|\mathbf{x})) \\
&= -log(\frac{exp[\sum_{l=2}^{L}(U(\mathbf{x}, y_l^{(n)}) + T(y_l^{(n)}, y_{l-1}))]}{Z(\mathbf{x})}) \\
&= -[log(exp[\sum_{l=2}^{L}(U(\mathbf{x}, y_l^{(n)}) + T(y_l^{(n)}, y_{l-1}))]) - log(Z(\mathbf{x}))] \\
&= log(Z(\mathbf{x})) - \sum_{l=2}^{L}(U(\mathbf{x}, y_l^{(n)}) + T(y_l^{(n)}, y_{l-1}))
\end{aligned}$$

Conditional Random Field (CRF)

There is an effective way to compute $log(Z(\mathbf{x}))$ with a complexity of $\mathcal{O}(L)$ using the Log-Sum-Exp trick.

$$
\begin{aligned}
log(Z(\mathbf{x})) &= log(\sum_{n'=1}^{N} exp[\sum_{l=2}^{L} (U(\mathbf{x}, y_l^{(n')}) + T(y_l^{(n')}, y_{l-1}))]) \\
&= c + log(\sum_{n'=1}^{N} exp[\sum_{l=2}^{L} (U(\mathbf{x}, y_l^{(n')}) + T(y_l^{(n')}, y_{l-1})) - c])
\end{aligned}
$$

Conditional Random Field (CRF)

If we fix
$c = max\{U(\mathbf{x}, y_l^{(1)}) + T(y_l^{(1)}, y_{l-1}), \ldots, U(\mathbf{x}, y_l^{(N)}) + T(y_l^{(N)}, y_{l-1})\}$
we ensure that the largest positive exponentiated term is $exp(0) = 1$.

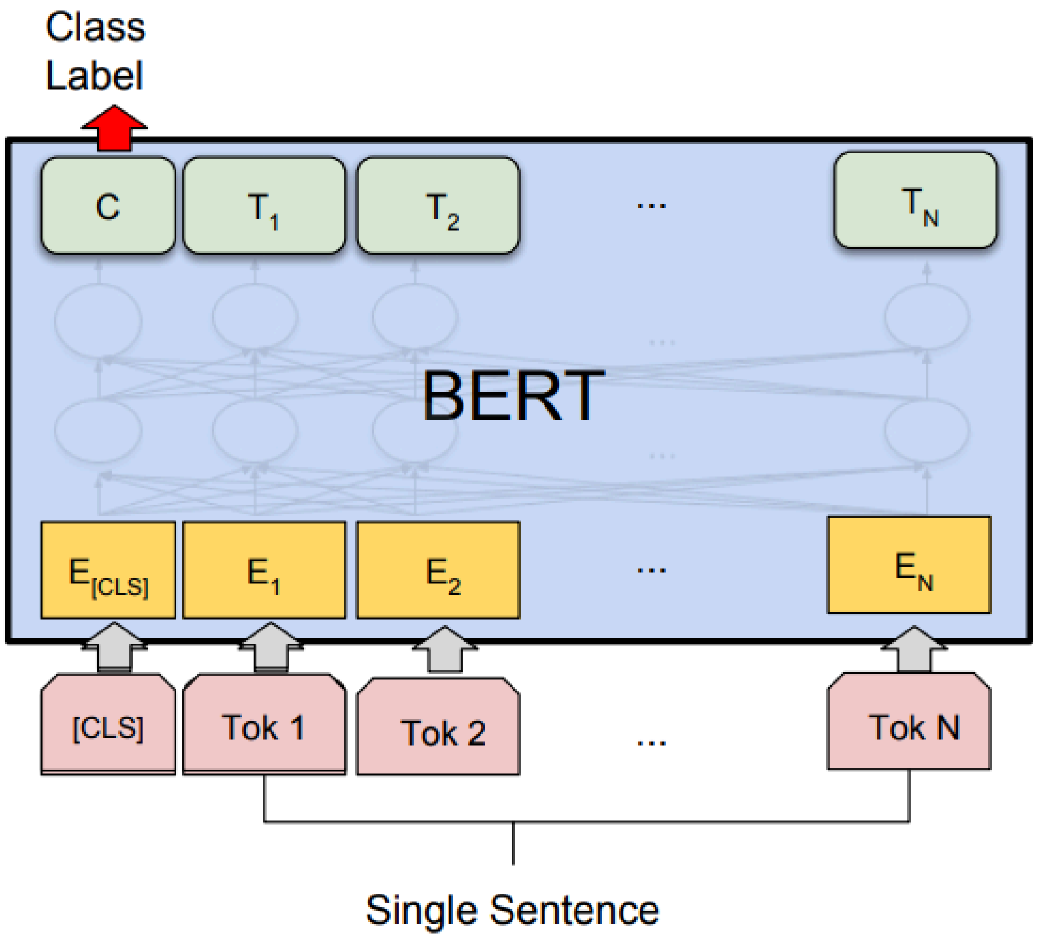
Sentiment Analysis

Sentiment Analysis

Sentiment analysis is a sentence classification task aiming at **automatically mapping data to their sentiment**.

It can be **binary** classification (e.g., positive or negative) or **multiclass** (e.g., enthusiasm, anger, etc)

Sentiment Analysis



Sentiment Analysis

The loss can be the likes of cross-entropy (CE), binary cross-entropy (BCE) or KL-Divergence (KL).

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{n'=1}^{N} y^{(n)}.\, log(f(\mathbf{x}, \theta)^{(n)})$$

$$\mathcal{L}_{BCE} = -y^{(n)}.\, log(f(\mathbf{x}, \theta)^{(n)}) + (1 - y^{(n)}).\, (1 - f(\mathbf{x}, \theta)^{(n)})$$

$$\mathcal{L}_{KL} = -\frac{1}{N}\sum_{n'=1}^{N} y^{(n)}.\, log(\frac{y^{(n)}}{f(\mathbf{x}, \theta)^{(n)}})$$

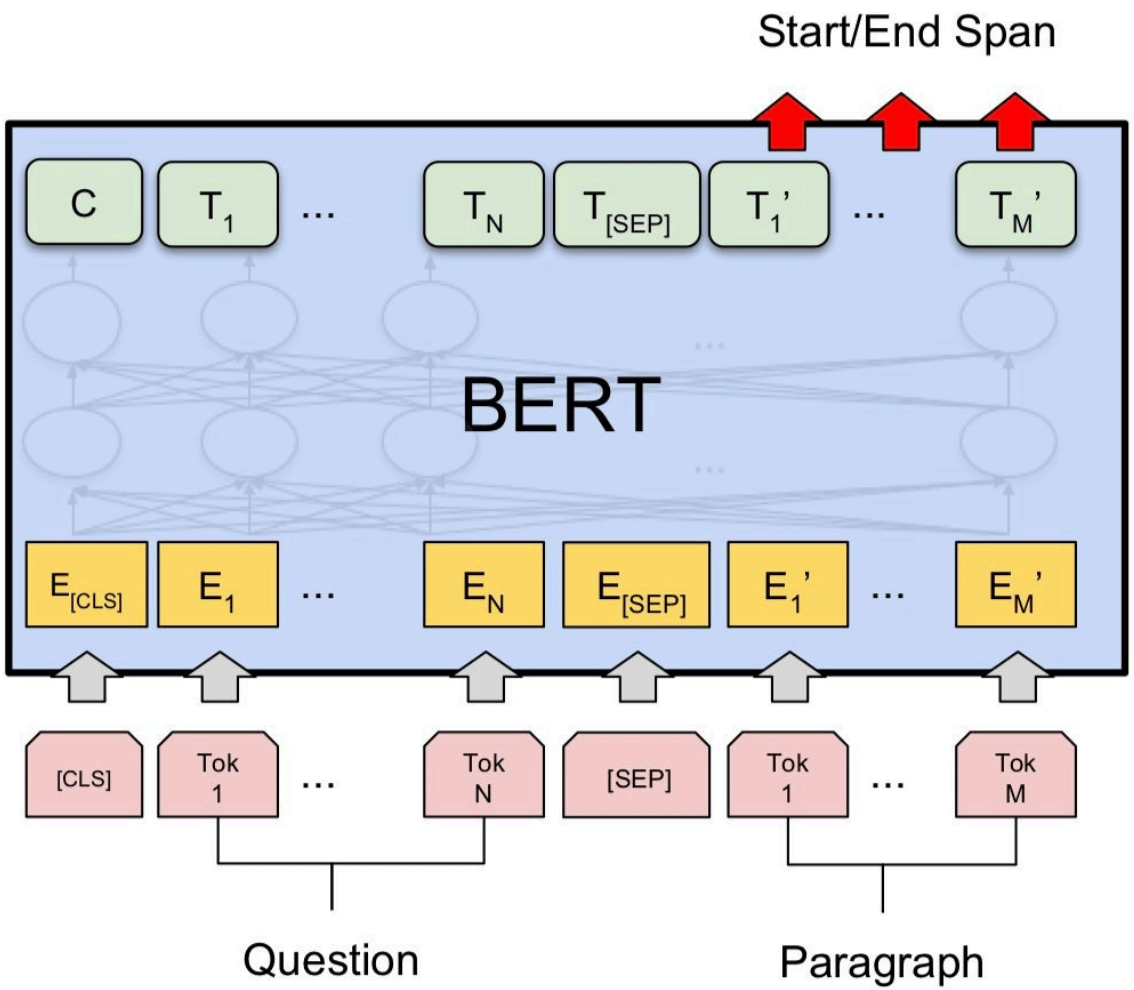
Question Answering (QA)

QA

QA is the task of **retrieving a span of text from a context** that is best suited to answer a question.

This task is extractive, and can be seen as information retrieval (more on that later).

QA



Start/End Span
C
T_1
...
T_N
$T_{[SEP]}$
T_1'
...
T_M'
BERT
$E_{[CLS]}$
E_1
...
E_N
$E_{[SEP]}$
E_1'
...
E_M'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
Tok M
Question
Paragraph

QA

The loss is the cross entropy over the output of the starting token and the ending one:

$$\mathcal{L}_{CE_{QA}} = \mathcal{L}_{CE_{start}} + \mathcal{L}_{CE_{end}}$$

Natural Language Inference (NLI)

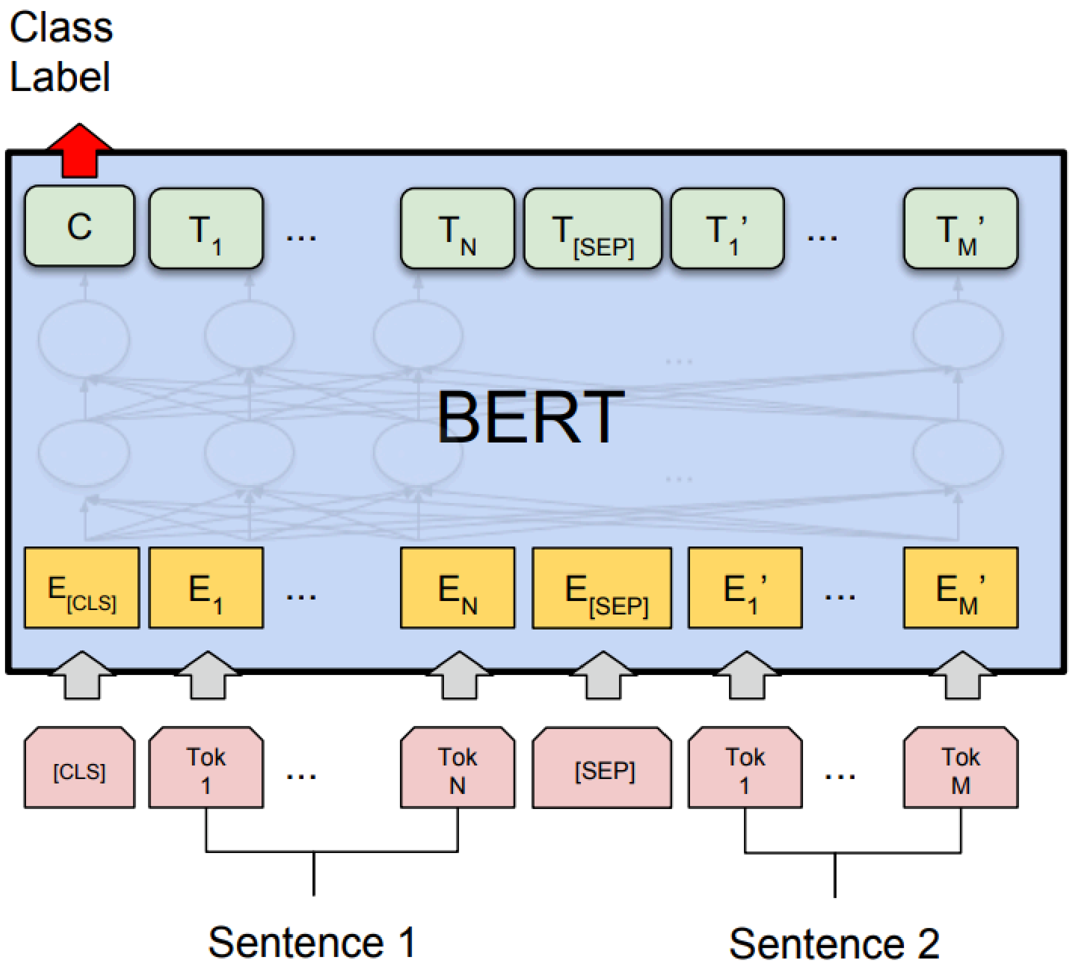
NLI

NLI is the task of **determining whether a "hypothesis" is true (entailment), false (contradiction), or undetermined (neutral)** given a "premise". [1]

NLI

Premise	Label	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction	The man is sleeping.
An older and younger man smiling.	neutral	Two men are smiling and laughing at the cats playing on the floor.
A soccer game with multiple males playing.	entailment	Some men are playing a sport.

NLI



Class
Label
C
T_1
...
T_N
$T_{[SEP]}$
T_1'
...
T_M'
BERT
$E_{[CLS]}$
E_1
...
E_N
$E_{[SEP]}$
E_1'
...
E_M'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
Tok M
Sentence 1
Sentence 2

NLI

The loss is simply the cross entropy or the divergence over the output of the `CLS` token and the true label.

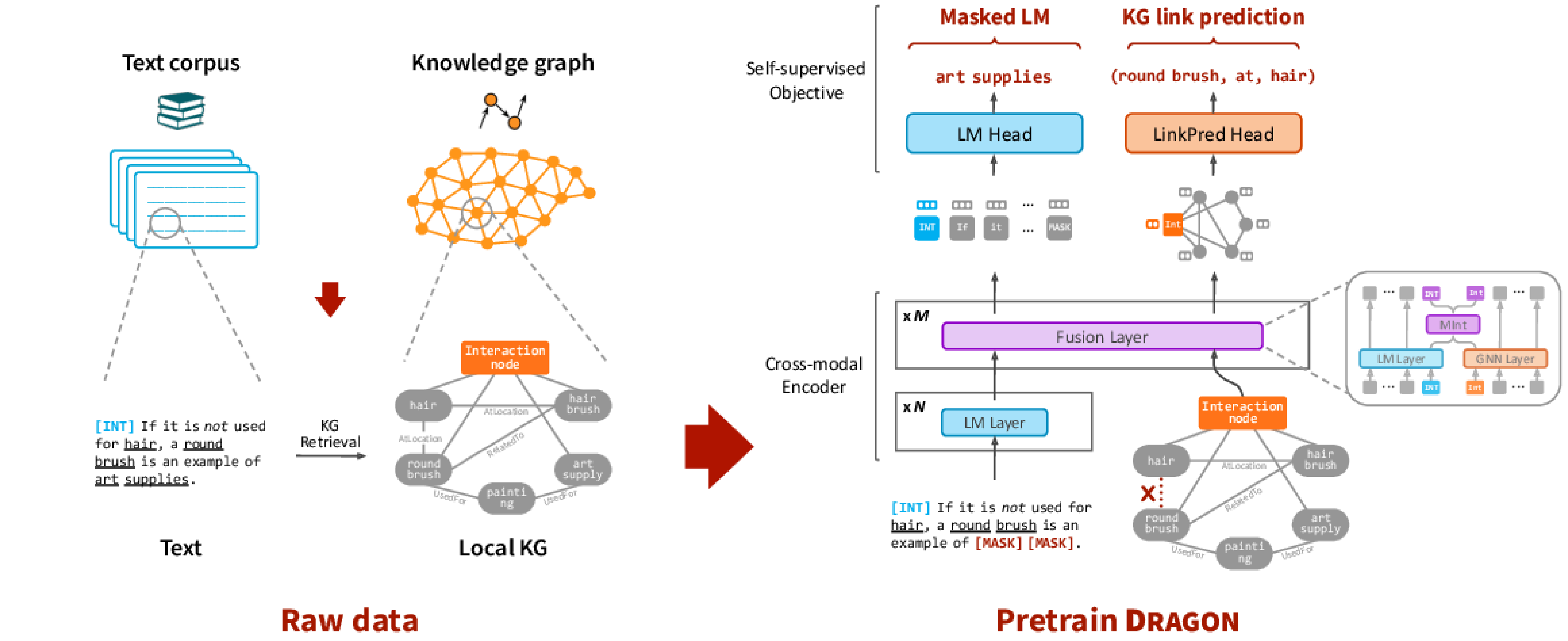
$$\mathcal{L}_{NLI} = \mathcal{L}_{CE_{CLS}}$$

We are trying to compress the information about both sentence in one `CLS` token via attention and decide about their relationship.

Is it possible to help the model infering more information with les text data?

Going Further: LM as Knowledge Graphs

Going Further: LM as Knowledge Graphs



Going Further: LM as Knowledge Graphs

	CSQA	OBQA	Riddle	ARC	CosmosQA	HellaSwag	PIQA	SIQA	aNLI
RoBERTa [18]	68.7	64.9	60.7	43.0	80.5	82.3	79.4	75.9	82.7
QAGNN [8]	73.4	67.8	67.0	44.4	80.7	82.6	79.6	75.7	83.0
GreaseLM [9]	74.2	66.9	67.2	44.7	80.6	82.8	79.6	75.5	83.3
DRAGON (**Ours**)	**76.0**	**72.0**	**71.3**	**48.6**	**82.3**	**85.2**	**81.1**	**76.8**	**84.0**

1: Accuracy on downstream commonsense reasoning tasks. DRAGON consistently outperforms the existin

Improvements are mostly on dataset with few training examples and complicated examples (negations, non-verbal sentences, ...).

Going Further: LM as Knowledge Graphs

This architecture ***involves a KG ready to use beforehead and pre-training from scratch***.

How can we better **perform NLP task without having to retrain or fine-tune** a model?

Exploit LLMs capacities: Chain-of-thoughts & In context Learning

Exploit LLMs capacities

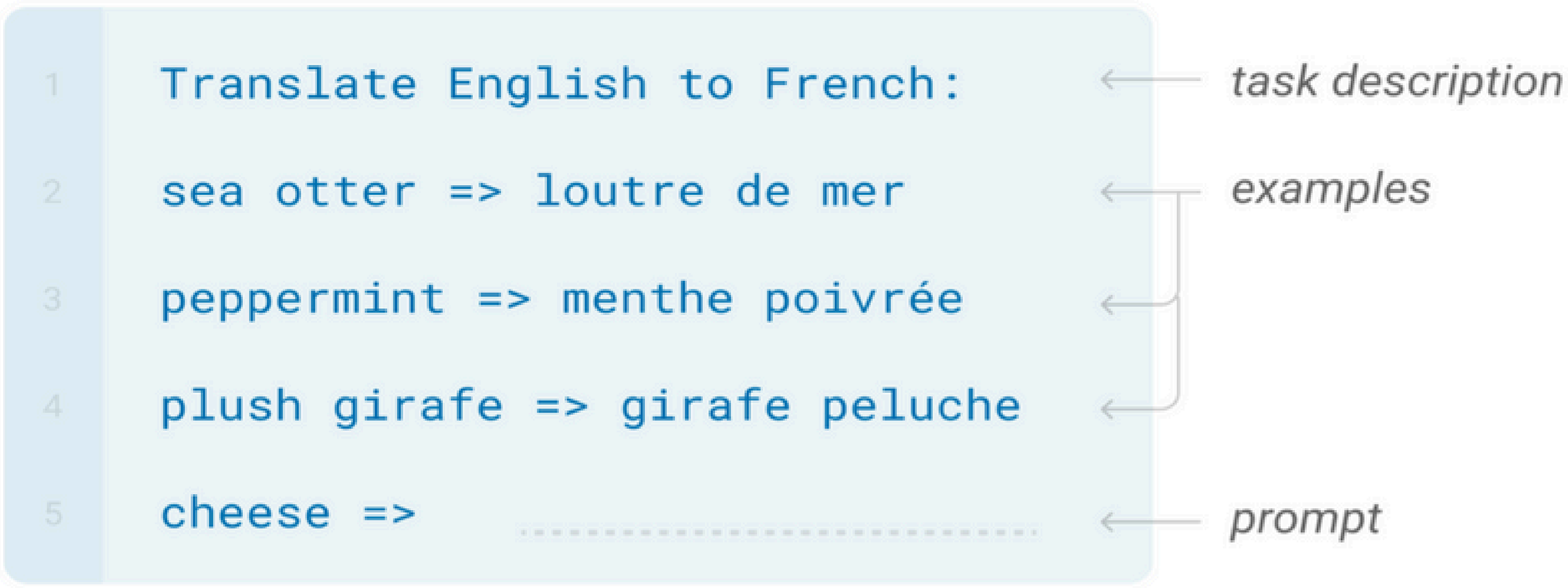
ICL enables LLMs to **learn new tasks** using natural language prompts **without explicit retraining or fine-tuning**.

The **efficacy** of ICL is **closely tied to** the model's **size**, training **data quality**, and **domain specificity**.

Exploit LLMs capacities

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Exploit LLMs capacities

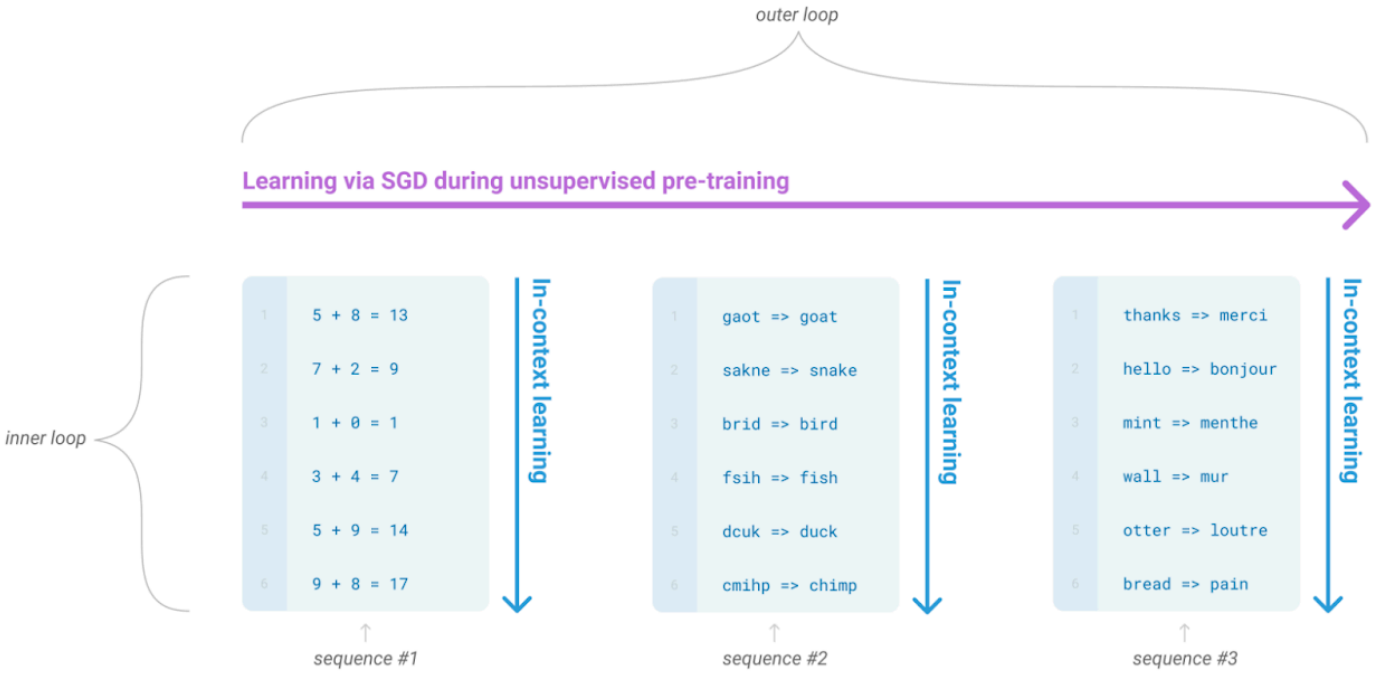


Figure 1.1: Language model meta-learning. During unsupervised pre-training, a language model develops a broad set of skills and pattern recognition abilities. It then uses these abilities at inference time to rapidly adapt to or recognize the desired task. We use the term "in-context learning" to describe the inner loop of this process, which occurs within the forward-pass upon each sequence. The sequences in this diagram are not intended to be representative of the data a model would see during pre-training, but are intended to show that there are sometimes repeated sub-tasks embedded within a single sequence.

Exploit LLMs capacities

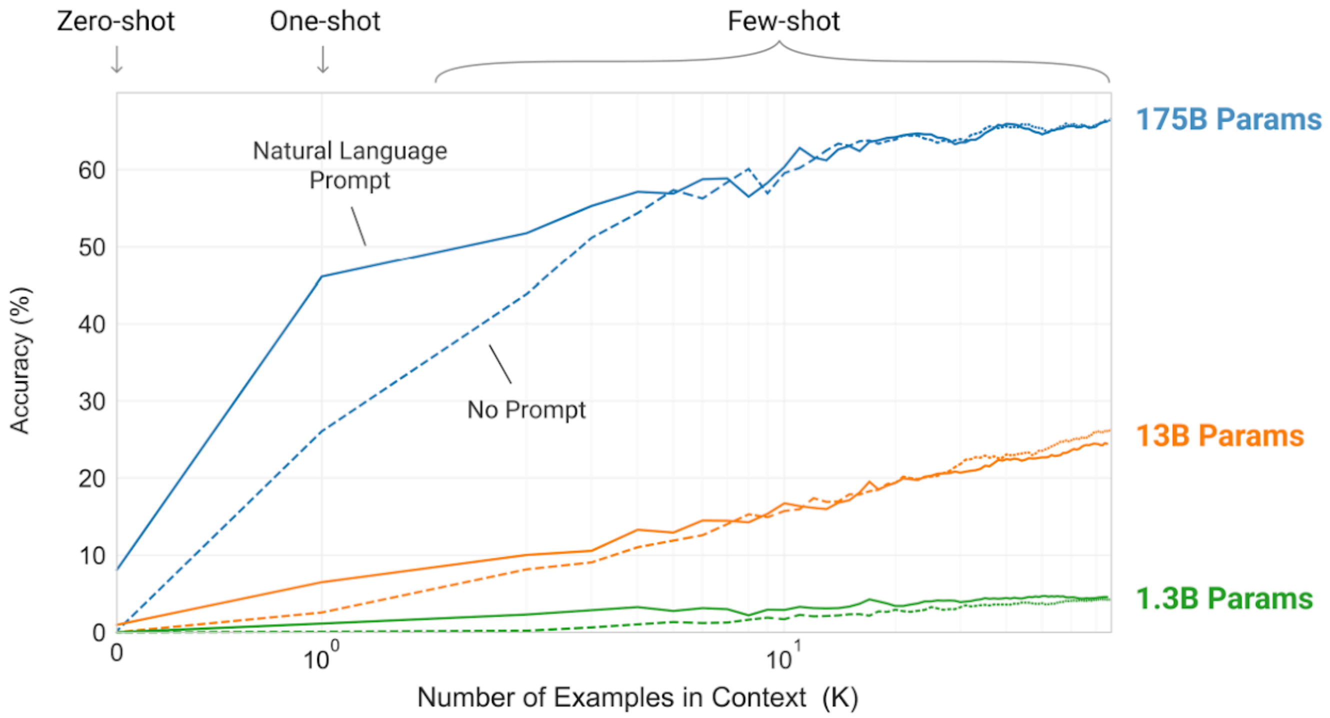
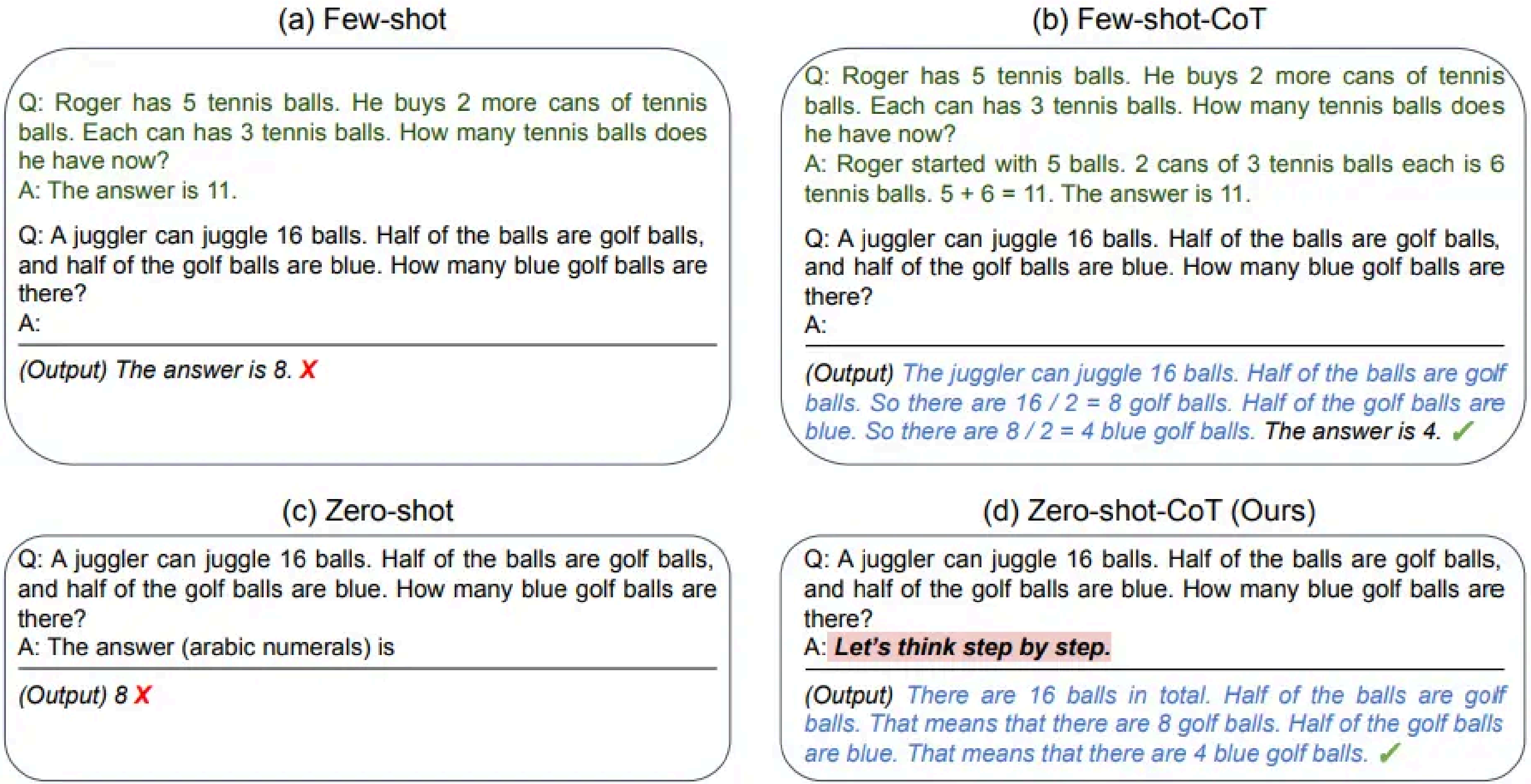


Figure 1.2: Larger models make increasingly efficient use of in-context information. We show in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task description (see Sec. 3.9.2). The steeper "in-context learning curves" for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.

Exploit LLMs capacities



Questions?