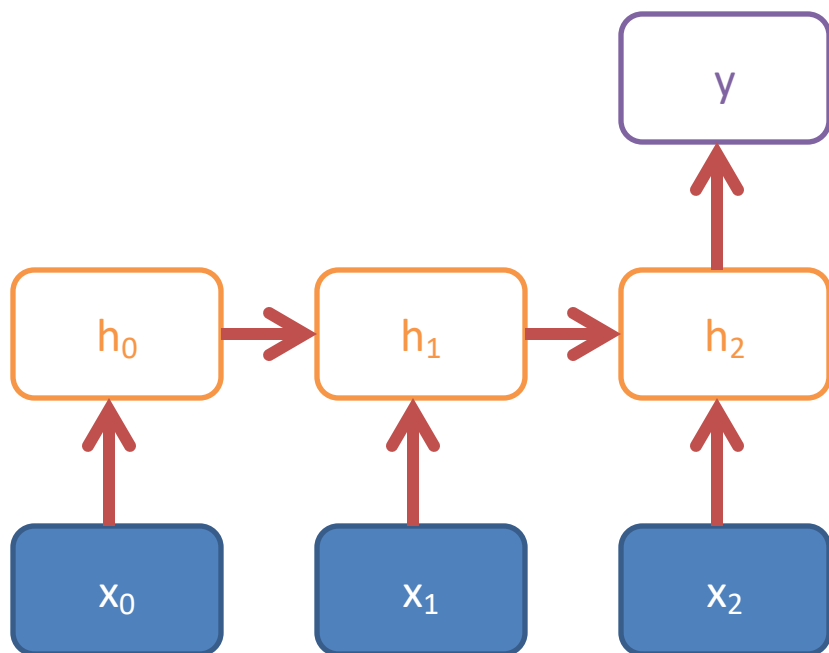


(Even More) Language Modeling: Attention, and Building Blocks of Transformers

CMSC 473/673

Frank Ferraro

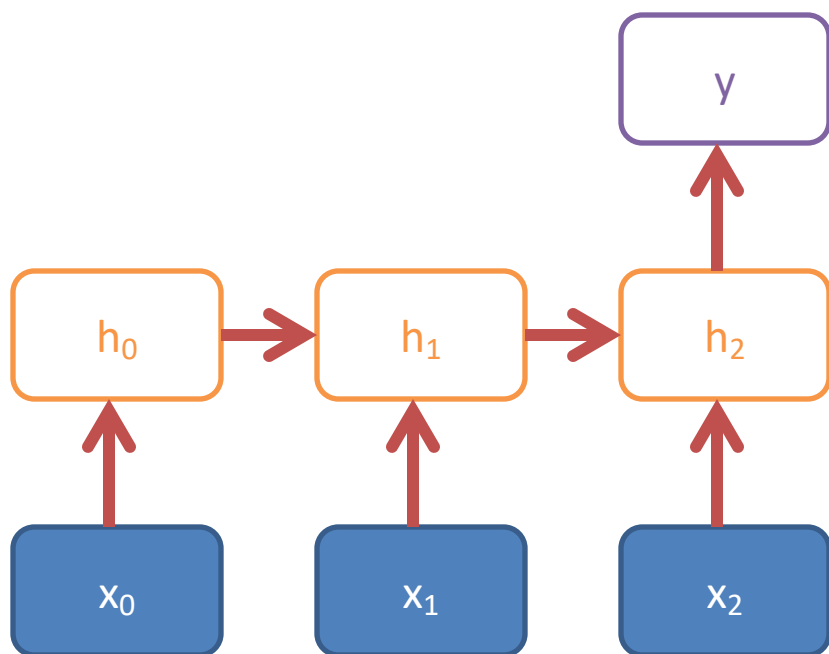
Sequence **Input**, Label **Output**



Recurrent: Sequence input, one output

Document classification
Action recognition in video (high-level)

Sequence **Input**, Label **Output**



Recurrent: Sequence input, one output

Document classification
Action recognition in video (high-level)

Think of this as generalizing using maxent models to build discriminatively trained classifiers

$$p(y | x) = \text{maxent}(x, y)$$



$$p(y | x) = \text{recurrent_classifier}(x, y)$$

Example: RTE (many options)

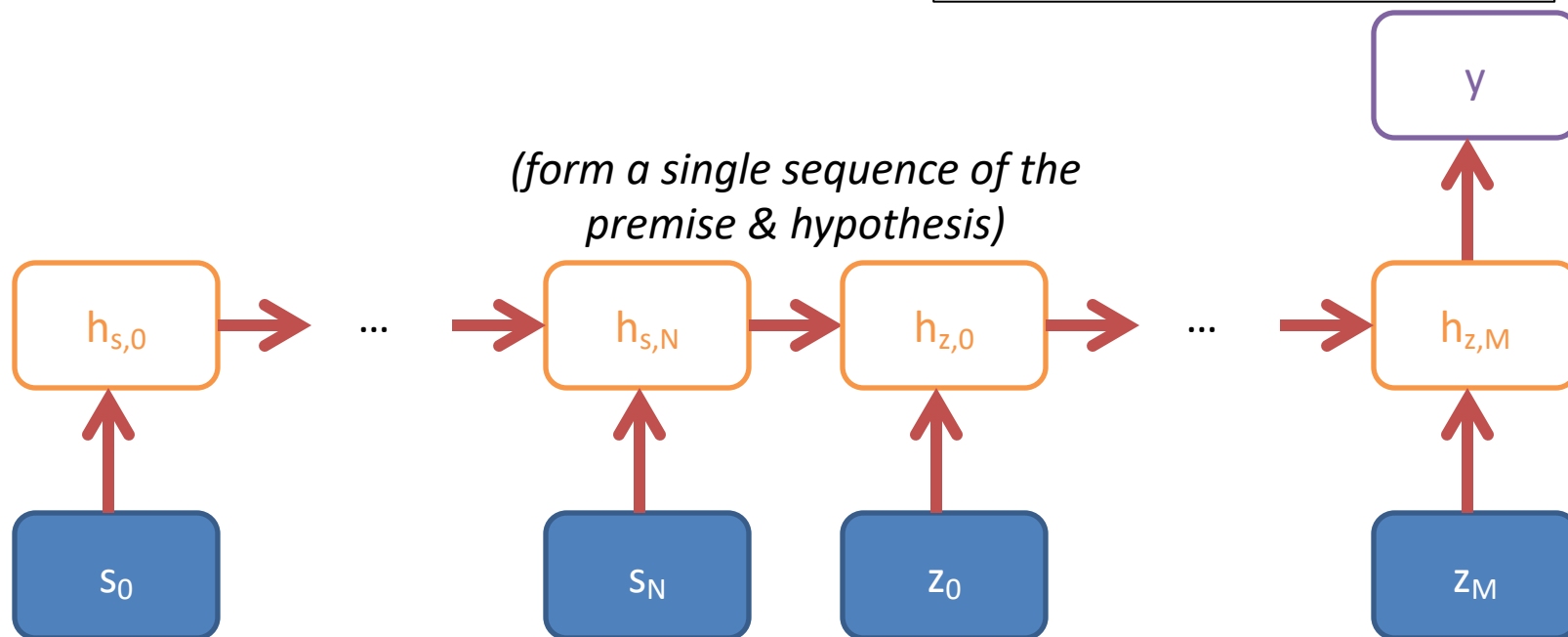
p (

ENTAILED

|

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
z: The Bulls basketball team is based in Chicago.

)



Example: RTE (many options)

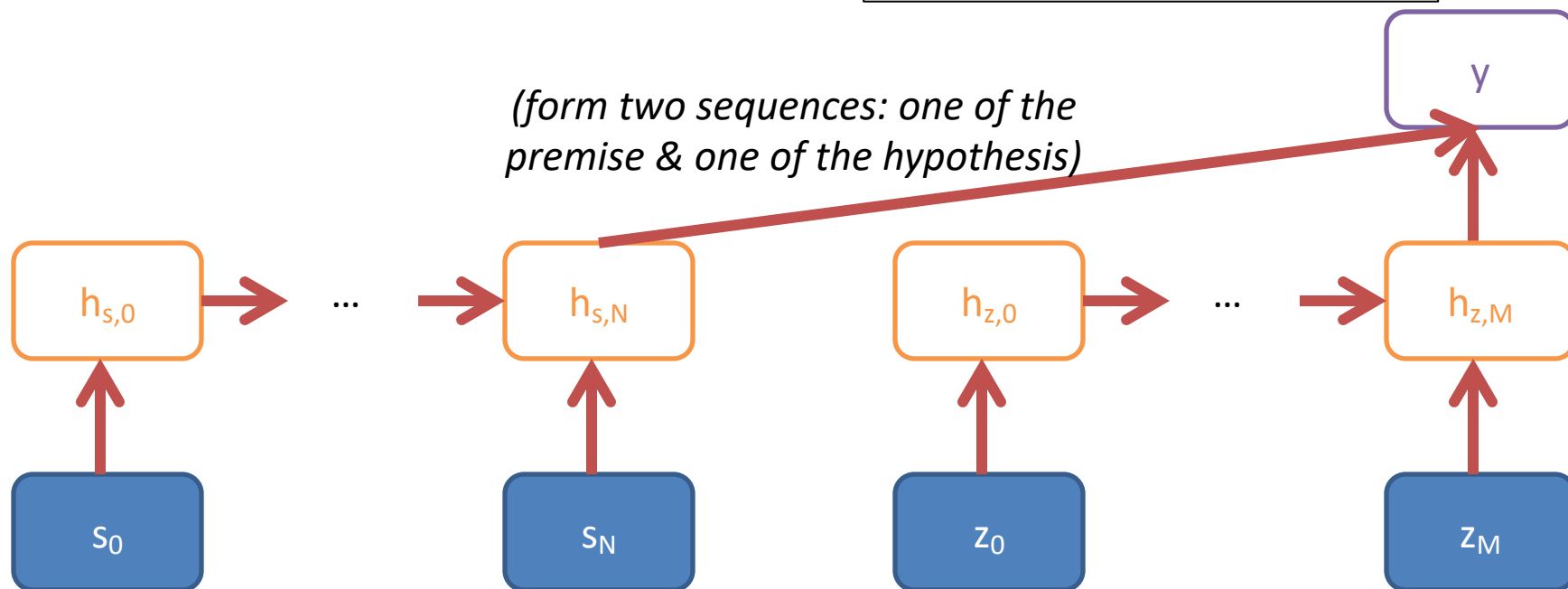
p (

ENTAILED

|

s : Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
 z : The Bulls basketball team is based in Chicago.

)



Reminder!

Many (but not all) of these
tasks fall into the
Sequence **Input**, Label **Output**

GLUE

<https://gluebenchmark.com/>

GLUE Tasks

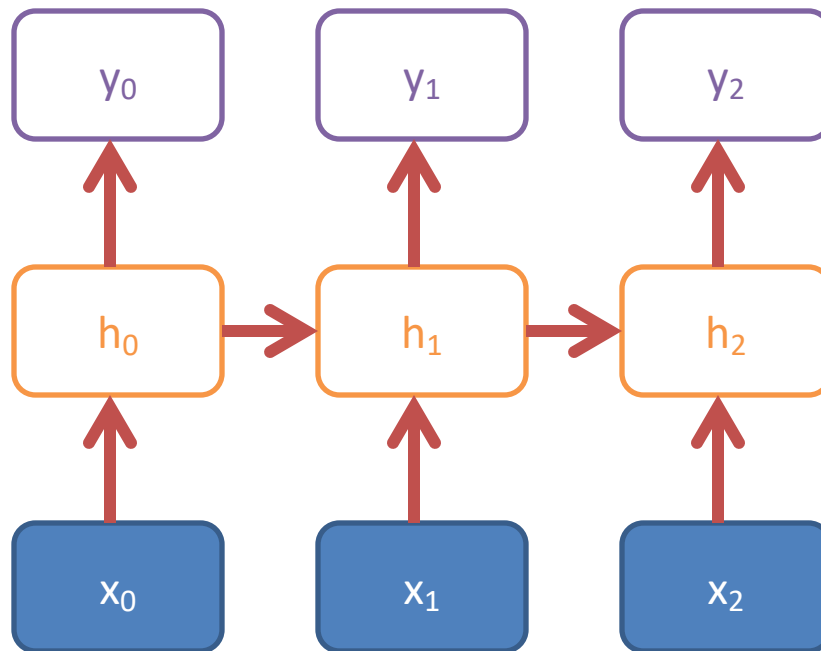
Name	Download
The Corpus of Linguistic Acceptability	
The Stanford Sentiment Treebank	
Microsoft Research Paraphrase Corpus	
Semantic Textual Similarity Benchmark	
Quora Question Pairs	
MultiNLI Matched	
MultiNLI Mismatched	
Question NLI	
Recognizing Textual Entailment	
Winograd NLI	
Diagnostics Main	

Name	Identifier
Broadcoverage Diagnostics	AX-b
CommitmentBank	CB
Choice of Plausible Alternatives	COPA
Multi-Sentence Reading Comprehension	MultiRC
Recognizing Textual Entailment	RTE
Words in Context	WiC
The Winograd Schema Challenge	WSC
BoolQ	BoolQ
Reading Comprehension with Commonsense Reasoning	ReCoRD
Winogender Schema Diagnostics	AX-g

 **SuperGLUE**

<https://super.gluebenchmark.com/>

Sequence **Input**, Sequence **Output** ("sequence prediction": no time delay)



Recursive: Sequence input, Sequence output

Part of speech tagging
Named entity recognition

Example 1: Part of Speech Tagging

Noun

Verb

Noun

Prep

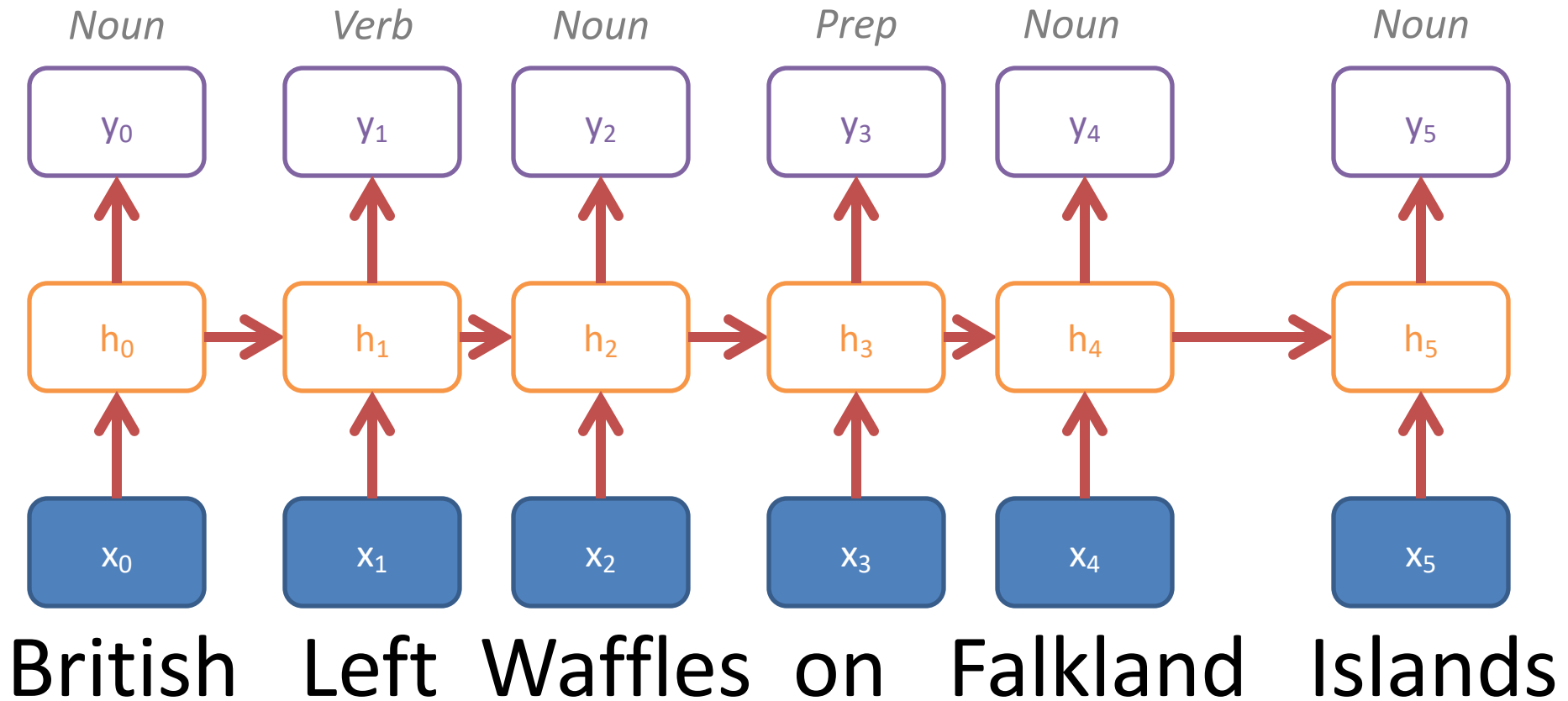
Noun

Noun

Task: Predict a **part-of-speech** tag
for each word in a provided
sentence

British Left Waffles on Falkland Islands

Example 1: Part of Speech Tagging



Example 2: Named Entity Recognition

Task: Predict a **named entity** tag
for each word in a provided
sentence

British Left Waffles on Falkland Islands



Reminder!

What are Named Entities?

Named entity recognition (NER)

Identify proper names in texts, and classification into a set of predefined categories of interest

- Person names

- Organizations (companies, government organisations, committees, etc)

- Locations (cities, countries, rivers, etc)

- Date and time expressions

- Measures (percent, money, weight etc), email addresses, Web addresses, street addresses, etc.

- Domain-specific: names of drugs, medical conditions, names of ships, bibliographic references etc.

Example 2: Named Entity Recognition

ORG

ORG

Other

Other

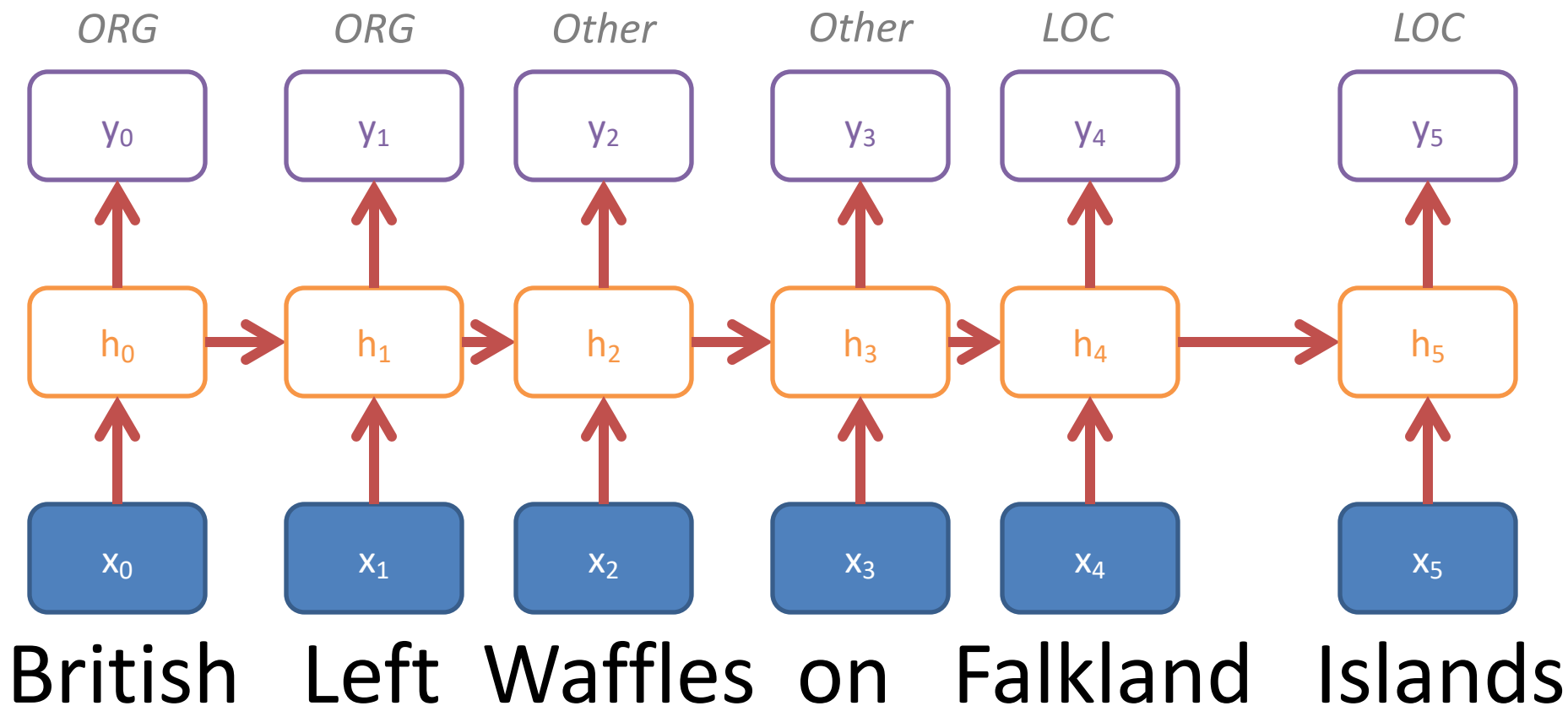
LOC

LOC

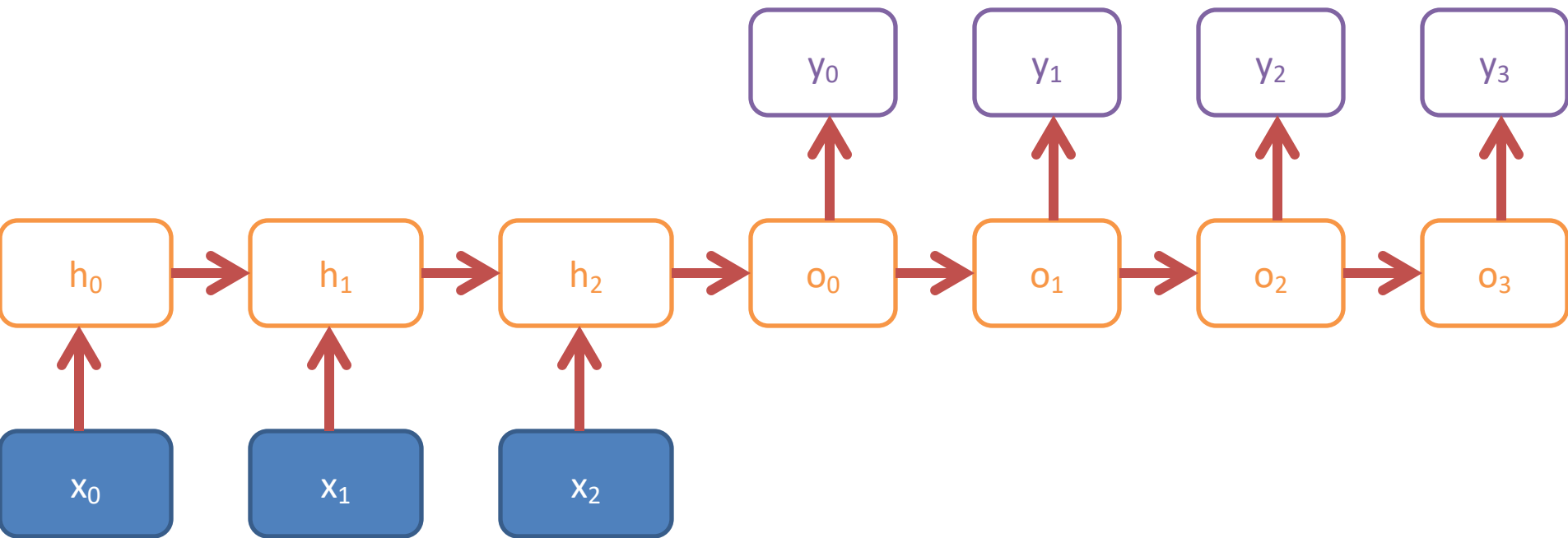
Task: Predict a **named entity** tag
for each word in a provided
sentence

British Left Waffles on Falkland Islands

Example: Named Entity Recognition



Sequence **Input**, Sequence **Output** ("sequence-to-sequence": time delay)



Recursive: Sequence input, Sequence output (time delay)

Machine translation
Sequential description
Summarization

Example: Translation

Translate French (observed) into English:

Le chat est sur la chaise.

variable # of
input words



The cat is on the chair.

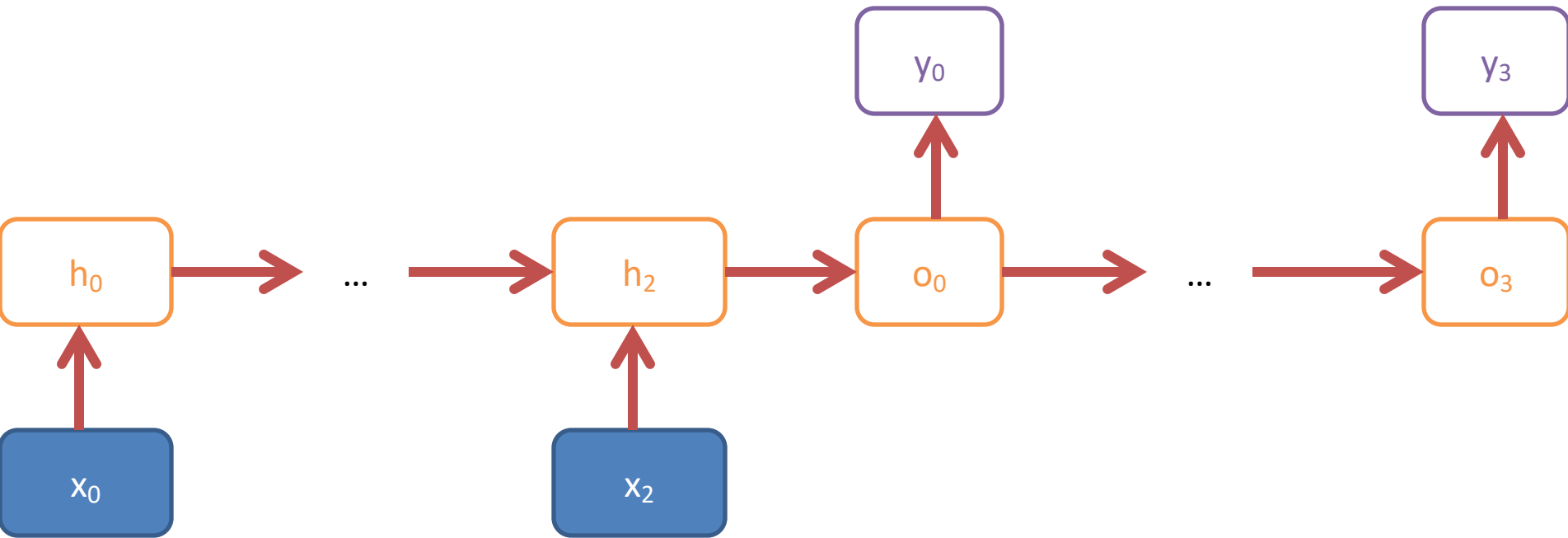
variable # of
output words

$$p(\textit{English}|\textit{French})$$

Example: Translation

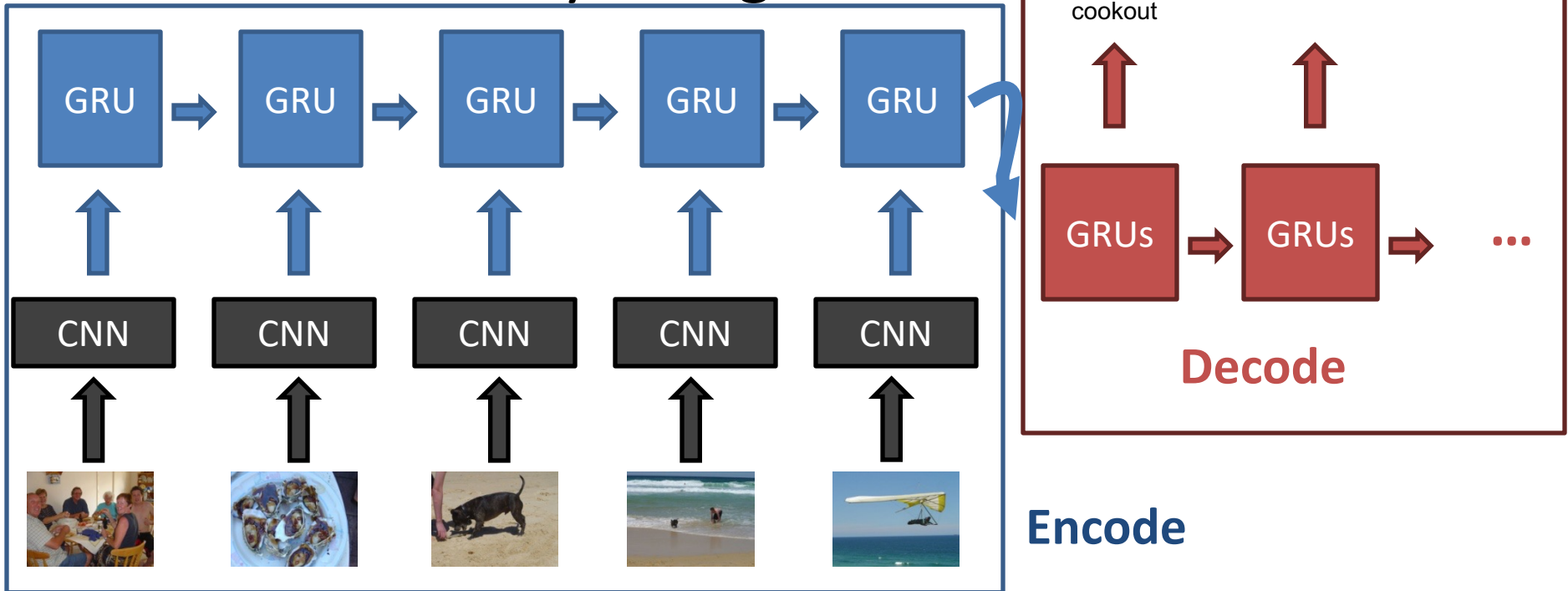
Translate French (observed) into English:

Le chat est sur la chaise.



The cat is on the chair.

RNN Output: Visual Storytelling



Human Reference

The family has gathered around the dinner table to share a meal together. They all pitched in to help cook the seafood to perfection. Afterwards they took the family dog to the beach to get some exercise. The waves were cool and refreshing! The dog had so much fun in the water. One family member decided to get a better view of the waves!

Huang et al. (2016)

The family got together for a cookout. They had a lot of delicious food. The dog was happy to be there. They had a great time on the beach. They even had a swim in the water.

Effective, but challenges remain

- Sequence **Input**,
Label **Output** (w/
time delay)
- Sequence **Input**,
Sequence **Output**
(w/o time delay)
- Sequence **Input**,
Sequence **Output**
(w/ time delay)

Core effective idea:

Use the basic recurrent
(autoregressive) structure to
capture longer-range dependencies
that enable us to map from **Input** to
Output

Challenges:

- Key, salient portions of the **Input**
can become “buried”
- Knowing what to pay **attention** to
is difficult

Outline

The Attention Mechanism

Transformer Language Models as General
Language Encoders

Attention

A mechanism for signaling where in the input to focus (“attend to”) when producing some output

Each attention mechanism results in a probability distribution over the input

There are many ways of computing this

Example: Translation

Le chat est sur la chaise.

variable # of
input words

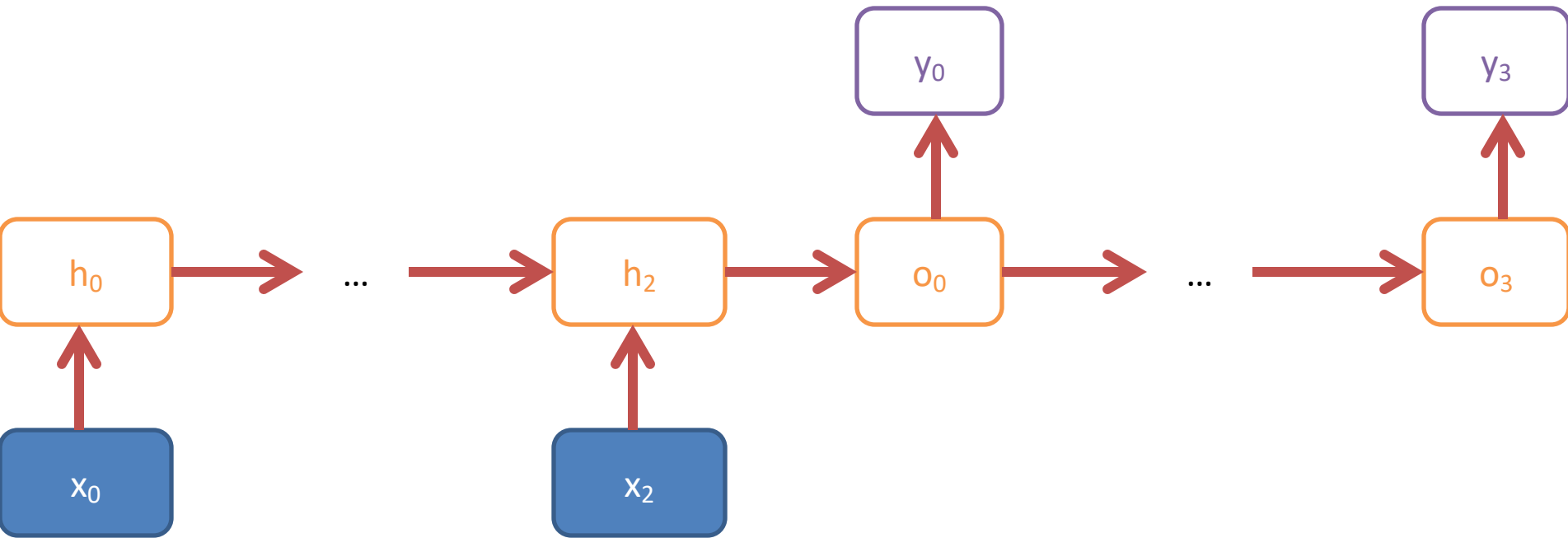


The cat is on the chair.

variable # of
output words

Example: Translation

Le chat est sur la chaise.

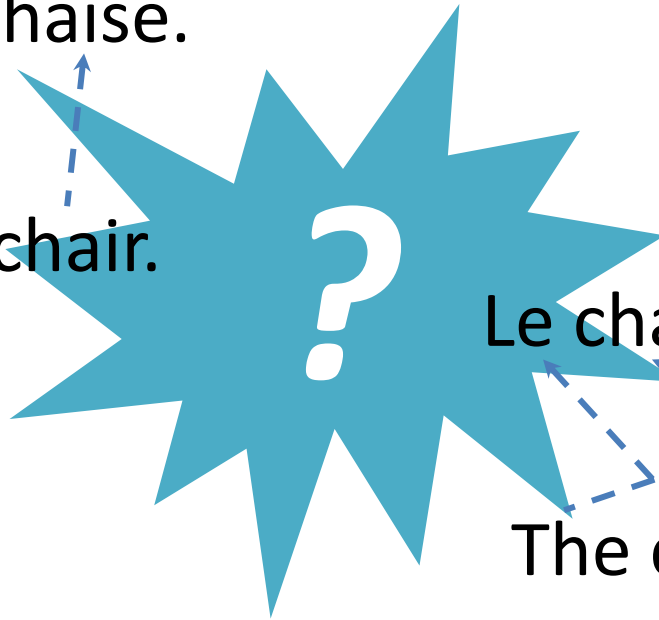


The cat is on the chair.

Attention

Le chat est sur la chaise.

The cat is on the chair.



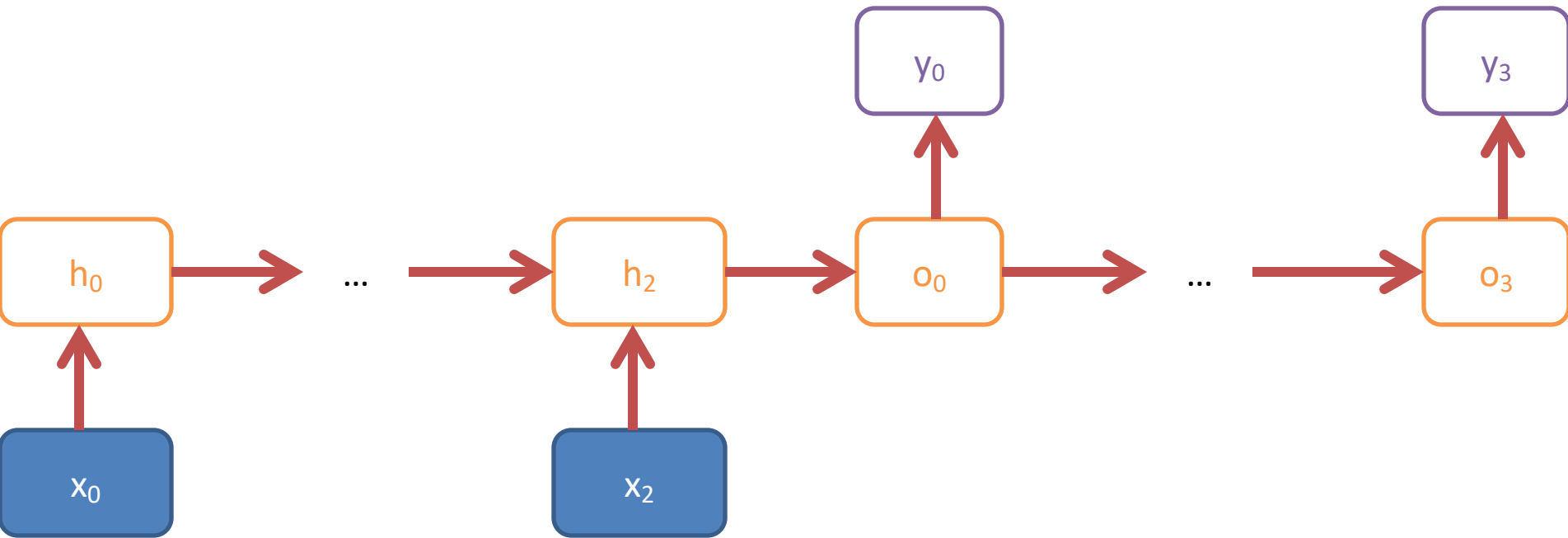
Le chat est sur la chaise.

The cat is on the chair.



Example: Translation

Le chat est sur la chaise.



The cat is on the chair.

Vaswani et al. (NeurIPS, 2017)

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez*[†]
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin*[‡]
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

“Attention Is All You Need”: Take-Aways

1. Formulation of attention as a query-key-value triple
2. “Transformer” model that uses self-attention
3. Demonstration that the transformer can outperform sequence-to-sequence recurrent models (but at a large computational cost!)

“Attention Is All You Need” Description of Attention

“An attention function can be described as mapping a **query** and a set of **key-value** pairs to an **output**, where the query, keys, values, and output are all vectors.

*these **query**, **key**, **value**, and **output** items will be **task dependent***

“Attention Is All You Need” Description of Attention

“An attention function can be described as mapping a **query** and a set of **key-value** pairs to an **output**, where the query, keys, values, and output are all vectors.

The **output** is computed as a weighted sum of the **values**,

“Attention Is All You Need” Description of Attention

“An attention function can be described as mapping a **query** and a set of **key-value** pairs to an **output**, where the query, keys, values, and output are all vectors.

The **output** is computed as a weighted sum of the **values**,

where the weight assigned to each **value** is computed by a compatibility function of the **query** with the corresponding **key**.”

“Attention Is All You Need” Description of Attention

“An attention function can be described as mapping a **query** and a set of **key-value** pairs to an **output**, where the query, keys, values, and output are all vectors.

query: input & current translation

key: English words

value: French words

output: next translated word

Le  chat

[Le] [chat] ... [fromage] ...

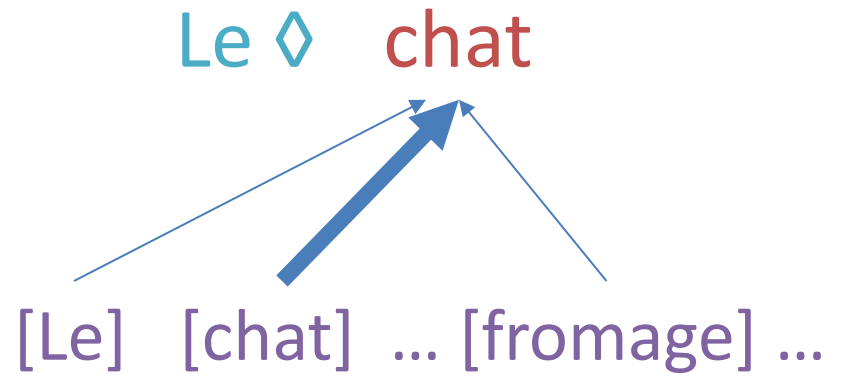
[The] [cat] ... [bandage] ...

The cat is on the chair.

“Attention Is All You Need” Description of Attention

“An attention function can be described as mapping a **query** and a set of **key-value** pairs to an **output**, where the query, keys, values, and output are all vectors.

The **output** is computed as a weighted sum of the **values**,



[The] [cat] ... [bandage] ...

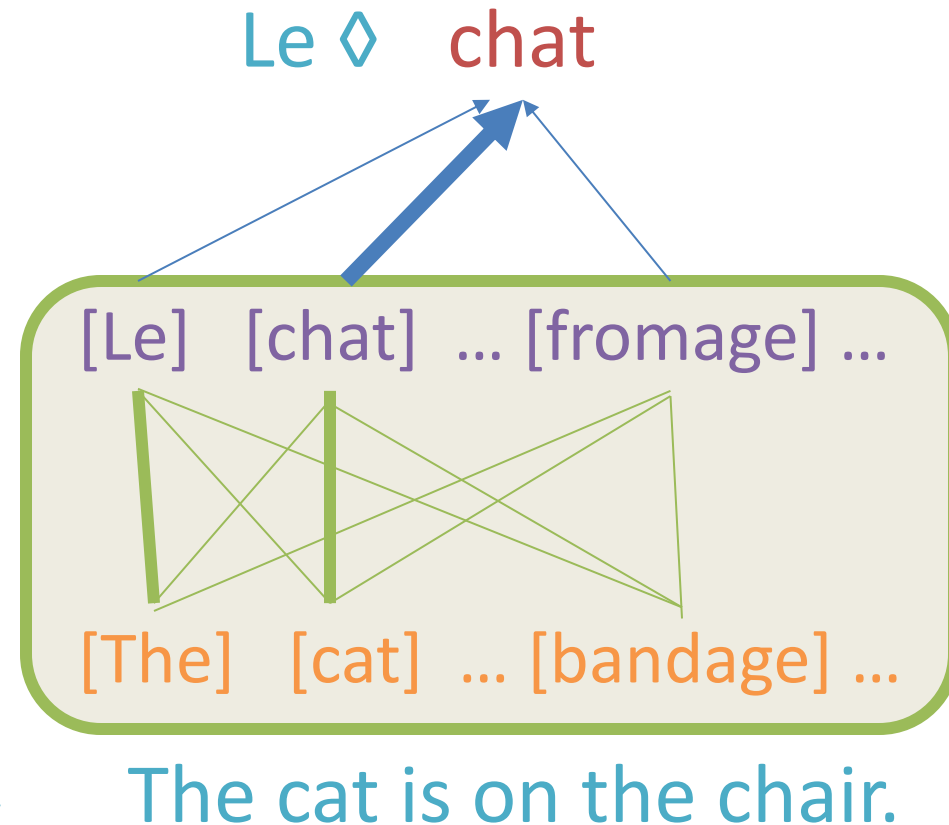
The cat is on the chair.

“Attention Is All You Need” Description of Attention

“An attention function can be described as mapping a **query** and a set of **key-value** pairs to an **output**, where the query, keys, values, and output are all vectors.

The **output** is computed as a weighted sum of the **values**,

where the weight assigned to each **value** is computed by a **compatibility function** of the **query** with the corresponding **key**.”



Outline

The Attention Mechanism

Transformer Language Models as General
Language Encoders

Two Well-Known Instances of Multi-Task Learning in NLP



Collobert and Weston (2008, ICML)



BERT [Devlin et al., 2019 NAACL]

BERT: Pre-training of Deep Bidirectional **Transformers** for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin, mingweichang, kentonl, kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

We argue that current techniques restrict the power of the pre-trained representations, especially for the fine-tuning approaches. The major limitation is that standard language models are

Two Well-Known (Recent) Instances of Learning from Language Models

GPT2 [Radford et al., 2018]

Language Models are Unsupervised Multitask Learners

Alec Radford^{*1} Jeffrey Wu^{*1} Rewon Child¹ David Luan¹ Dario Amodei^{**1} Ilya Sutskever^{**1}

Abstract

Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset - matching or exceeding the performance of 3 out of 4 baseline systems without using the 127,000+ training examples. The capacity of the language model is essential to the success of zero-shot task transfer and increasing it improves performance in a log-linear fashion across tasks. Our largest model, GPT-2, is a 1.5B parameter Transformer that achieves state of the art results on 7 out of 8 tested language modeling datasets in a zero-shot setting but still underfits WebText. Samples from the model reflect these improvements and contain coherent paragraphs of text. These findings suggest a promising path towards building language processing systems which learn to perform tasks from their naturally occurring demonstrations.

competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent and identically distributed (IID) held-out examples. This has served well to make progress on narrow experts. But the often erratic behavior of captioning models (Lake et al., 2017), reading comprehension systems (Jia & Liang, 2017), and image classifiers (Alcorn et al., 2018) on the diversity and variety of possible inputs highlights some of the shortcomings of this approach.

Our suspicion is that the prevalence of single task training on single domain datasets is a major contributor to the lack of generalization observed in current systems. Progress towards robust systems with current architectures is likely to require training and measuring performance on a wide range of domains and tasks. Recently, several benchmarks have been proposed such as GLUE (Wang et al., 2018) and decaNLP (McCann et al., 2018) to begin studying this.

Multitask learning (Caruana, 1997) is a promising framework for improving general performance. However, multitask training in NLP is still nascent. Recent work reports modest performance improvements (Yogatama et al., 2019). In this paper, we propose a new framework for

BERT [Devlin et al., 2019 NAACL]

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language

{jacobdevlin, mingweichang, kentonl, kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **Bidirectional Encoder Representations from Transformers**. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

We argue that current techniques restrict the power of the pre-trained representations, especially for the fine-tuning approaches. The major limitation is that standard language models are

GPT-2 & BERT (and others) In Practice

- Use tensorflow code
- Use pytorch code
 - The huggingface transformers package is very popular

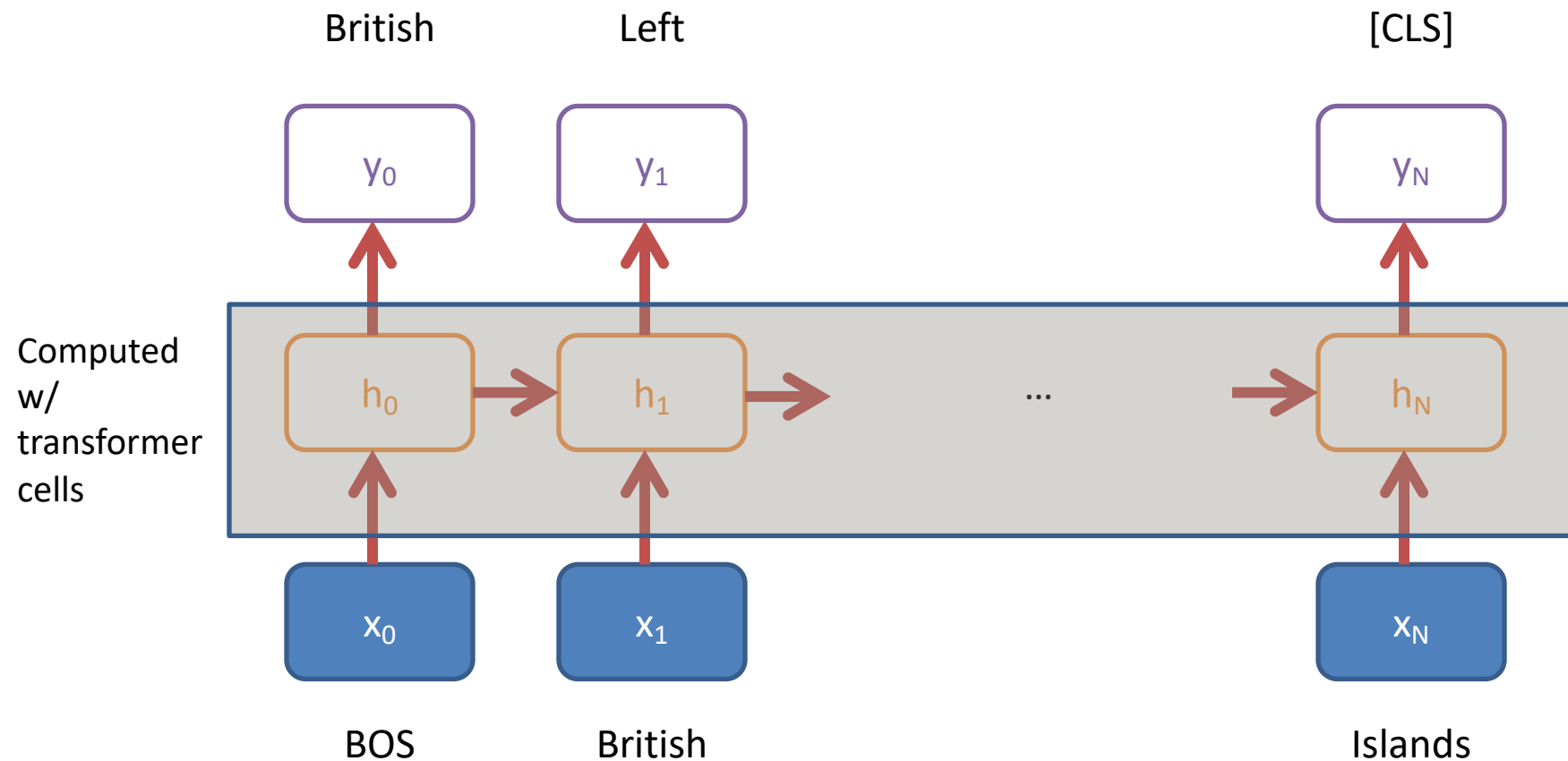
GPT-2 Take-Away

Language models can provide an effective way of learning embeddings that are useful for downstream tasks

- Auto-regressive model that uses a **transformer cell**

$$p(w_1, w_N) = \prod_i p(w_i | w_1, \dots, w_{i-1})$$

GPT-2 Model & Representation



BERT Take-Aways

1. Demonstration of bidirectional transformer for language understanding
2. Clean separation of “pre-training” and “fine-tuning” tasks
3. Clear demonstration that language model “pre-training” can yield useful embeddings

Pretraining vs. Fine-tuning

Pre-training

Learning an **encoder** to produce effective embeddings through “general” training objectives that are end-task agnostic

Pretraining vs. Fine-tuning

Pre-training

Learning an **encoder** to produce effective embeddings through “general” training objectives that are end-task agnostic

1. Next-sentence prediction
[NSP]
2. Masked Language Modeling
[MLM]

Pre-training: NSP

- Given two sentences s_1 and s_2 , predict whether s_2 follows s_1 in “natural” text

Pretraining vs. Fine-tuning

Pre-training

Learning an **encoder** to produce effective embeddings through “general” training objectives that are end-task agnostic

1. Next-sentence prediction
[NSP]
2. Masked Language Modeling
[MLM]

Pre-training: MLM

- Given a sentence $s = w_1 \dots w_N$, mask out (remove) a word w_i and predict what that word should be

“The cat chased the mouse” →
“The cat [MASK] the mouse”

$$p(w \mid \text{The cat [MASK]the mouse})$$

Pretraining vs. Fine-tuning

Pre-training

Learning an **encoder** to produce effective embeddings through “general” training objectives that are end-task agnostic

1. Next-sentence prediction [NSP]
2. Masked Language Modeling [MLM]

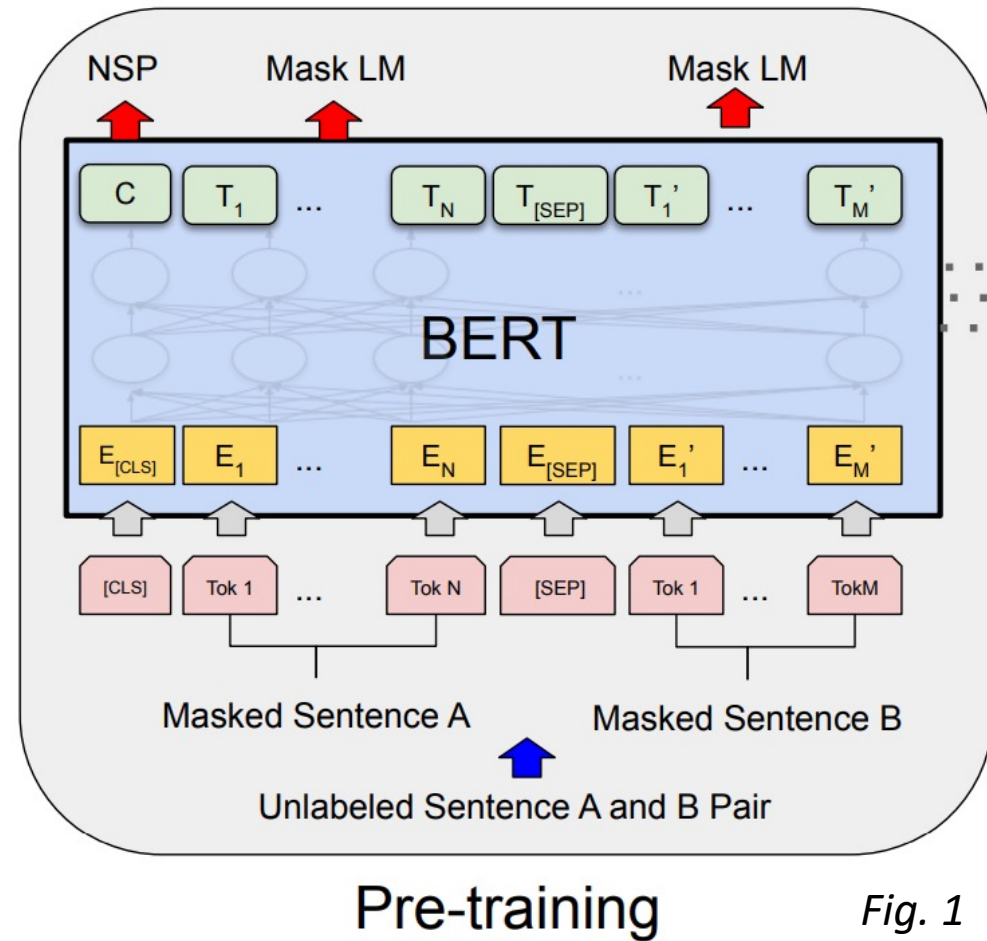


Fig. 1

Pretraining vs. Fine-tuning

Pre-training

Learning an **encoder** to produce effective embeddings through “general” training objectives that are end-task agnostic

1. Next-sentence prediction
[NSP]
2. Masked Language Modeling
[MLM]

Fine-Tuning

Learning task-specific **decoders** using the embeddings produced from the pre-training, e.g.,

- RTE
- Question-answering
- <Your task here>

Pretraining vs. Fine-tuning

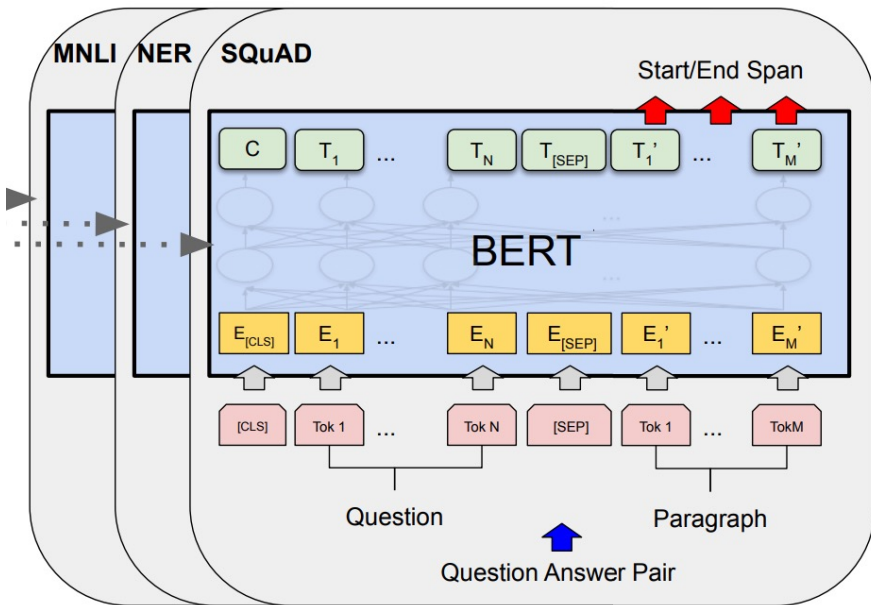


Fig. 1

Fine-Tuning

Fine-Tuning

Learning task-specific **decoders** using the embeddings produced from the pre-training, e.g.,

- RTE
- Question-answering
- <Your task here>

Pre-training *then* Fine-tuning

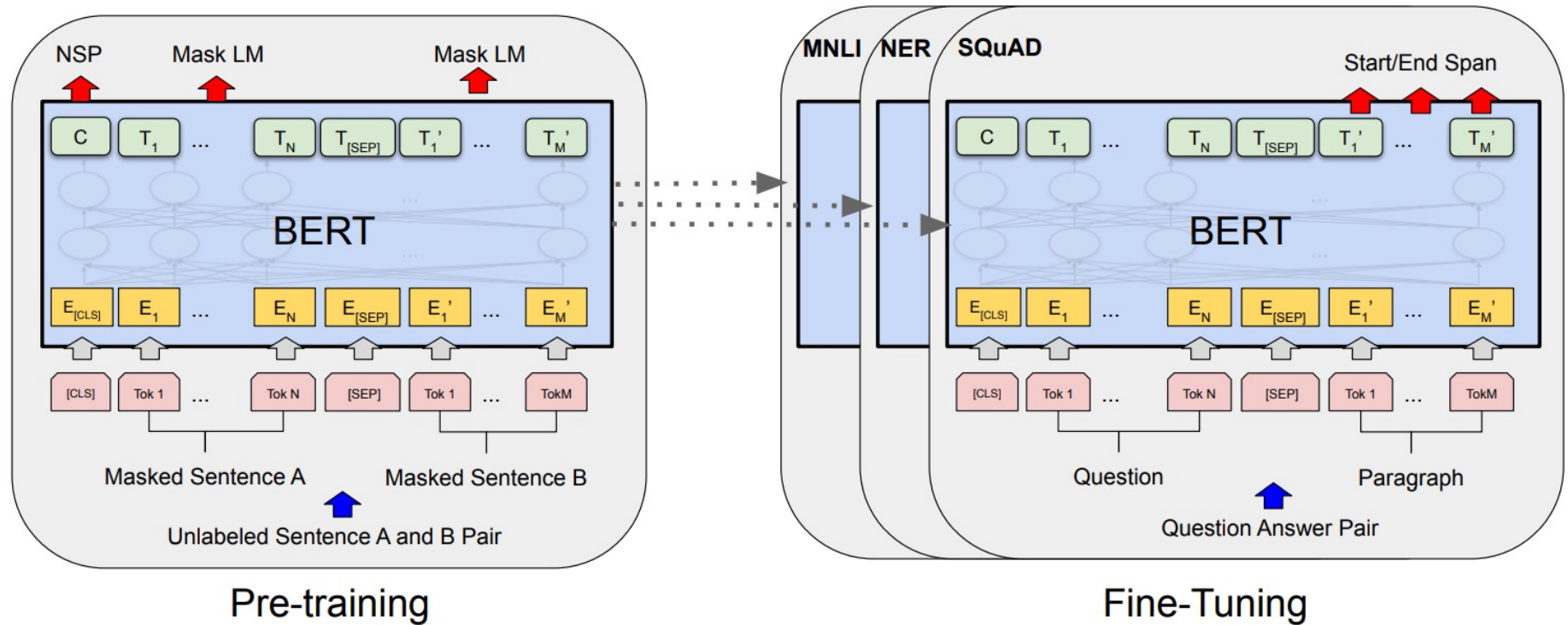
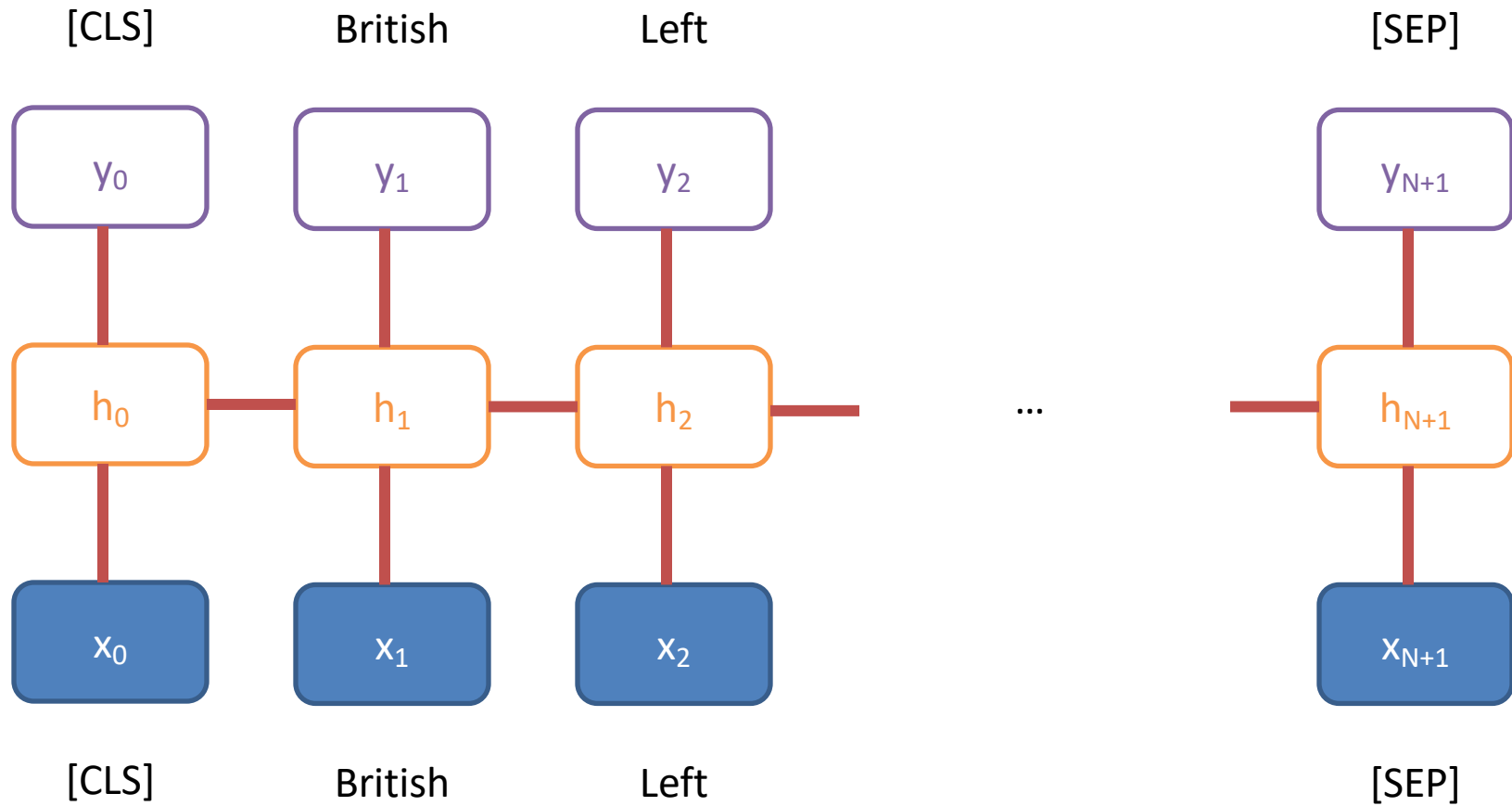


Fig. 1

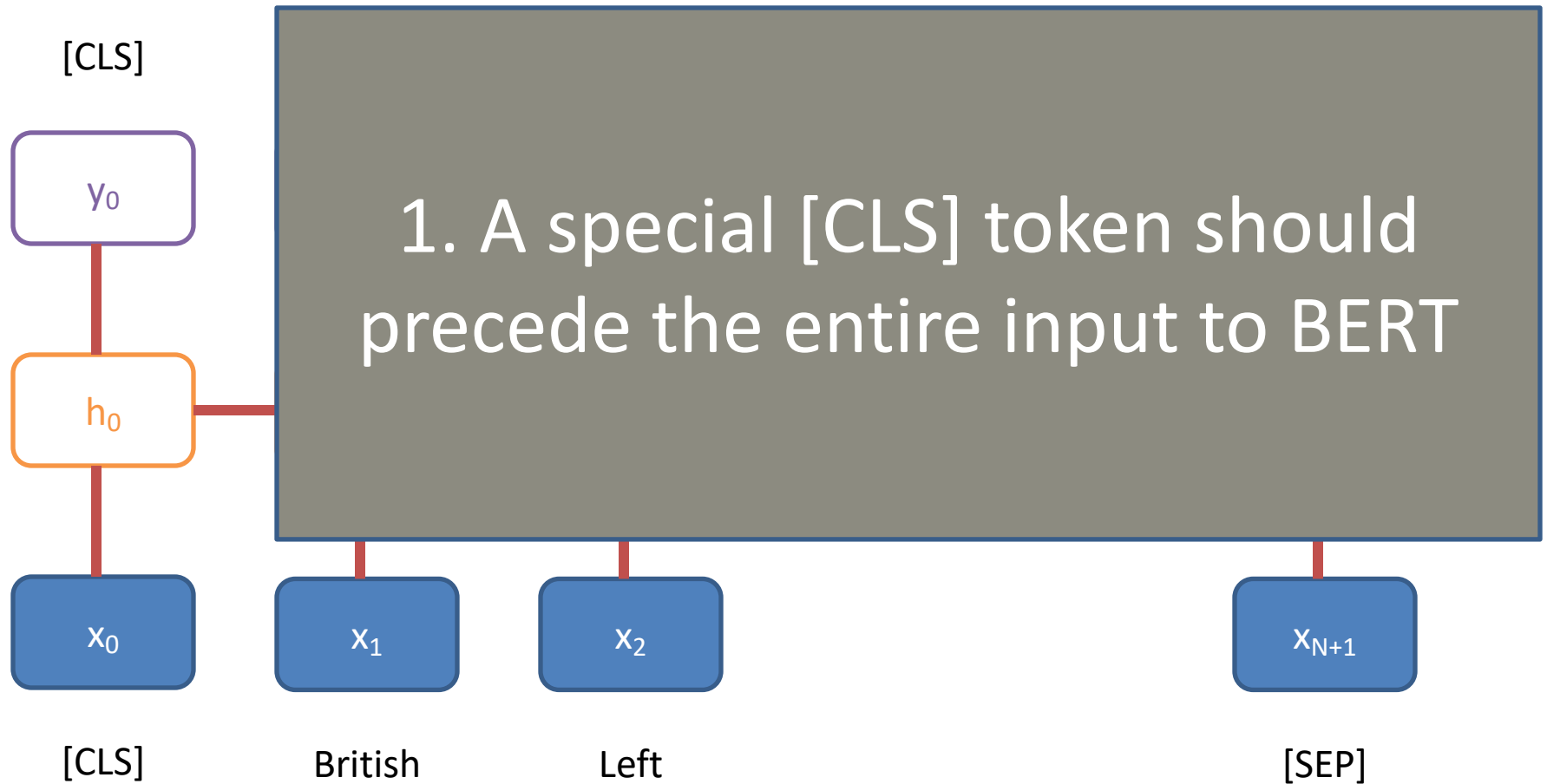
BERT Representation

1. A special [CLS] token should precede the entire input to BERT
2. Every sentence should be followed by a special [SEP] token
3. The input must be tokenized in a special way
4. Segment & *position* embeddings must be provided

BERT Representation

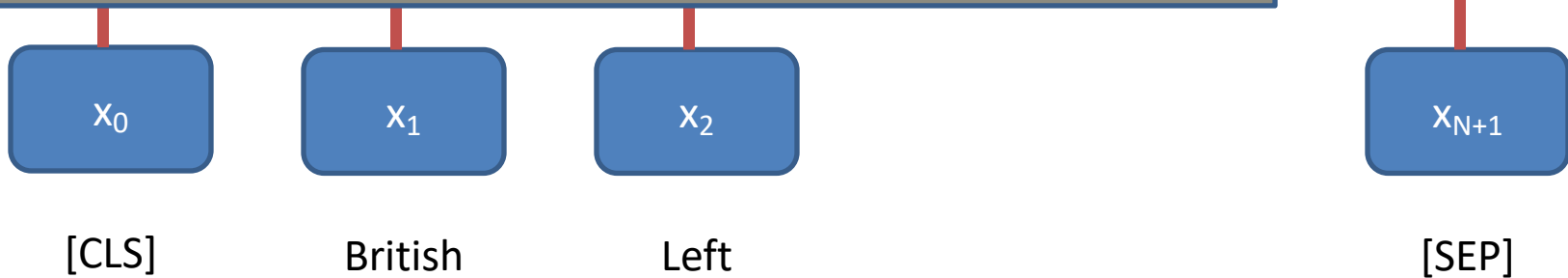


BERT Representation



BERT Representation

2. Every sentence should be followed by a special [SEP] token



BERT Representation (Even More)

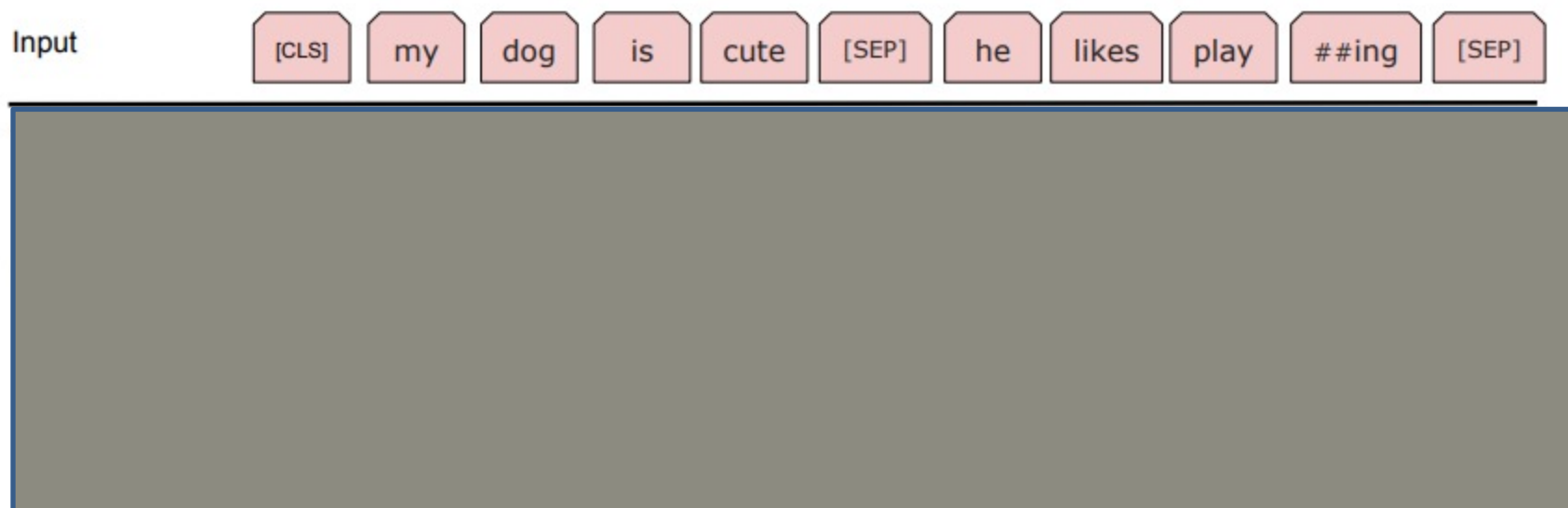
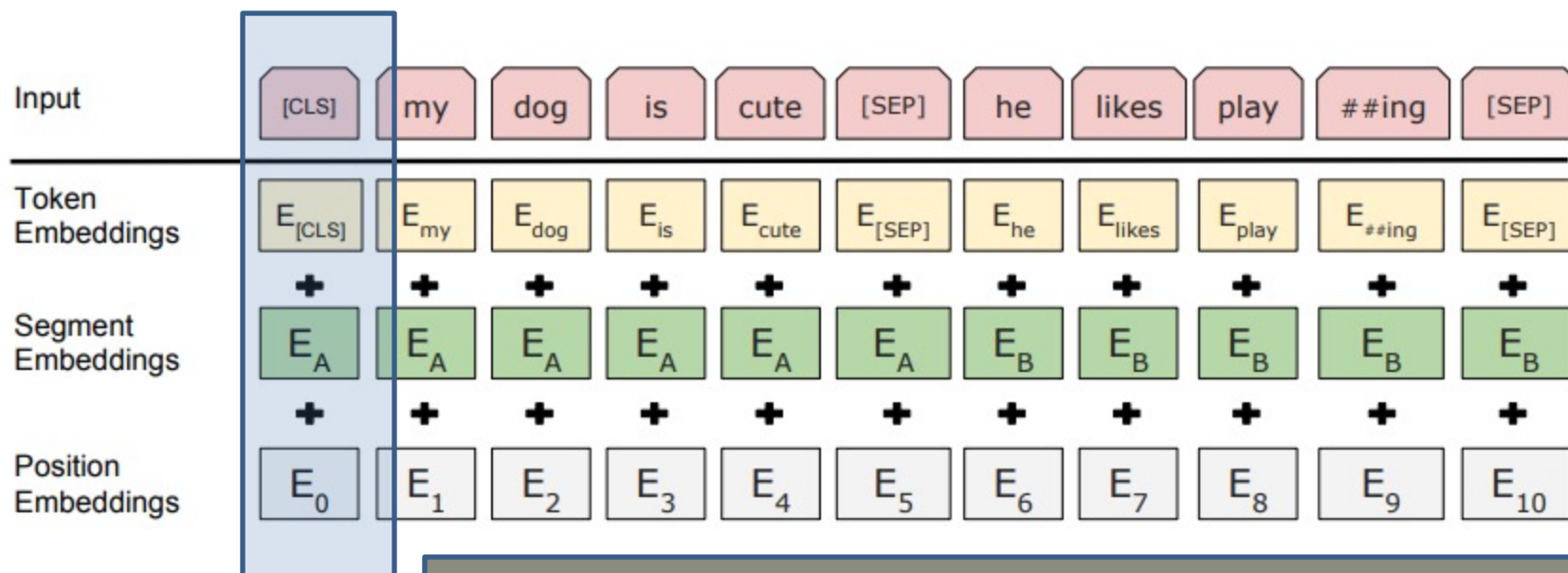


Fig. 2

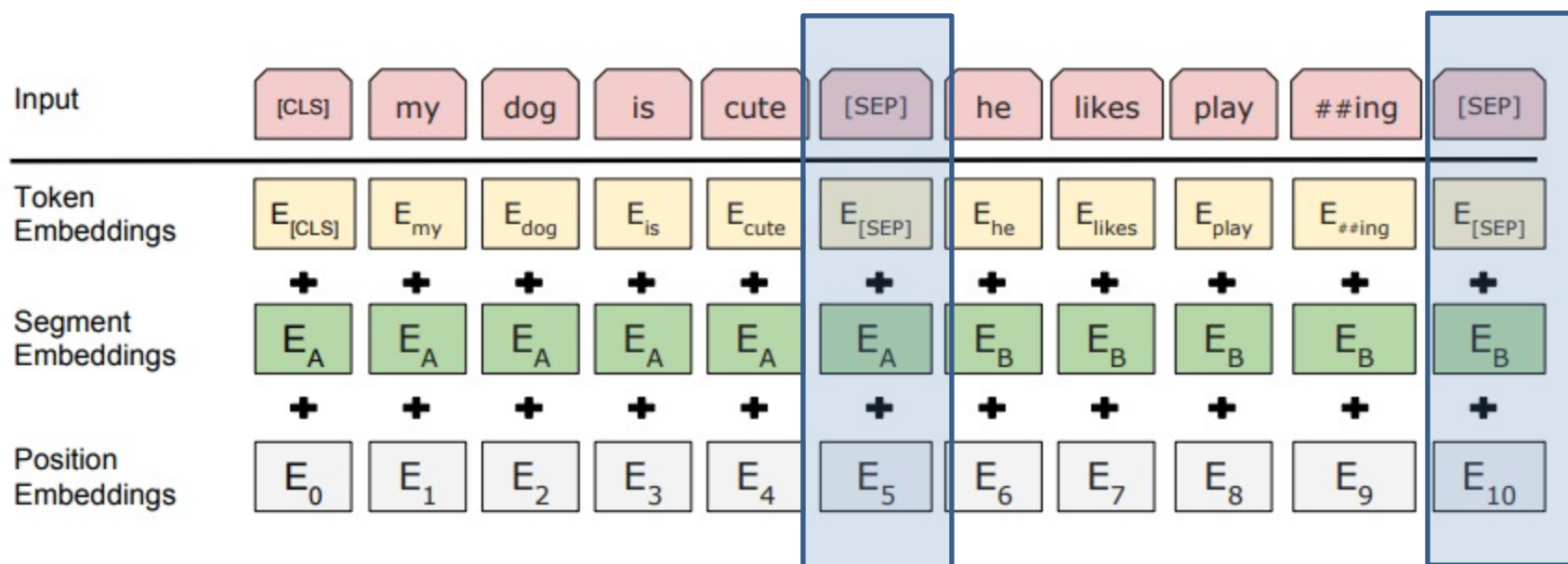
BERT Representation (Even More)



1. A special [CLS] token should precede the entire input to BERT

Fig. 2

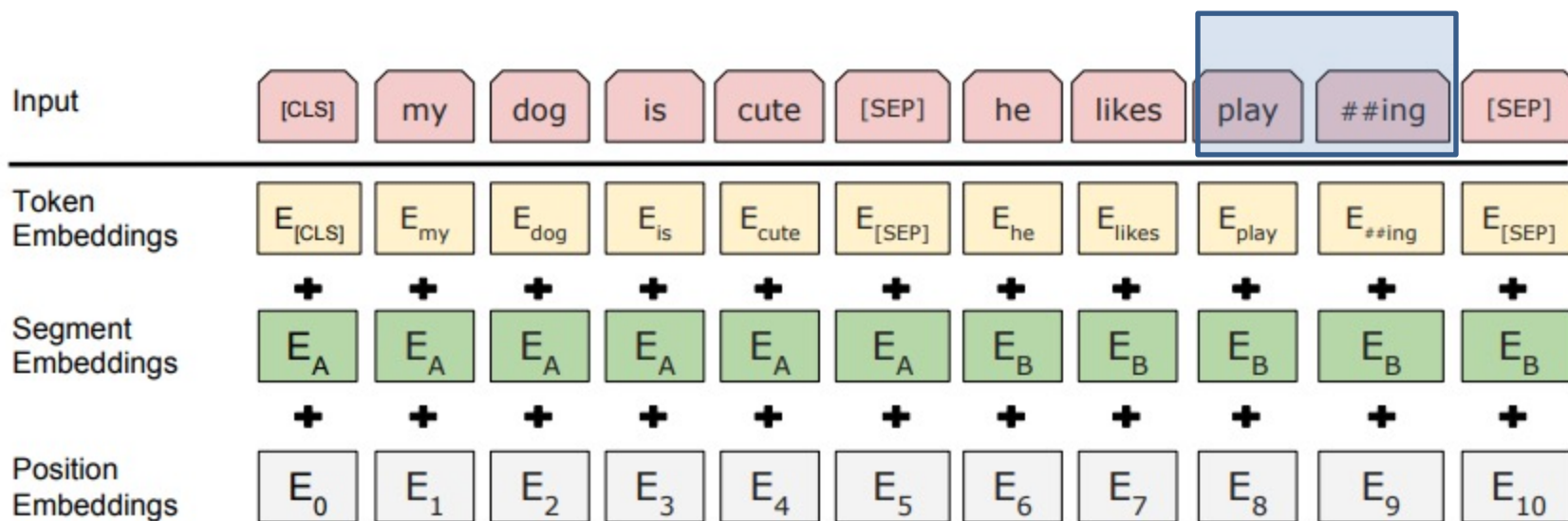
BERT Representation (Even More)



2. Every sentence should be followed by a special [SEP] token

Fig. 2

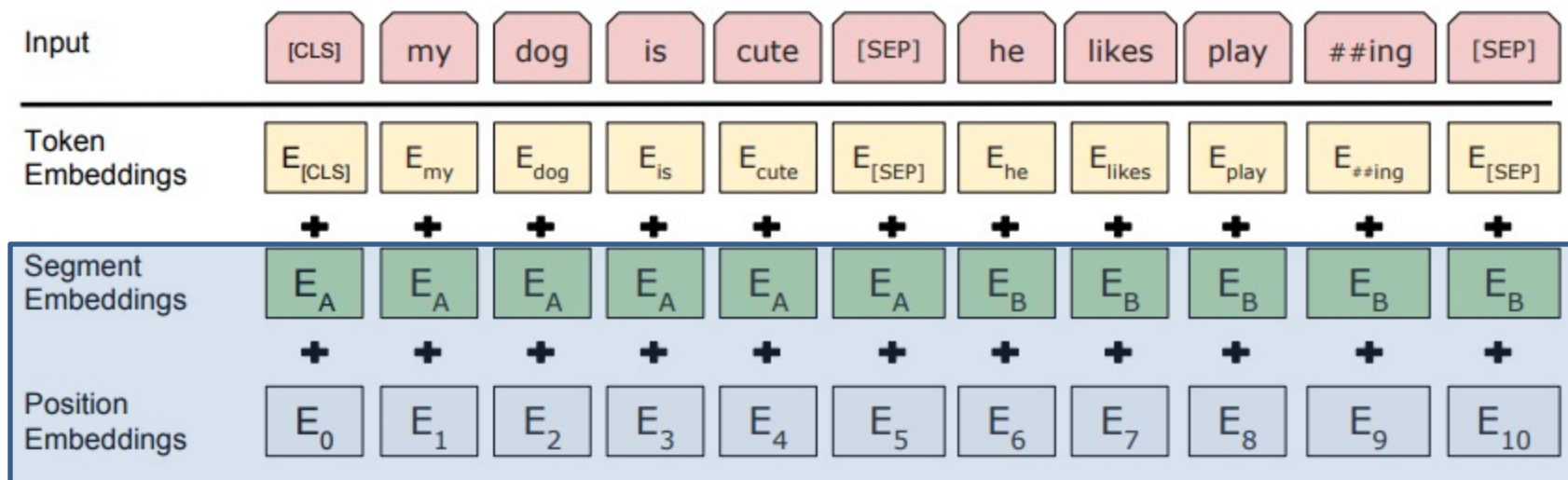
BERT Representation (Even More)



3. The input must be tokenized in a special way

Fig. 2

BERT Representation (Even More)



4. Segment & *position* embeddings must be provided

Fig. 2

Transformer Language Model Take-Aways

1. Clean separation of “pre-training” and “fine-tuning” tasks
2. Clear demonstration that language model “pre-training” can yield useful embeddings

GPT-2 & BERT (and others) In Practice

- Use tensorflow code
- Use pytorch code
 - The **huggingface transformers** package is very popular

Transformers	
Masked $[x_{1:N \setminus n} \Rightarrow x_n]$	
BERT	(Devlin et al., 2018)
RoBERTa	(Liu et al., 2019a)
Autoregressive $[x_{1:n-1} \Rightarrow x_n]$	
GPT / GPT-2	(Radford et al., 2019)
Trans-XL	(Dai et al., 2019)
XLNet	(Yang et al., 2019)
Seq-to-Seq $[\sim x_{1:N} \Rightarrow x_{1:N}]$	
BART	(Lewis et al., 2019)
T5	(Raffel et al., 2019)
MarianMT	(J.-Dowmunt et al., 2018)
Specialty: Multimodal	
MMBT	(Kiela et al., 2019)
Specialty: Long-Distance	
Reformer	(Kitaev et al., 2020)
Longformer	(Beltagy et al., 2020)
Specialty: Efficient	
ALBERT	(Lan et al., 2019)
Electra	(Clark et al., 2020)
DistilBERT	(Sanh et al., 2019)
Specialty: Multilingual	
XLNet/RoBERTa	(Lample and Conneau, 2019b)

Fig. 2 (Wolf et al., 2020: <https://arxiv.org/pdf/1910.03771.pdf>)

GPT-2 & BERT (and others) In Practice

- Use tensorflow code
- Use pytorch code
 - The **huggingface transformers** package is very popular

Name	Input	Heads Output	Tasks	Ex. Datasets
Language Modeling	$x_{1:n-1}$	$x_n \in \mathcal{V}$	Generation	WikiText-103
Sequence Classification	$x_{1:N}$	$y \in \mathcal{C}$	Classification, Sentiment Analysis	GLUE, SST, MNLI
Question Answering	$x_{1:M}, x_{M:N}$	$y \text{ span } [1 : N]$	QA, Reading Comprehension	SQuAD, Natural Questions
Token Classification	$x_{1:N}$	$y_{1:N} \in \mathcal{C}^N$	NER, Tagging	OntoNotes, WNUT
Multiple Choice	$x_{1:N}, \mathcal{X}$	$y \in \mathcal{X}$	Text Selection	SWAG, ARC
Masked LM	$x_{1:N \setminus n}$	$x_n \in \mathcal{V}$	Pretraining	Wikitext, C4
Conditional Generation	$x_{1:N}$	$y_{1:M} \in \mathcal{V}^M$	Translation, Summarization	WMT, IWSLT, CNN/DM, XSum

Fig. 2 (Wolf et al., 2020:
<https://arxiv.org/pdf/1910.03771.pdf>)

Outline

The Attention Mechanism

Transformer Language Models as General
Language Encoders