Sequence labeling with Transfer Learning

การกำกับข้อมูลลำดับด้วยการเรียนรู้แบบถ่ายโอน

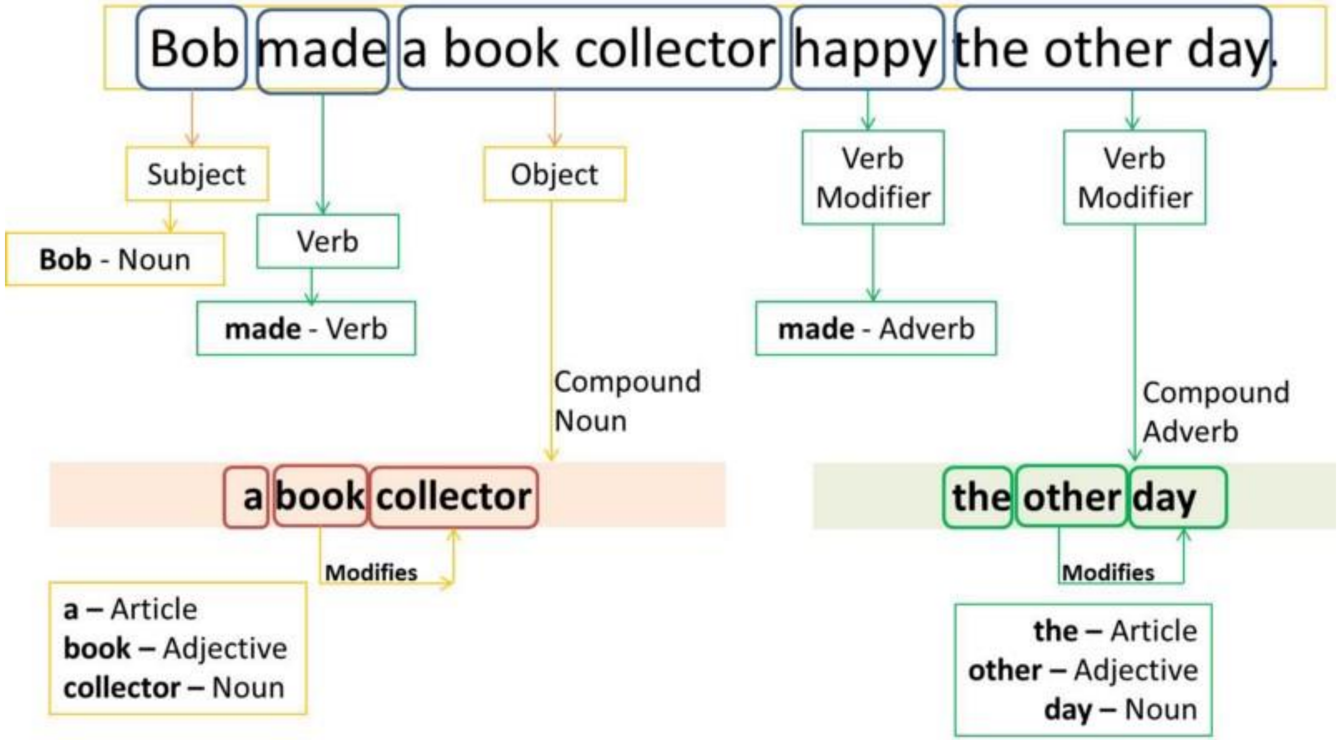
Pongsathon Janyoi, Ph.D.
College of Computing, Khon Kaen University

Outline

01. Introduction to Sequence labeling
02. Part of Speech Tagging
03. Named Entities Recognition
04. Traditional approaches
05. Neural approaches

Sequence labeling

- Identifying a categorical label to each token (e.g. word, phrase) of a sequence



Sequence labeling

- Main Tasks of Sequence Labeling:
  - Part-of-speech (POS) Tagging
  - Named-entity recognition
  - Chunking

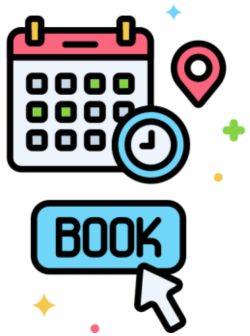
Past-of-speech Tagging

Part of Speech Tagging

- Words can be classified into grammatical categories
- Part of speech, Word classes, POS, POS tags
- 8 parts of speech attributed to Dionysius Thrax of Alexandria (c. 1st C. BCE):
  - noun, verb, pronoun, preposition, adverb, conjunction, participle, article
- Many NLP tasks require POS:
  - Machine Translation, Grammar checking, Summarization

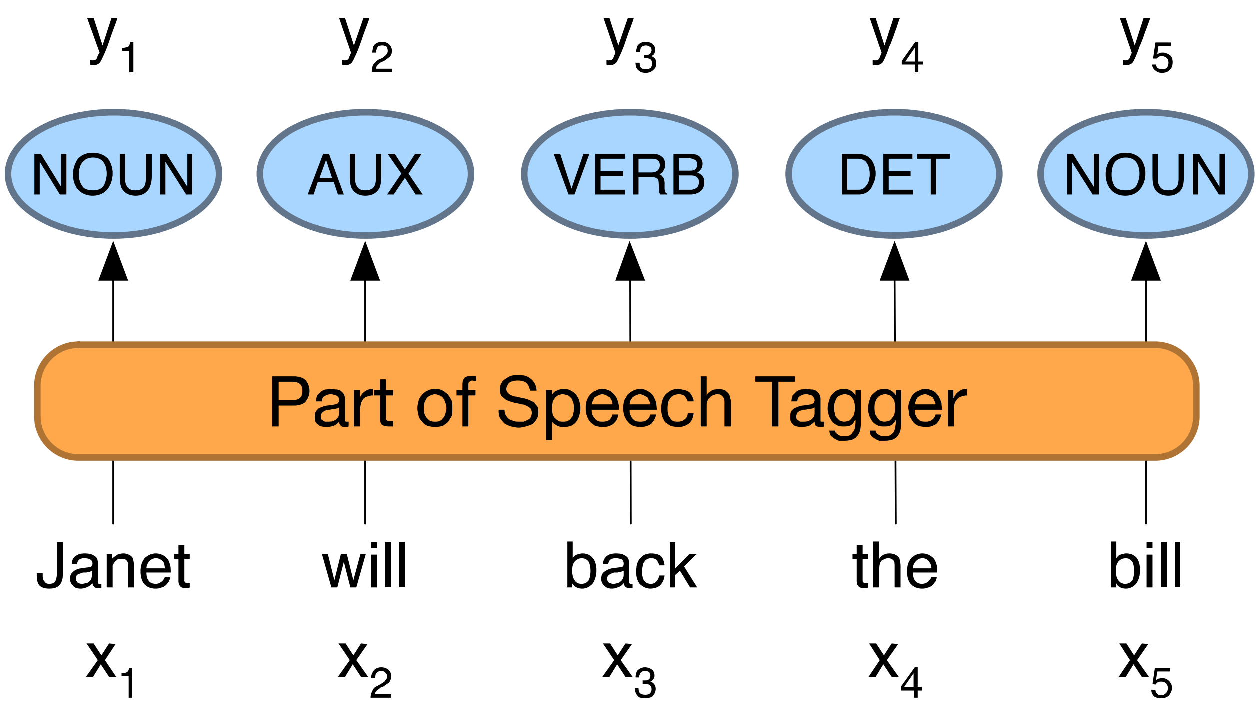
Part-of-Speech Tagging

- Assigning a part-of-speech to each word in a text.
- Words often have more than one POS.
- Book:
  - VERB: (*Book* that flight)
  - NOUN: (Hand me that *book*).



Part-of-Speech Tagging

- Map from sequence $x_1,\ldots,x_n$ of words to $y_1,\ldots,y_n$ of POS tags



"Universal Dependencies" Tagset

	Tag	Description	Example
Open Class	**ADJ**	Adjective: noun modifiers describing properties	*red*, *young*, *awesome*
	ADV	Adverb: verb modifiers of time, place, manner	*very*, *slowly*, *home*, *yesterday*
	NOUN	words for persons, places, things, etc.	*algorithm*, *cat*, *mango*, *beauty*
	VERB	words for actions and processes	*draw*, *provide*, *go*
	PROPN	Proper noun: name of a person, organization, place, etc..	*Regina*, *IBM*, *Colorado*
	INTJ	Interjection: exclamation, greeting, yes/no response, etc.	*oh*, *um*, *yes*, *hello*
Closed Class Words	**ADP**	Adposition (Preposition/Postposition): marks a noun's spacial, temporal, or other relation	*in, on, by under*
	AUX	Auxiliary: helping verb marking tense, aspect, mood, etc.,	*can, may, should, are*
	CCONJ	Coordinating Conjunction: joins two phrases/clauses	*and*, *or*, *but*
	DET	Determiner: marks noun phrase properties	*a, an, the, this*
	NUM	Numeral	*one, two, first, second*
	PART	Particle: a preposition-like form used together with a verb	*up, down, on, off, in, out, at, by*
	PRON	Pronoun: a shorthand for referring to an entity or event	*she, who, I, others*
	SCONJ	Subordinating Conjunction: joins a main clause with a subordinate clause such as a sentential complement	*that*, *which*
Other	**PUNCT**	Punctuation	; , ()
	SYM	Symbols like $ or emoji	$, %
	X	Other	asdf, qwfg

Nivre et al. 2016

Sample "Tagged" English sentences

- There/PRO were/VERB 70/NUM children/NOUN there/ADV ./PUNC
- Preliminary/ADJ findings/NOUN were/AUX reported/VERB in/ADP today/NOUN ’s/PART New/PROPN England/PROPN Journal/PROPN of/ADP Medicine/PROPN

How difficult is POS tagging in English?

- E.g., *back*
  - earnings growth took a back/ADJ seat
  - a small building in the back/NOUN
  - a clear majority of senators back/VERB the bill
  - enable the country to buy back/PART debt
  - I was twenty-one back/ADV then

Named Entity tagging

- The task of named entity recognition (NER):
  - find spans of text that constitute proper names
  - tag the type of the entity.

Citing high fuel prices, [$_{ORG}$ **United Airlines**] said [$_{TIME}$ **Friday**] it has increased fares by [$_{MONEY}$ **$6**] per round trip on flights to some cities also served by lower-cost carriers. [$_{ORG}$ **American Airlines**], a unit of [$_{ORG}$ **AMR Corp.**], immediately matched the move, spokesman [$_{PER}$ **Tim Wagner**] said. [$_{ORG}$ **United**], a unit of [$_{ORG}$ **UAL Corp.**], said the increase took effect [$_{TIME}$ **Thursday**] and applies to most routes where it competes against discount carriers, such as [$_{LOC}$ **Chicago**] to [$_{LOC}$ **Dallas**] and [$_{LOC}$ **Denver**] to [$_{LOC}$ **San Francisco**].

Named Entity tagging

ลุงตู่ต่อว่าผู้สื่อข่าวที่ตึกไทยคู่ฟ้าเมื่อเช้า

ลุง	ตู่	ต่อว่า	ผู้สื่อข่าว	ที่	ตึกไทยคู่ฟ้า	เมื่อ	เช้า
Noun	PNoun	Verb	Noun	Adj	PNoun	ADP	Noun
B-PER	I-PER	O	O	O	B-PLACE	B-TIME	I-TIME

https://www.youtube.com/watch?v=LgQwsosz-e8

LST20 Corpus

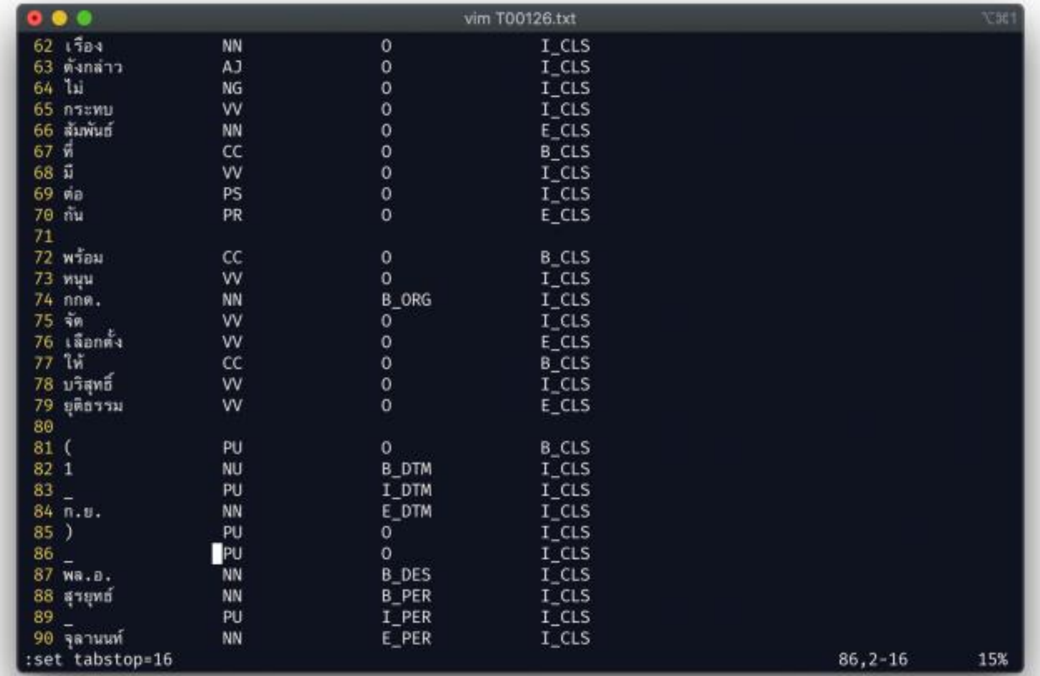
- LST20 Corpus
  - Dataset for training fundamental Thai language processing tasks
  - Featured linguistic information
    - **Word boundaries** for word segmentation
    - **Named entities** for named entity recognition
    - **Clause boundaries** for clause segmentation
    - **Sentence boundaries** for sentence segmentation
    - **News genres** for document classification
  - CoNLL-2003 format: tab-separated columns

Words	3,163,034
Named entities	288,020
Clauses	248,181
Sentences	74,180
Distinct words	46,692
Genres	15
News articles	3,745

Available at
https://aiforthai.in.th

LST20 Corpus

- Format: CoNLL-2003 Style



vim T00126.txt

```
62 เรื่อง	NN	O	I_CLS
63 ดังกล่าว	AJ	O	I_CLS
64 ไม่	NG	O	I_CLS
65 กระทบ	VV	O	I_CLS
66 สัมพันธ์	NN	O	E_CLS
67 ที่	CC	O	B_CLS
68 มี	VV	O	I_CLS
69 ต่อ	PS	O	I_CLS
70 กัน	PR	O	E_CLS
71
72 พร้อม	CC	O	B_CLS
73 หนุน	VV	O	I_CLS
74 กกต.	NN	B_ORG	I_CLS
75 จัด	VV	O	I_CLS
76 เลือกตั้ง	VV	O	E_CLS
77 ให้	CC	O	B_CLS
78 บริสุทธิ์	VV	O	I_CLS
79 ยุติธรรม	VV	O	E_CLS
80
81 (	PU	O	B_CLS
82 1	NU	B_DTM	I_CLS
83 _	PU	I_DTM	I_CLS
84 ก.ย.	NN	E_DTM	I_CLS
85 )	PU	O	I_CLS
86 _	PU	O	I_CLS
87 พล.อ.	NN	B_DES	I_CLS
88 สุรยุทธ์	NN	B_PER	I_CLS
89 _	PU	I_PER	I_CLS
90 จุลานนท์	NN	E_PER	I_CLS
:set tabstop=16                    86,2-16        15%
```

- Four columns
  - Word
  - POS tag
  - Named entity
  - Clause boundary
- Notes
  - Each column is separated by a tab
  - Empty line marks sentence boundary

LST20 Corpus

- POS Tagset

Tags	Names	Brief Descriptions
AJ	Adjective	Attribute, modifier, or description of a noun
AV	Adverb	Word that modifies or qualifies an adjective, verb, or another adverb
AX	Auxiliary	Tense, aspect, mood, and voice
CC	Connector	Conjunction and relative pronoun
CL	Classifier	Class or measurement unit to which a noun or an action belongs
FX	Prefix	Inflectional (nominalizer, adjectivizer, adverbializer, and courteous verbalizer), and derivational
IJ	Interjection	Exclamation word
NG	Negator	Word of negation

Tags	Names	Brief Descriptions
NN	Noun	Person, place, thing, abstract concept, and proper name
NU	Number	Quantity for counting and calculation
PA	Particle	Politeness, intention, belief, question
PR	Pronoun	Word used to refer to an element in the discourse
PS	Preposition	Location, comparison, instrument, exemplification
PU	Punctuation	Punctuation mark
VV	Verb	Action, state, occurrence, and word that forms the predicate part
XX	Others	Unknown category

* Green texts = content word | **Black texts = function word** | Red texts = undesirable (yet they still exist)

Named Entity Recognition (NER)

Named Entities

- Named entity, in its core usage, means anything that can be referred to with a proper name.

- Most common 4 tags:
  - **PER** (Person): “Anna”
  - **LOC** (Location): “Khon Kaen City”
  - **ORG** (Organization): “Khon Kaen University”
  - **GPE** (Geo-Political Entity): “Ban Non Muang, Khon Kaen"

Why is NER an Important?

- **Sentiment analysis:** consumer's sentiment toward a particular company or person?
- **Question Answering:** answer questions about an entity?
- **Information Extraction:** Extracting facts about entities from text.

Why NER is hard ?

- **Segmentation**
  - In POS tagging, no segmentation problem since each word gets one tag.
  - In NER we have to find and segment the entities!
- Type ambiguity

[$_{\text{PER}}$ Washington] was born into slavery on the farm of James Burroughs.
[$_{\text{ORG}}$ Washington] went up 2 games to 1 in the four-game series.
Blair arrived in [$_{\text{LOC}}$ Washington] for what may well be his last state visit.
In June, [$_{\text{GPE}}$ Washington] passed a primary seatbelt law.

BIO Tagging

- How can we turn this structured problem into a sequence problem like POS tagging, with one label per word?
- [PER Jane Villanueva] of [ORG United] , a unit of [ORG United Airlines Holding] , said the fare applies to the [LOC Chicago] route.

BIO Tagging

- [PER Jane Villanueva] of [ORG United] , a unit of [ORG United Airlines Holding] , said the fare applies to the [LOC Chicago] route.

Words	**BIO Label**
Jane	B-PER
Villanueva	I-PER
of	O
United	B-ORG
Airlines	I-ORG
Holding	I-ORG
discussed	O
the	O
Chicago	B-LOC
route	O
.	O

BIO Tagging variants: IO and BIOES

- [PER Jane Villanueva] of [ORG United] , a unit of [ORG United Airlines Holding] , said the fare applies to the [LOC Chicago] route.

Words	IO Label	BIO Label	BIOES Label
Jane	I-PER	B-PER	B-PER
Villanueva	I-PER	I-PER	E-PER
of	O	O	O
United	I-ORG	B-ORG	B-ORG
Airlines	I-ORG	I-ORG	I-ORG
Holding	I-ORG	I-ORG	E-ORG
discussed	O	O	O
the	O	O	O
Chicago	I-LOC	B-LOC	S-LOC
route	O	O	O
.	O	O	O

How to tag POS or NE?

Approach for sequence modeling

- Supervised Machine Learning given a human-labeled training set of text annotated with tags
- Statistical based:
  - Hidden Markov Models (HMM)
  - Conditional Random Fields (CRF)
- Neural based:
  - Neural sequence models
  - Large Language Models (like BERT), finetuned

Traditional approaches

Hidden Markov Models

- HMM is based on augmenting the Markov chain.
- A Markov chain is a model predicting the probabilities of sequences of random variables

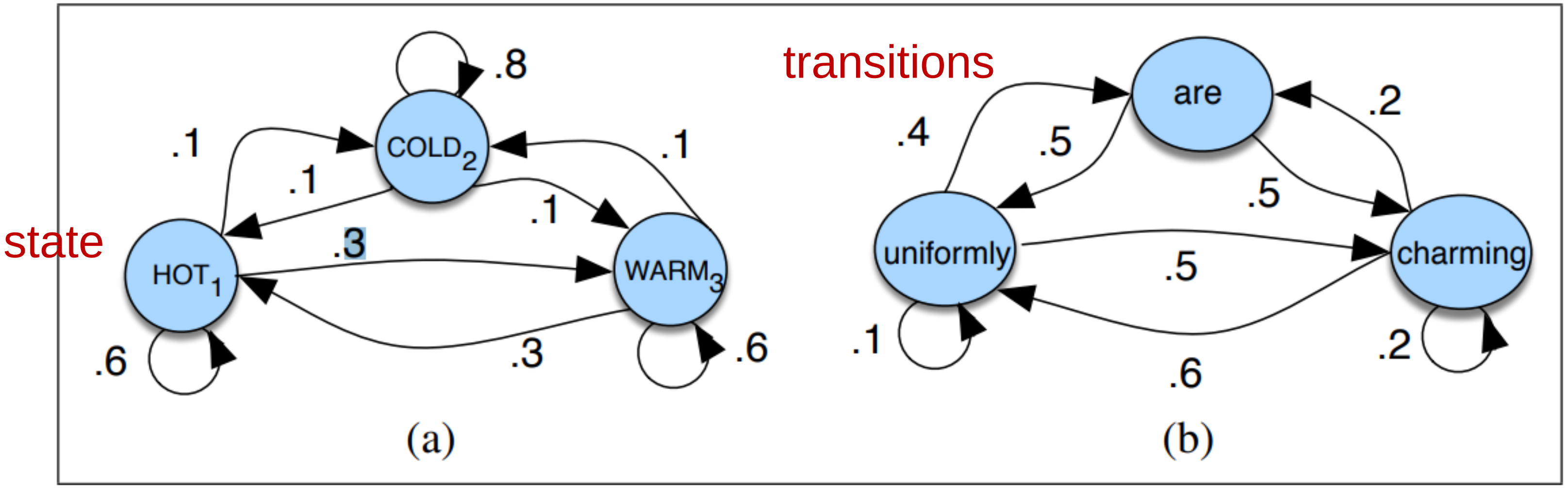


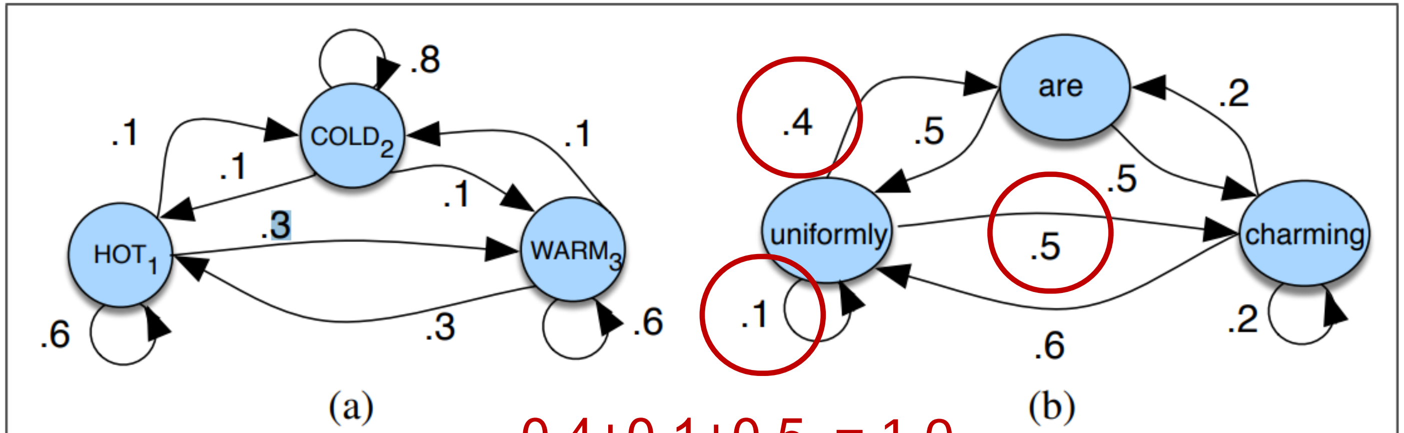
Figure 8.8 A Markov chain for weather (a) and one for words (b), showing states and transitions. A start distribution π is required; setting $\pi = [0.1, 0.7, 0.2]$ for (a) would mean a probability 0.7 of starting in state 2 (cold), probability 0.1 of starting in state 1 (hot), etc.

Hidden Markov Models

- Markov assumption:
  - When predicting the future, the past doesn't matter, only the present.

$$P(q_i = a | q_1 \ldots q_{i-1}) = P(q_i = a | q_{i-1})$$

Where, q1,q2,...,qi is a sequence of state variables

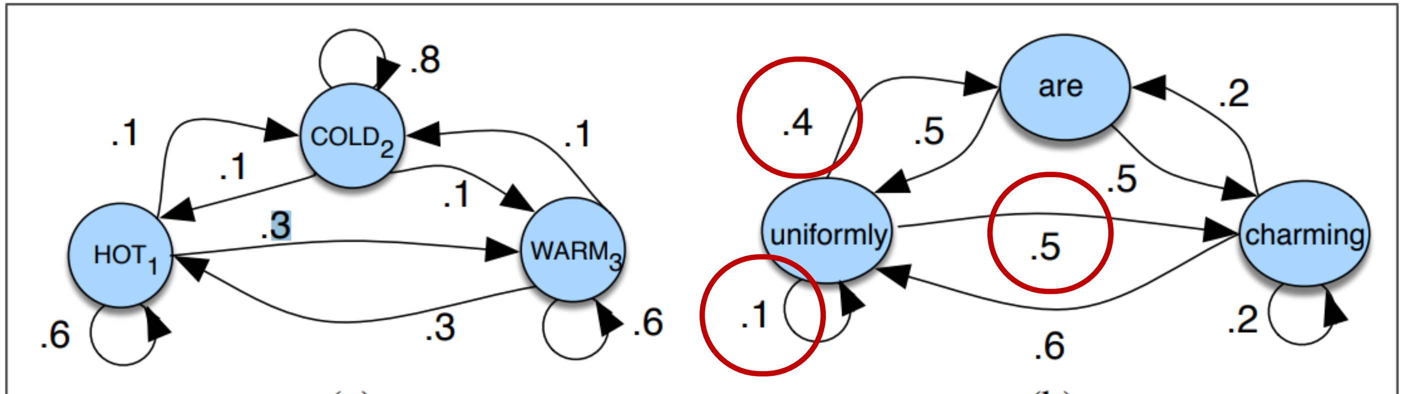


0.4+0.1+0.5 = 1.0

Hidden Markov Models

- **Markov chain** is specified by the following components:

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{N1} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$



(a) (b)

a bigram language model , edge expressing the probability p(wi |wj)

0.4+0.1+0.5 = 1.0

Hidden Markov Models

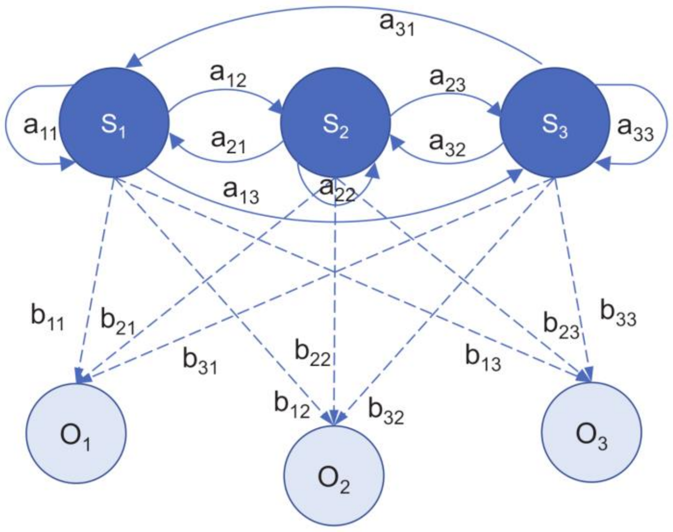
- Events we are interested in are **hidden**: we don't observe them directly

- For example, we don't normally observe part-of-speech tags in a text

Hidden Markov Models

- **A hidden Markov model (HMM)** allows us to talk about:
  - Observed events hidden Markov model (like words that we see in the input)
  - Hidden events (like part-of-speech tags)

$Q = q_1 q_2 \dots q_N$	a set of N **states**
$A = a_{11} \dots a_{ij} \dots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \dots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, ..., v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state q_i
$\pi = \pi_1, \pi_2, ..., \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$



Markov Assumption: $P(q_i | q_1, ..., q_{i-1}) = P(q_i | q_{i-1})$

Output Independence: $P(o_i | q_1, \dots q_i, \dots, q_T, o_1, \dots, o_i, \dots, o_T) = P(o_i | q_i)$

Hidden Markov Models: POS Tagging

- **Transition probabilities (A)** represent the probability of a tag occurring given the previous tag

ความน่าจะเป็นที่ VB จะเกิดต่อจาก MD

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1},t_i)}{C(t_{i-1})} \qquad P(VB|MD) = \frac{C(MD,VB)}{C(MD)} = \frac{10471}{13124} = .80$$

- **The B emission probabilities**, represent the probability, given a tag

ความน่าจะเป็นที่ “will” จะเป็น MD

$$P(w_i|t_i) = \frac{C(t_i,w_i)}{C(t_i)} \qquad P(will|MD) = \frac{C(MD,will)}{C(MD)} = \frac{4046}{13124} = .31$$

Hidden Markov Models: POS Tagging

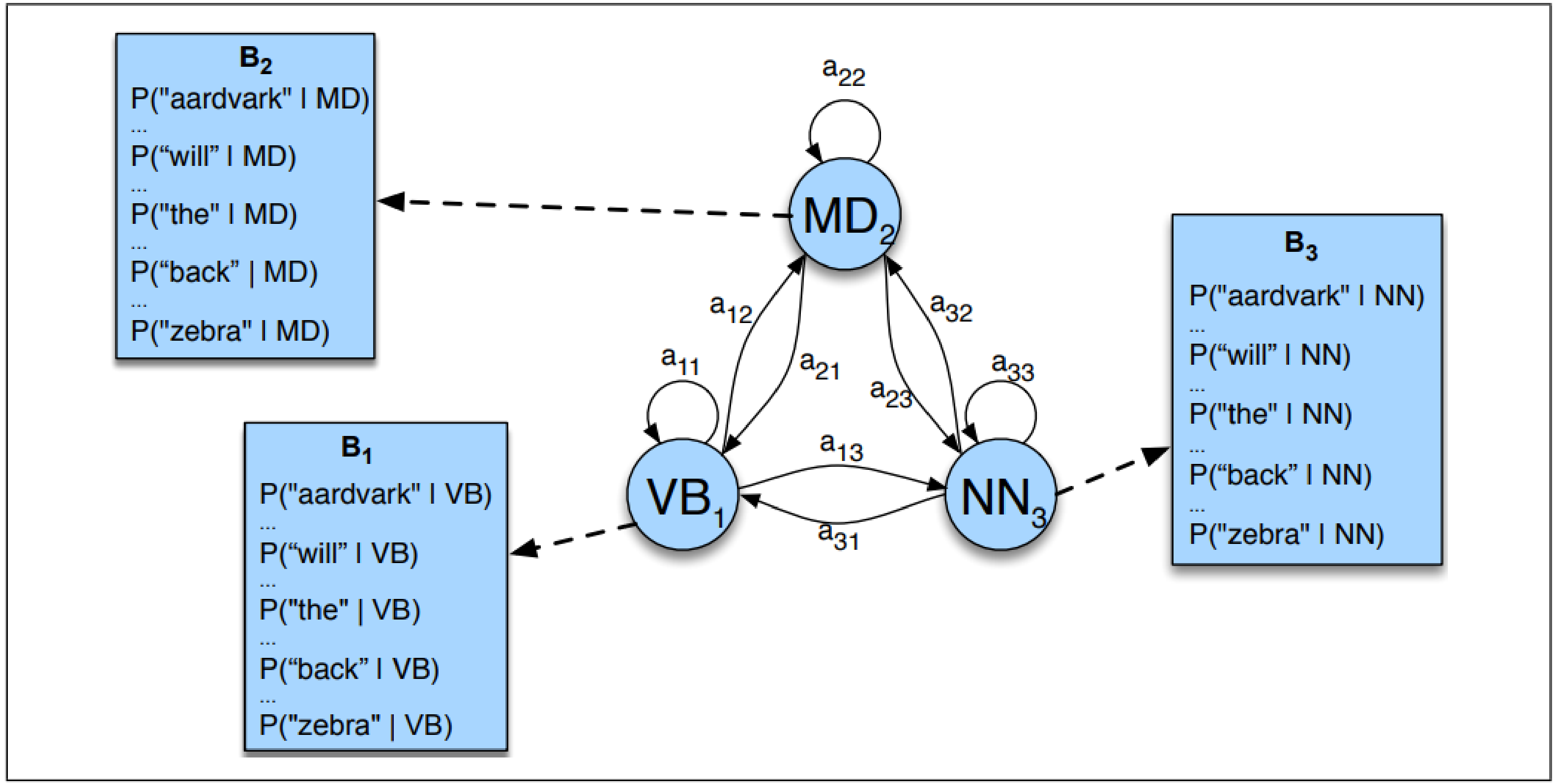


Figure 8.9 An illustration of the two parts of an HMM representation: the *A* transition probabilities used to compute the prior probability, and the *B* observation likelihoods that are associated with each state, one likelihood for each possible observation word.

HMM tagging as decoding

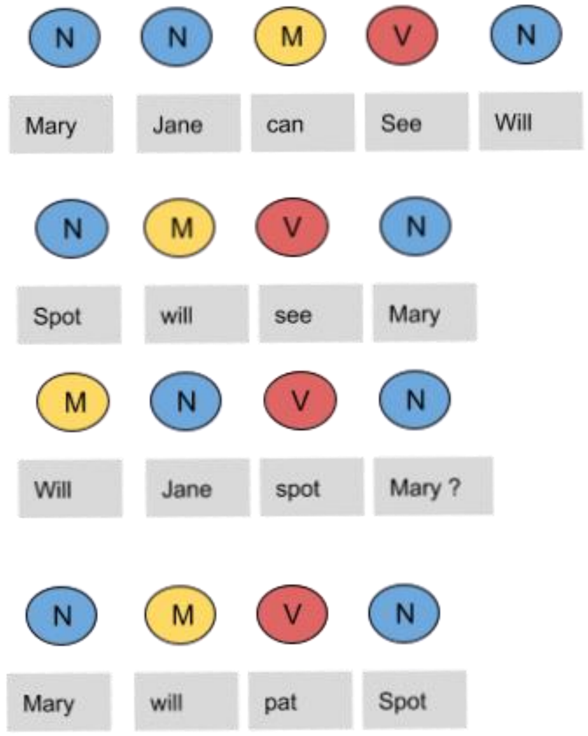
- **Decoding:** determining the hidden variables sequence corresponding to the sequence of observations
- Given as input an HMM λ = (A,B) and a sequence of observations $O = o_1,o_2,\ldots,o_T$, find the most probable sequence of states $Q = q_1q_2q_3 \ldots q_4$.
- For part-of-speech tagging:

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(t_1 \ldots t_n | w_1 \ldots w_n)$$

ต้องการหา (hidden variables sequence) t1…tn ที่ให้ prob มากสุด ซึ่งก็คือ POS นั่นเอง

POS Tagging Example

- Mini POS corpus

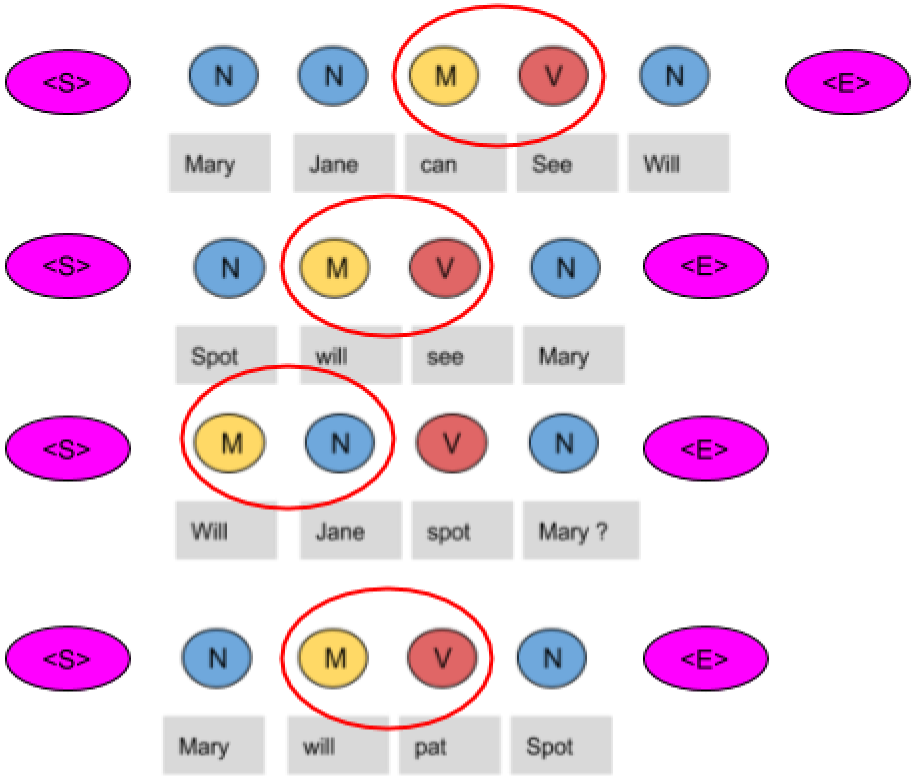


Words	Noun เกิด 9 ครั้ง	Modal เกิด 4 ครั้ง	Verb เกิด 4 ครั้ง
Mary	4/9	0	0
Jane	2/9	0	0
Will	1/9	3/4	0
Spot	2/9	0	1/4
Can	0	1/4	0
See	0	0	2/4
pat	0	0	1

https://www.mygreatlearning.com/blog/pos-tagging/

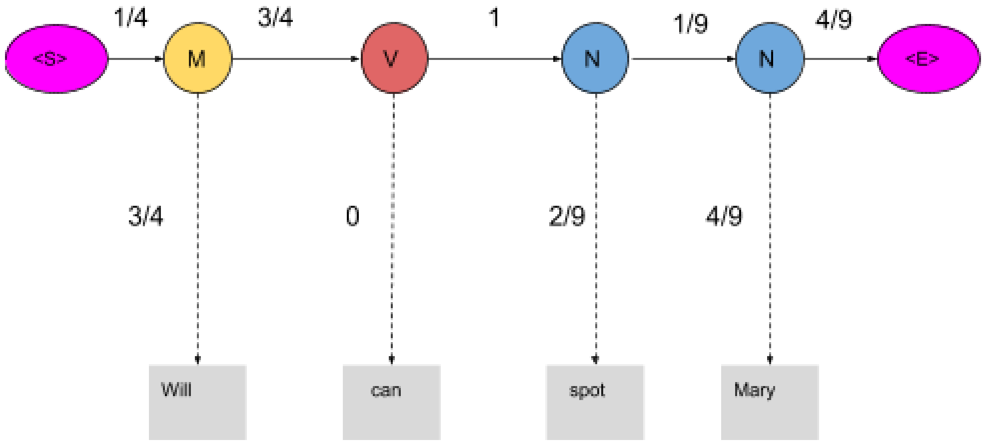
POS Tagging Example

- Include START <S> and END <E> tags:

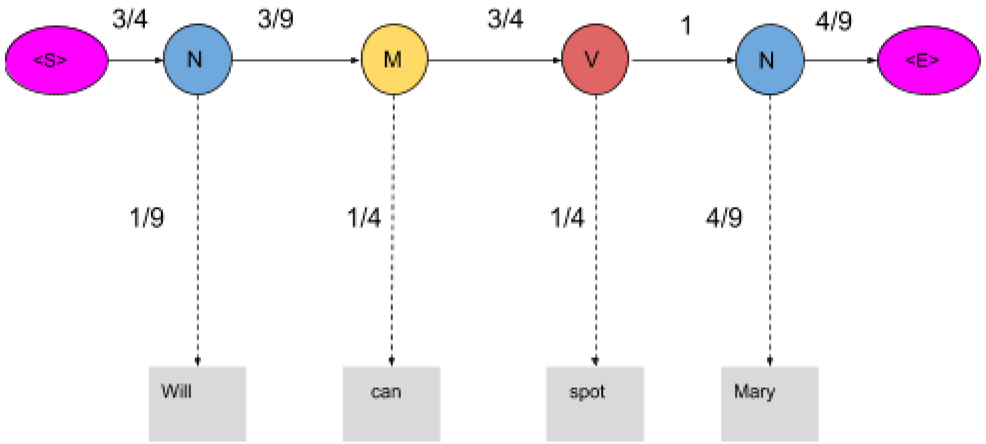


	N	M	V	<E>
<S>	3/4	1/4	0	0
N	1/9	3/9	1/9	4/9
M	1/4	0	3/4	0
V	4/4	0	0	0

Words	N	M	V
Mary	4/9	0	0
Jane	2/9	0	0
Will	1/9	3/4	0
Spot	2/9	0	1/4
Can	0	1/4	0
See	0	0	2/4
pat	0	0	1



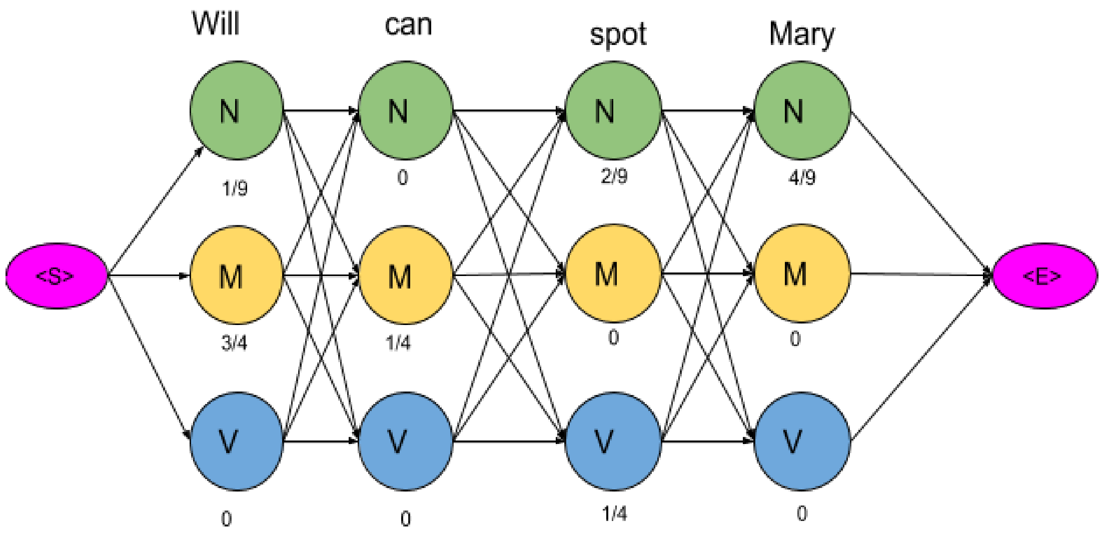
	N	M	V	<E>
<S>	3/4	1/4	0	0
N	1/9	3/9	1/9	4/9
M	1/4	0	3/4	0
V	4/4	0	0	0

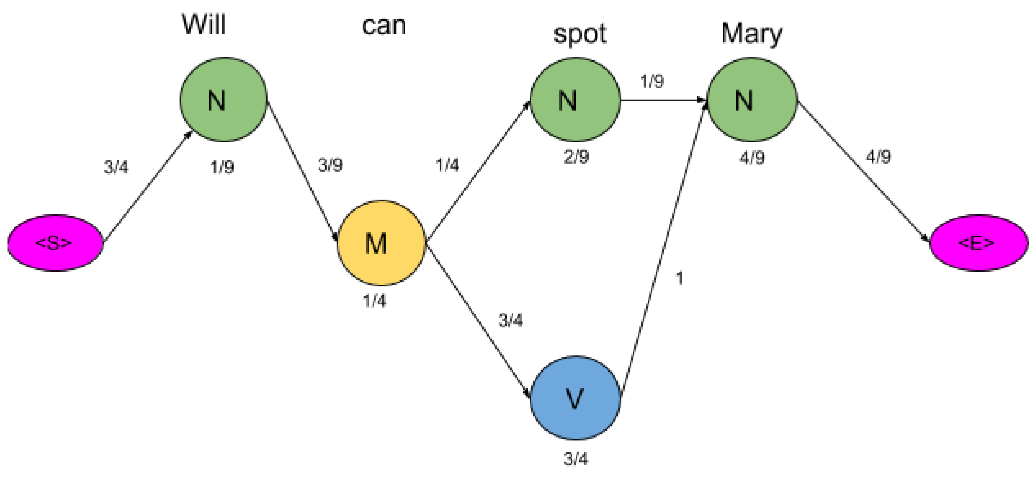


3/4*1/9*3/9*1/4*3/4*1/4*1*4/9*4/9=0.00025720164

POS Tagging Example

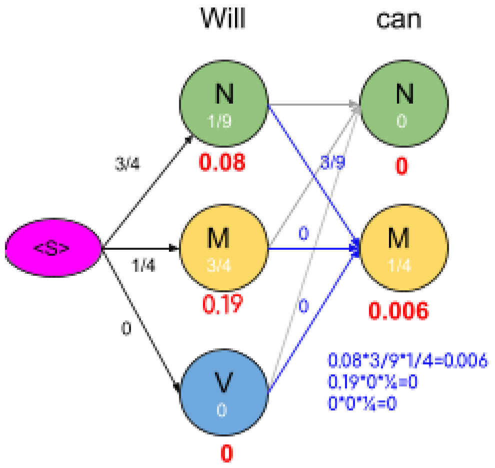
- Delete all the vertices and edges with probability zero

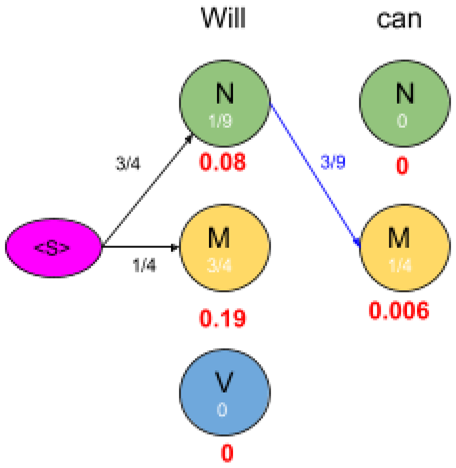


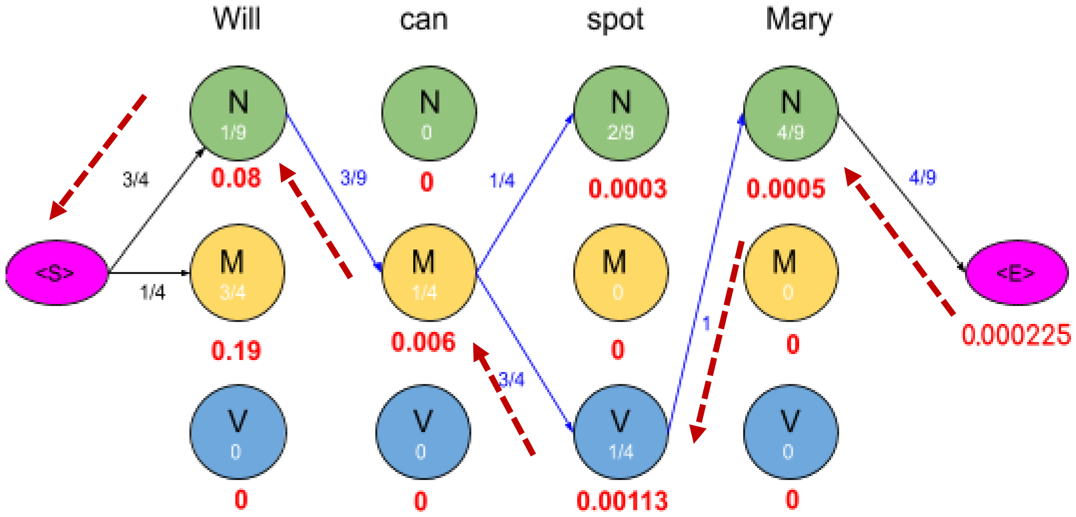


Applying Viterbi algorithm

- Optimize the HMM by using the Viterbi algorithm







start from the end and trace backwar

Conditional Random Fields (CRFs)

- HMM often run into unknown words
- But we can use the useful features to HMM
  - E.g. the previous or following words:
    - if the previous word is the, the current tag is unlikely to be a verb
- It’s hard for generative models like HMMs to add arbitrary features directly

การที่เราจะใช้ Feature อื่นเพิ่มเติม ทำได้ยาก

Conditional Random Fields (CRFs)

- A discriminative sequence model based on log-linear models
- **Linear chain CRF:**
  - Compute the posterior p(Y|X) directly:
  - Assuming we have a sequence of input words $X = x_1 \ldots x_n$ and want to compute a sequence of output tags $Y = y_1 \ldots y_n$.

$$\hat{Y} = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} P(\boxed{Y}|\boxed{X})$$

Conditional Random Fields (CRFs)

- CRF does not compute a probability for each tag at each time step
- CRF computes log-linear functions over a set of relevant features
- Assuming we have a sequence of input words $X = x_1 \ldots x_n$ and want to compute a sequence of output tags $Y = y_1 \ldots y_n$.
- Let's assume we have K features, with a weight w_k for each feature F_k :

Global features

$$p(Y|X) = \frac{\exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right)}{\sum_{Y' \in \mathcal{Y}} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y')\right)}$$

CRF as like a giant version of what multinomial logistic regression does for a single token.

$$F_k(X,Y) = \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i)$$

a sum of local features for each position *i* in *Y*

แทนที่จะใช้เฉพาะคำ ก็ใช้ Feature แทน เช่น คำข้างเคียง

Conditional Random Fields (CRFs)

- Features in a CRF POS Tagger:

$$F_k(X,Y) = \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i)$$

  - a linear-chain CRF, each local feature f_k at position i can depend on any information from: (y_{i-1}, y_i, X, i).

$$\mathbb{1}\{x_i = \textit{the},\ y_i = \text{DET}\}$$ เป็น 1 ถ้าเป็นจริง 0 ถ้าเป็นเท็จ

$$\mathbb{1}\{y_i = \text{PROPN},\ x_{i+1} = \textit{Street},\ y_{i-1} = \text{NUM}\}$$

$$\mathbb{1}\{y_i = \text{VERB},\ y_{i-1} = \text{AUX}\}$$

  - Example : Janet/NNP will/MD $\boxed{\text{back}}$/VB the/DT bill/NN, Xi = back

f_{3743}: y_i = VB and x_i = back

f_{156}: y_i = VB and y_{i-1} = MD

f_{99732}: y_i = VB and x_{i-1} = will and x_{i+2} = bill

Feature templates

$$\langle y_i, x_i \rangle, \langle y_i, y_{i-1} \rangle, \langle y_i, x_{i-1}, x_{i+2} \rangle$$

Neural based Sequence Labeling

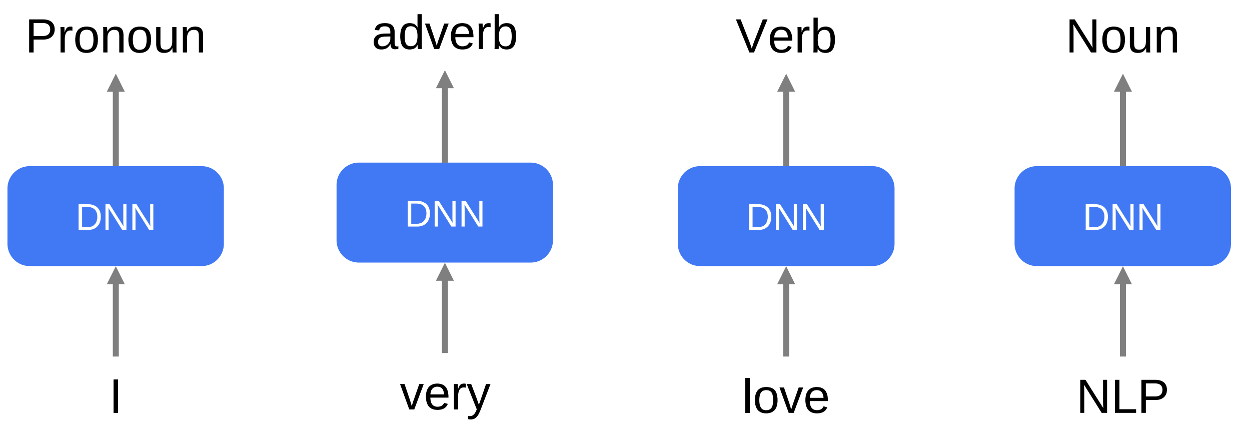
RNN Sequence labeling

- **Preparing data for sequence tagging**

Word	POS	NE	Clause
นายก	NN	O	B_CLS
ฯ	PU	O	I_CLS
_	PU	O	I_CLS
ยืนยัน	VV	O	I_CLS
_	PU	O	I_CLS
การ	FX	O	I_CLS
ประชุม	VV	O	I_CLS
_	PU	O	I_CLS
ก.ต.ช.	NN	B_ORG	I_CLS
พรุ่ง	NN	O	I_CLS
นี้	AJ	O	I_CLS
จะ	AX	O	I_CLS
ได้	VV	O	I_CLS
ตัว	NN	O	I_CLS

RNN Sequence labeling

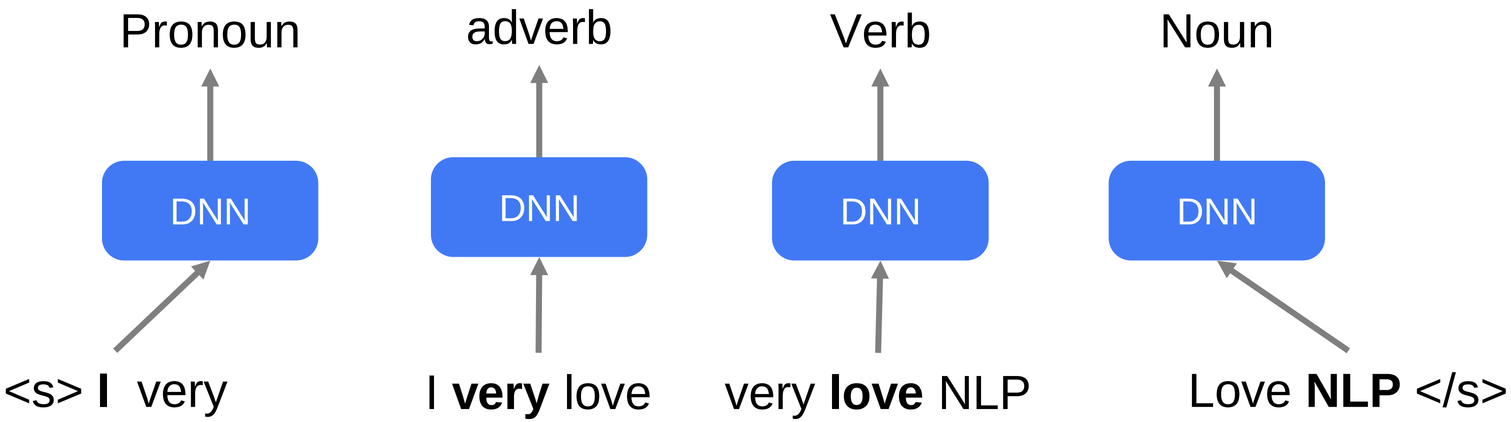
- Generating tags for sequence of tokens via DNN



- The most NLP tasks requires the contextual information (e.g. POS, NER, Word segmentation)
- We cannot get context information by DNN

RNN Sequence labeling

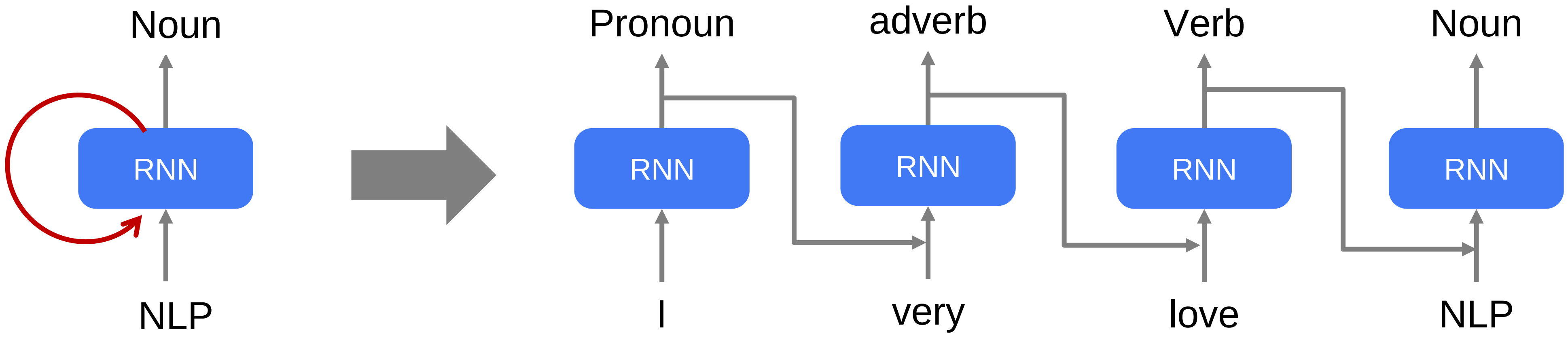
- Generating tags for sequence of tokens via DNN



- The most NLP tasks requires the contextual information (e.g. POS, NER, Word segmentation)
- We cannot get context information by DNN

Why RNN?

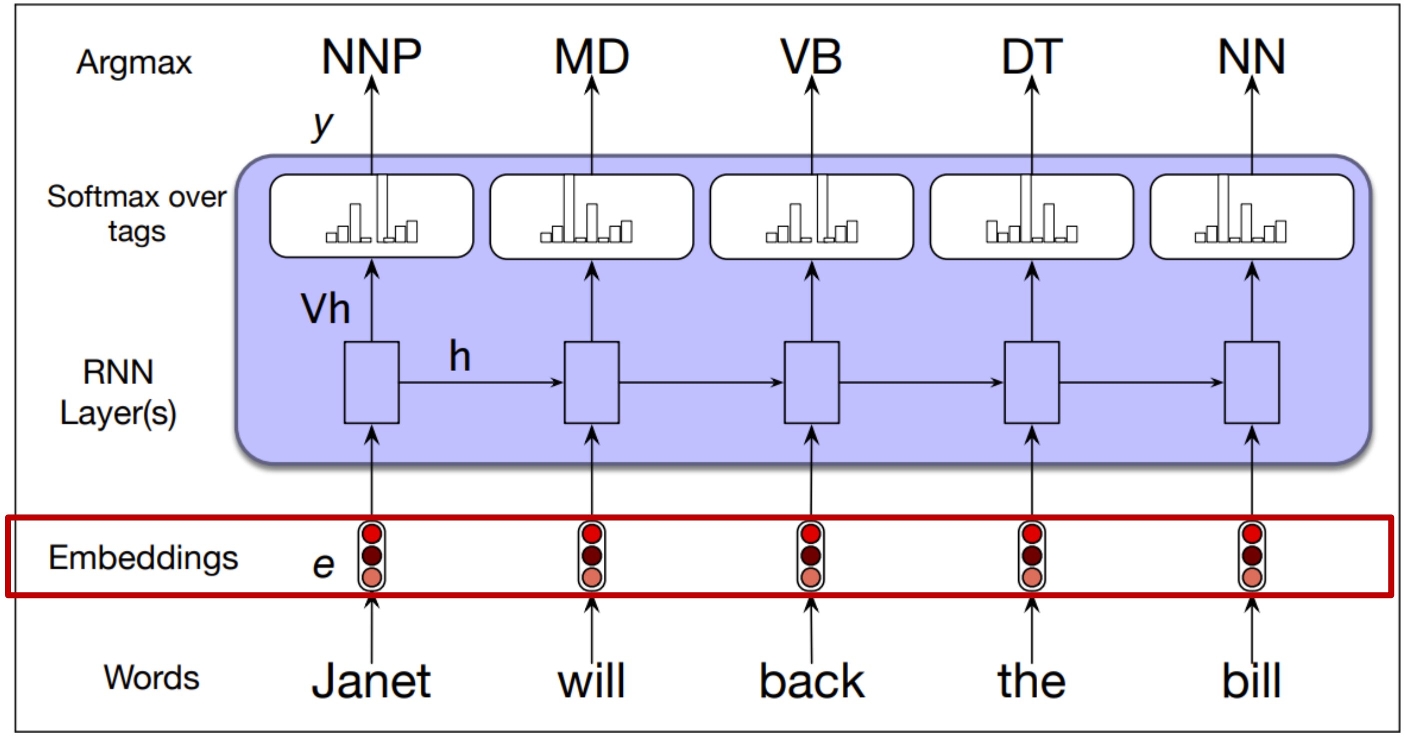
- Outputs from previous time steps are taken as inputs for the current time step
- Thus, it can learn the context information



ใช้ Weights อันเดียวกัน

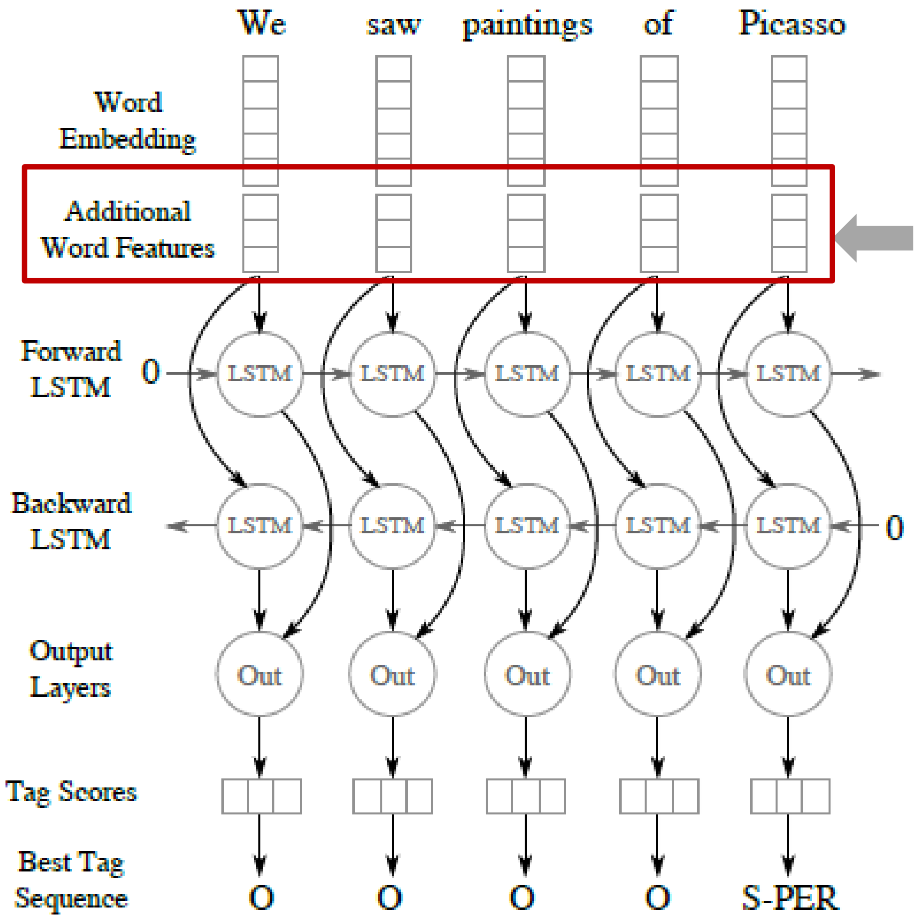
RNN Sequence labeling

- **Embedding layer:**



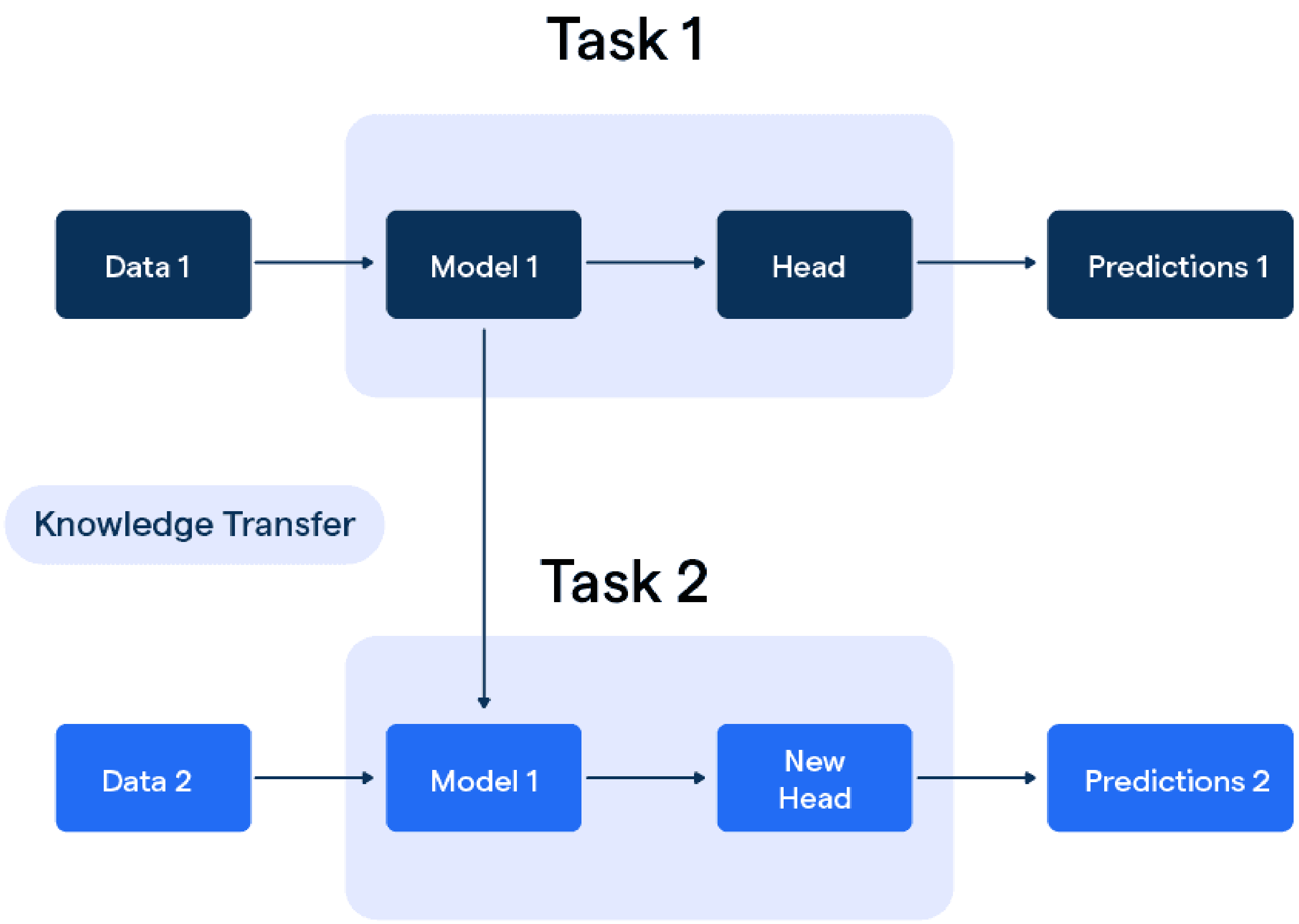
RNN Sequence labeling (LSTM, GRU)

- Generating tags for sequence of tokens via RNN
- NER:



You can add the additional features for training e.g. POS

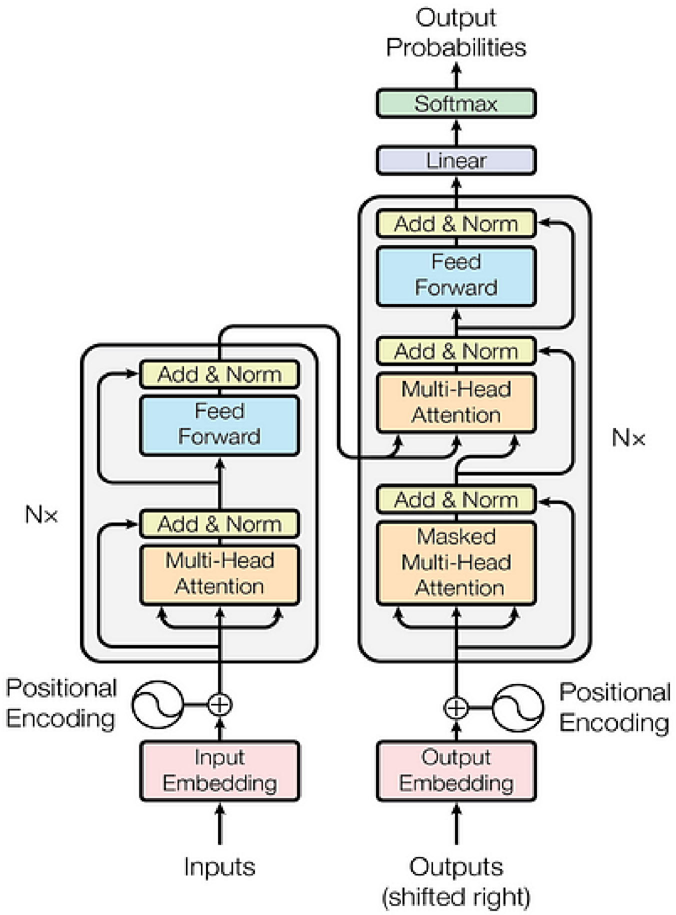
Transfer learning



BERT: Bidirectional Encoder Representations from Transformers

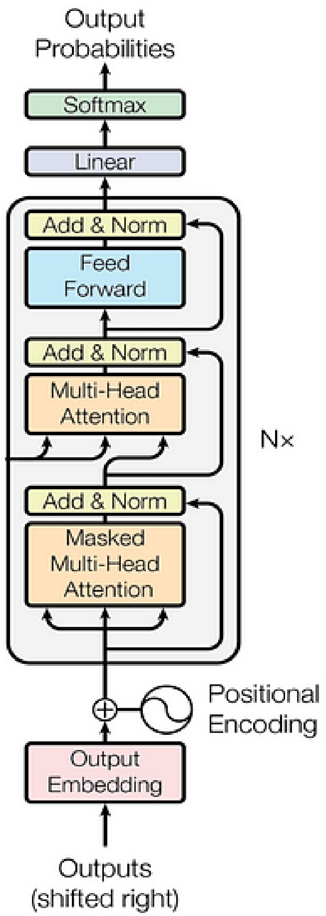
- BERT can be used as an all-purpose pre-trained model fine-tuned for specific tasks.
- Its goal is to generate a language model
- Bidirectional training using a Transformer Encoder
- **BERT** was trained on two modeling methods:
  - MASKED LANGUAGE MODEL (MLM)
  - NEXT SENTENCE PREDICTION (NSP)

Transformer



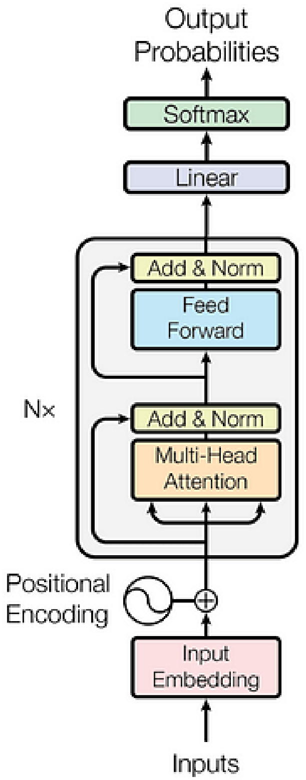
Encoder **Decoder**

GPT*



Decoder-only

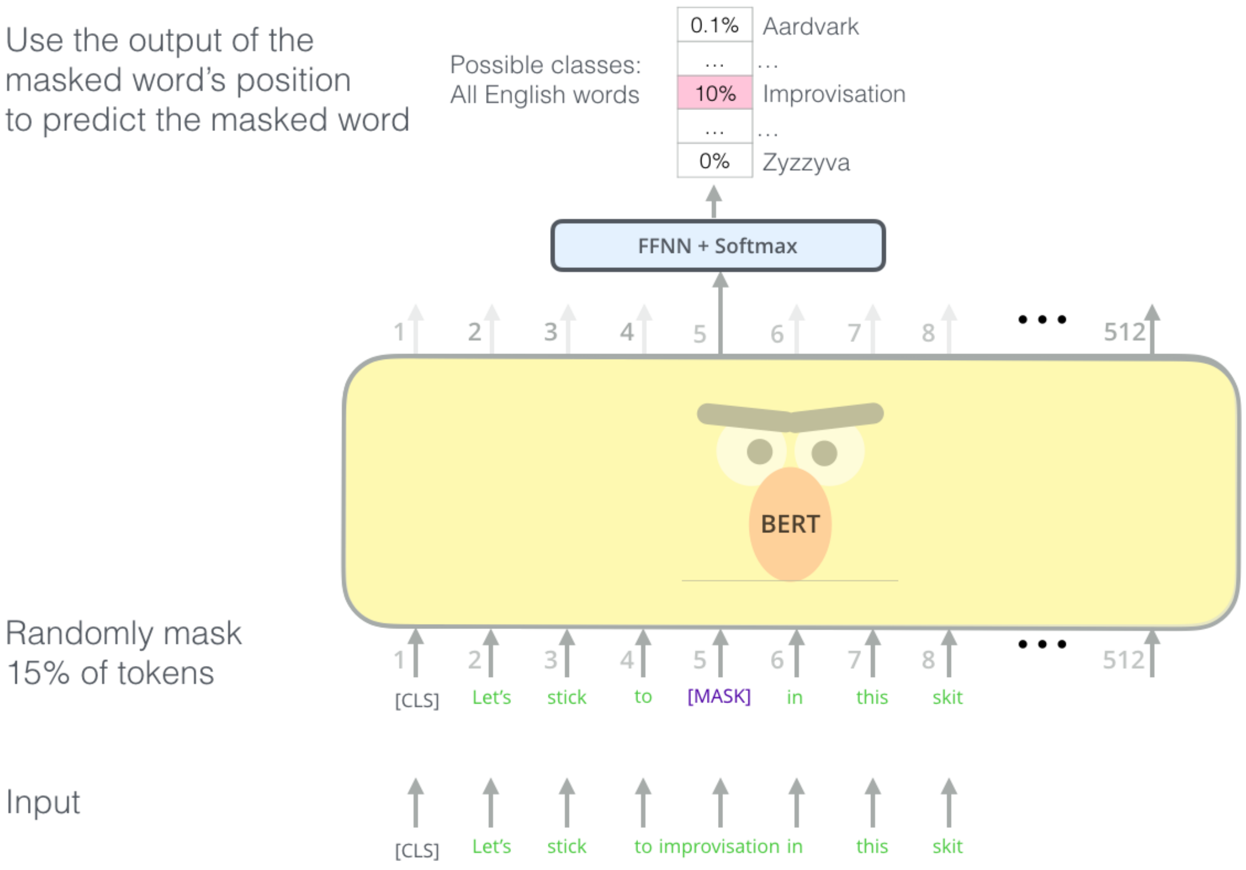
BERT*



Encoder-only

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

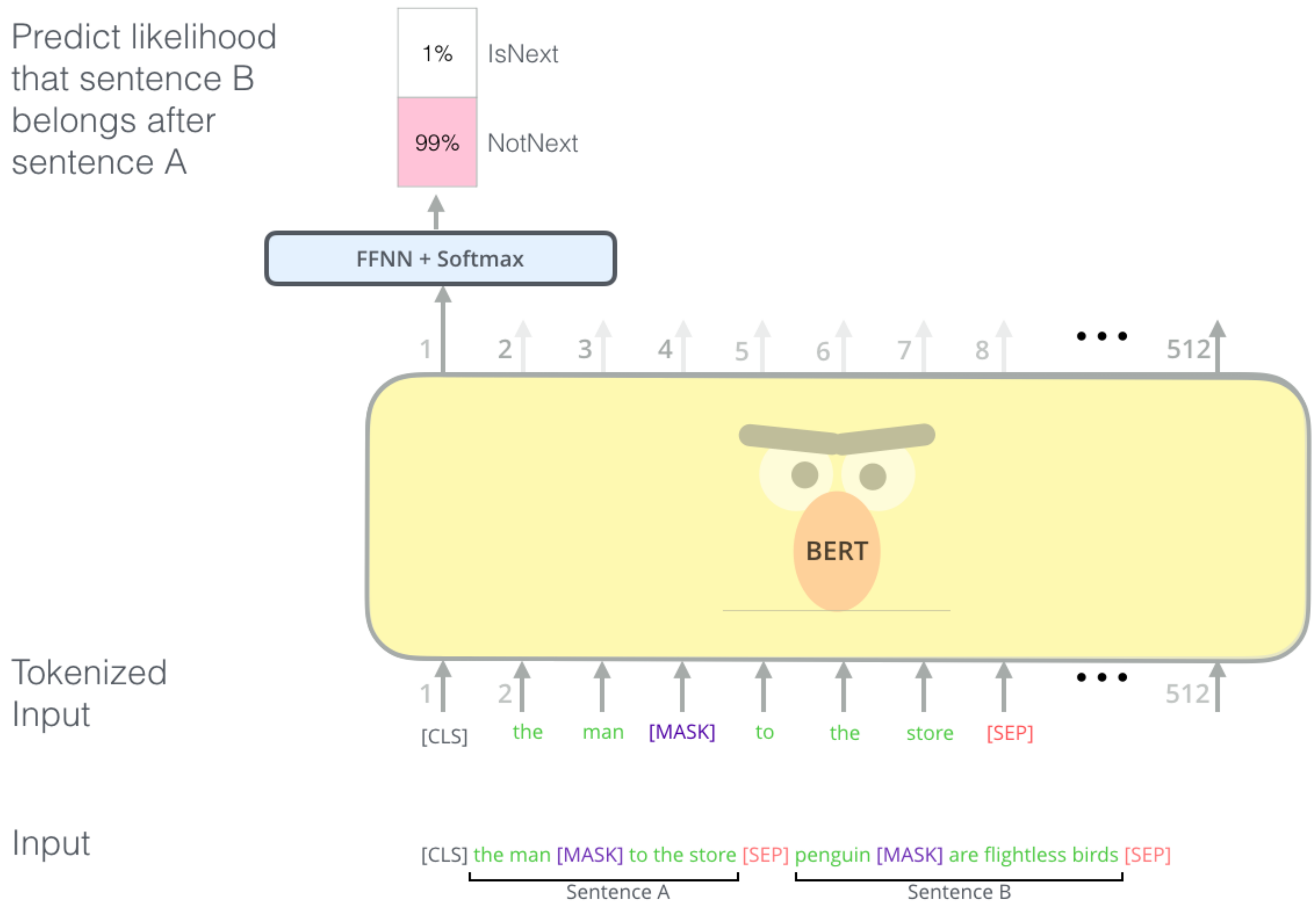
- MASKED LANGUAGE MODEL (MLM)
  - 15% of the words in each sequence are replaced with a [MASK] token.



http://jalammar.github.io/illustrated-bert/

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

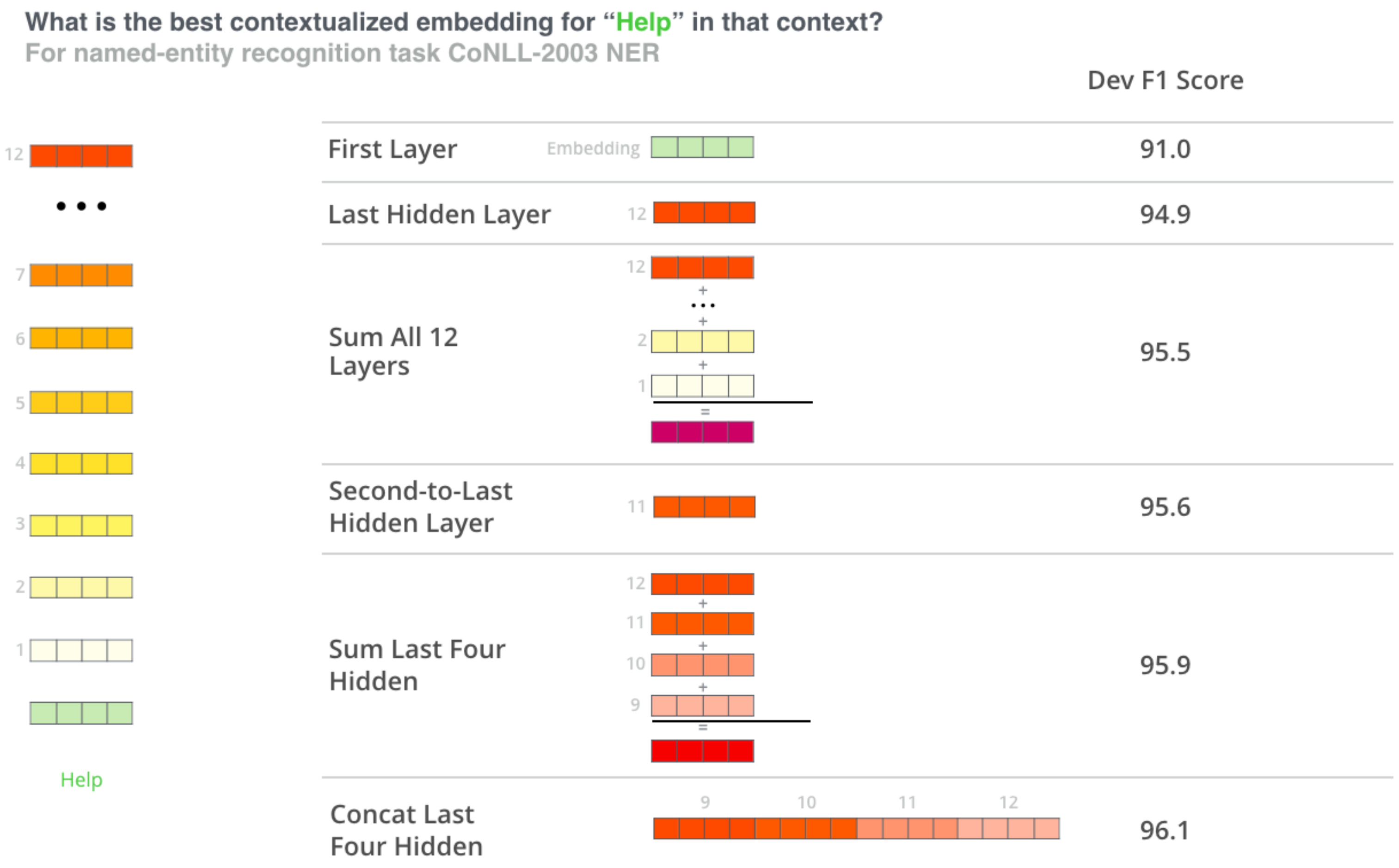
- Next Sentence Prediction (NSP)
  - Given two sentences (A and B), is B likely to be the sentence that follows A, or not?



http://jalammar.github.io/illustrated-bert/

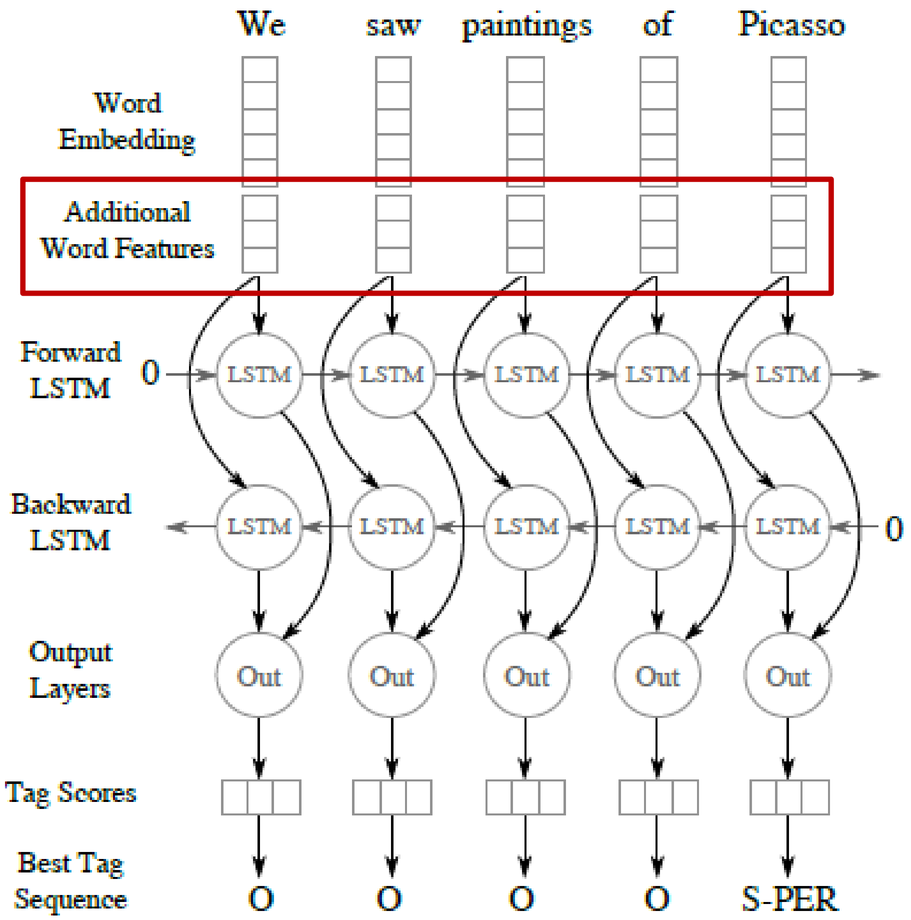
Using Pretrained Word Embedding

- Example : **Employing word embedding form BERT**



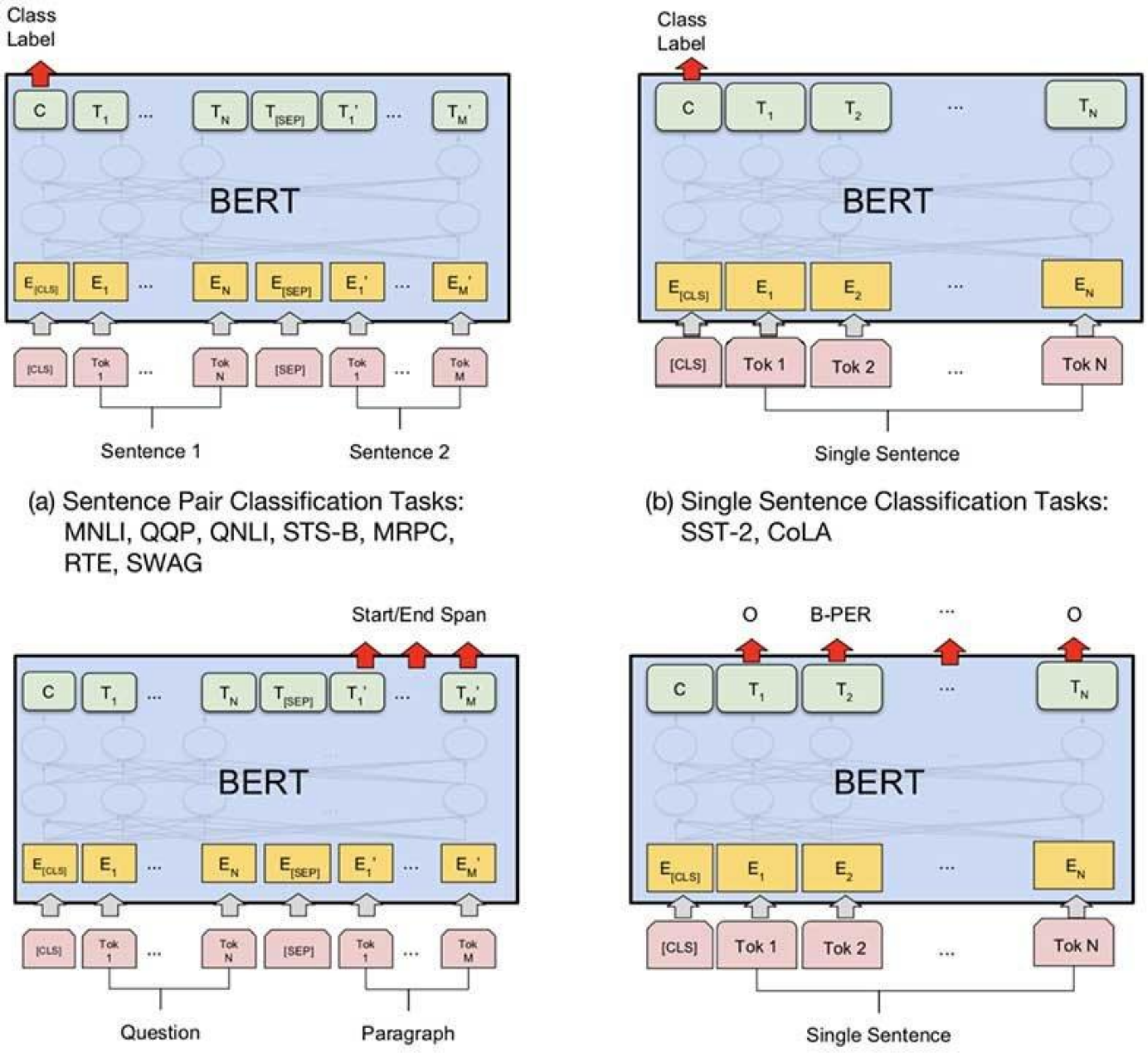
- BERT มี 12 Layer
- แต่ละ Layer เรียนรู้ในระดับที่แตกต่างกัน
- สามารถเลือกใช้ได้หลายแบบ
- First Layer
- Last Hidden Layer
- Sum All 12 layers

Using Pretrained Word Embedding



- **<u>Concatenate</u>** or **<u>replace</u>** the word embedding with **pre-trained word embedding**
- สามารถเลือกใช้ Embedding vector จาก Pretrained โมเดล อย่างเดียวก็ได้
- อาจจะต้องปรับโครงสร้างโมเดลให้สอดคล้องกับ input

Finetuning BERT



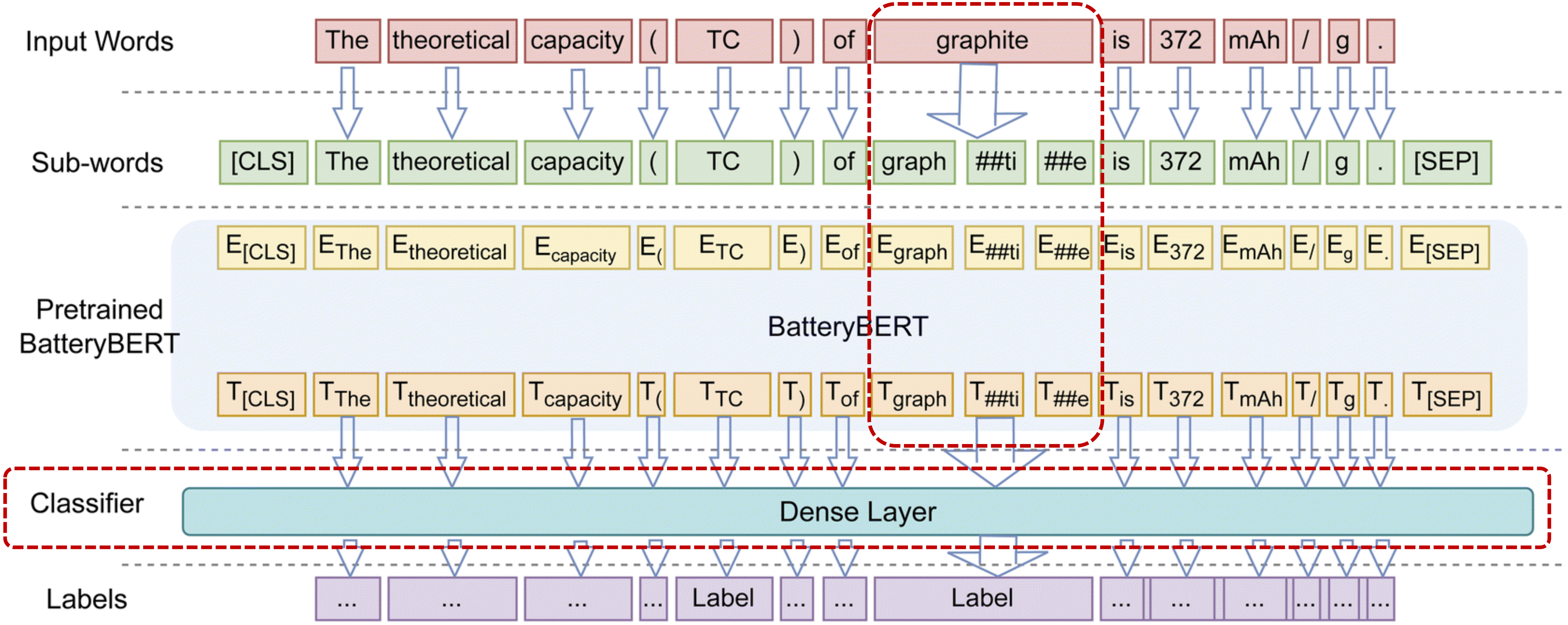
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Finetuning BERT for sequence tagging



https://pubs.rsc.org/image/article/2022/SC/d2sc04322j/d2sc04322j-f2_hi-res.gif

Conclusion

- Introduction to Sequence labeling
- NLP Sequence labeling tasks:
  - POS Tagging
  - NER
- Sequence labeling approaches:
  - HMM
  - CRF
  - RNN

Reference:

Dan Jurafsky and James H., Martin Speech and Language Processing (3rd ed. draft), https://web.stanford.edu/~jurafsky/slp3/