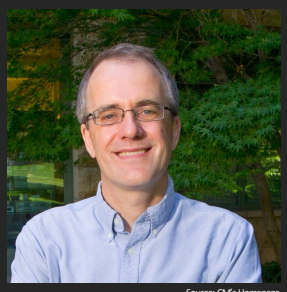


Source: CM's Homepage

Part-of-Speech Tagging from 97% to 100%
Is It Time for Some Linguistics?

Christopher Manning (2011)

All images from Wikimedia Commons unless specified.

1

Natural Language Processing (NLP)

COMP6713 – 2025 Term 1

Convener

Dr. Aditya Joshi

aditya.joshi@unsw.edu.au

Week/Module 7

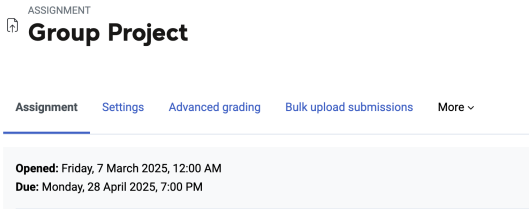
POS Tagging & NER

Schedule

2025 Term 1

2

Announcements (1/2)



Congratulations on completing your assignment!

Have you started working on the group project?

Thanks for the feedback on the course so far – from those who have attended the tutorials/consultations

Weeks 9 and 10:

- Public holidays on Friday
- Videos will be shared earlier in the week
- **Thursday lectures will assume you have seen the videos**

3

Announcements (2/2)

Taste-of-Research Projects

NLP for legal contracts

https://www.unsw.edu.au/engineering/student-life/undergraduate-research-opportunities/advertised-taste-research-areas/llms-for-law-legal-specific-llms-for-legal-contract-classification

NLP & misinformation

https://www.unsw.edu.au/engineering/student-life/undergraduate-research-opportunities/advertised-taste-research-areas/benchmark-for-misrepresented-social-media-responses-to-government-posts

4

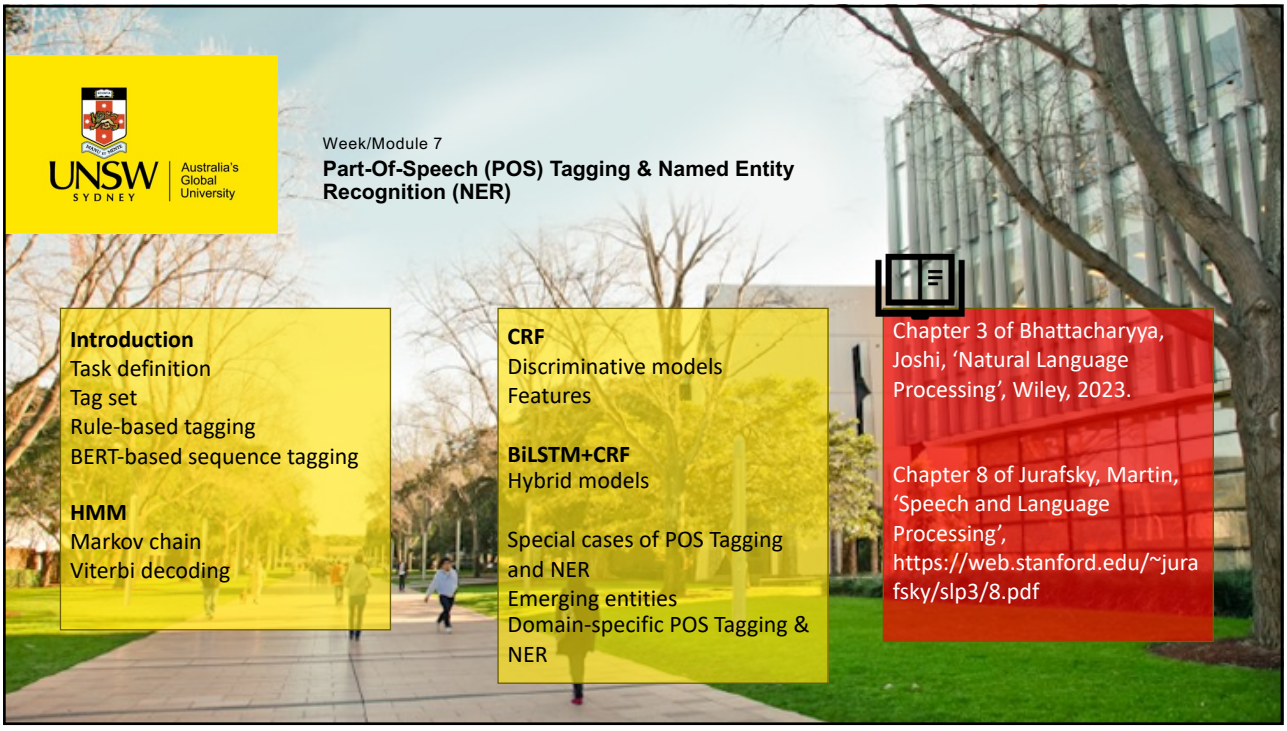
Fun Quiz

Please check Moodle for:

(NOT EVALUATED; Does NOT count towards your marks for the course)

Fun Quiz - Week 7

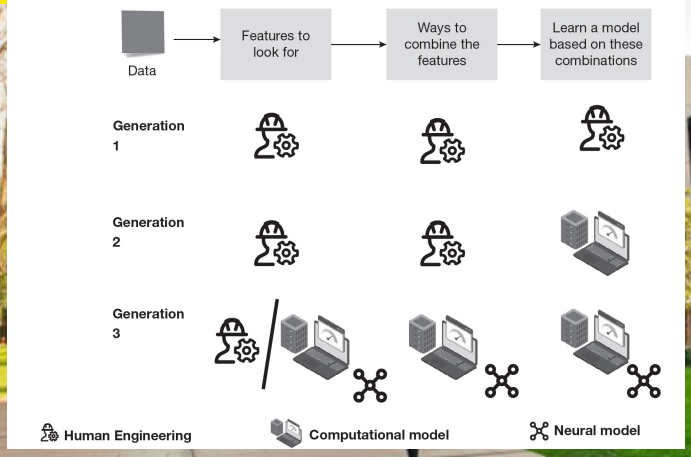
5



6

UNSW SYDNEY | Australia's Global University

Topics this week



HMM
CRF

BiLSTM+CRF
BERT

7

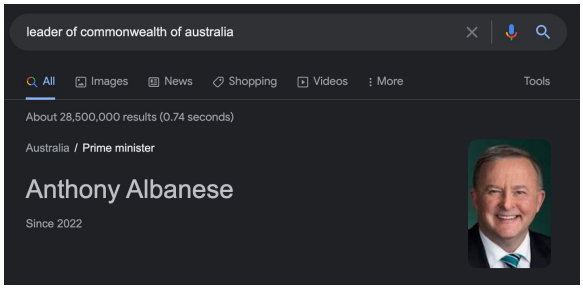
UNSW SYDNEY | Australia's Global University

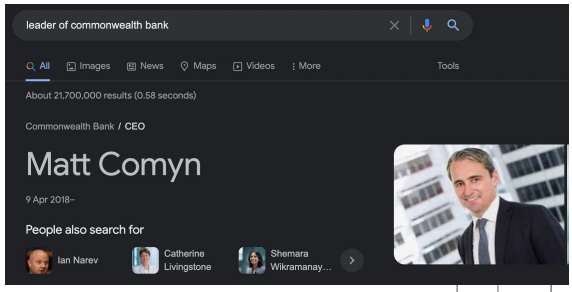
Part 0

Two Components of Tagging

8

Can you spot the NLP?





Screenshot from Google.com; Accessed on 25th September 2022.

UNSW SYDNEY

9

Welcome back!

Week 5: Sentiment analysis (Sequence classification)

This week: POS Tagging & NER (Sequence tagging/labeling)

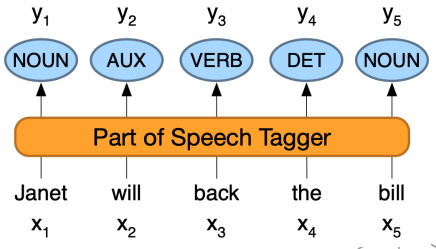
POS -> Part-of-speech

NER -> Named Entity Recognition

Sequence tagging:

Every unit in a sequence is assigned a tag among a set of tags of interest.

Transformers: **TokenClassification** **[1]**



"Performance on both datasets — 97.55% accuracy for POS tagging and 91.21% F1 for NER. (Ma and Hovy, 2016)"

UNSW SYDNEY

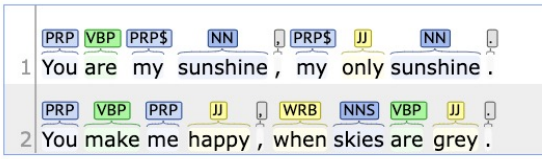
10

Part-of-speech (POS) tagging

Objective: Tag **every** word in a sentence with **a** part-of-speech (POS) tag.

Penn TreeBank (1989-1996) consists of over 4.5 million words of American English.
Also see: Brown Corpus

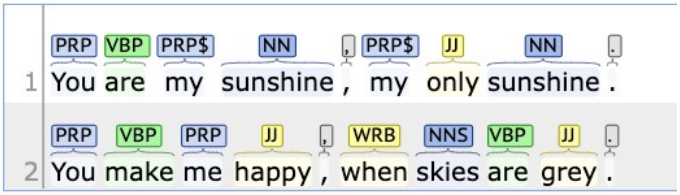
Example:
You are my sunshine, my only sunshine. You make me happy, when skies are grey.



https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html
Penn TreeBank: Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, *19*(2), 313-330.

11

POS Tagset



You	are	my	sunshine	,	my	only	sunshine	.
PRP	VBP	PRP$	NN	,	PRP$	JJ	NN	.

Notice:
IN: Preposition or subordinating conjunction
TO: to
But isn't TO a preposition too?

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

12

Named Entity Recognition (NER)

Objective: Tag every word with an NER tag

What is a named entity? An entity of a certain type

What is an NER tag? A tag that indicates presence and type of a named entity

Example: Commonwealth Bank of Australia is a bank located in the Commonwealth of Australia

ORGANIZATION: Commonwealth Bank of Australia is a bank located in the COUNTRY: Commonwealth of Australia

13

Wait a minute…

ORGANIZATION: Commonwealth Bank of Australia is a bank located in the COUNTRY: Commonwealth of Australia

- Not all words have been tagged in the above example!!!!!
- But, I said every word has a tag. That's not true in the visualization above!
- The above is only a front-end visualization.
- Each word is indeed labeled with one tag.
- **Tags in NER:**

B/I/O Tagset	B: Beginning of a named entity; I: Inside of a named entity; O: Outside of a named entity
ORG/LOC/…	Type of entity; ORG: Organisation, LOC: Location….

14

B/I/O Tagset

B/I/O Tagset B: Beginning of a named entity; I: Inside of a named entity; O: Outside of a named entity

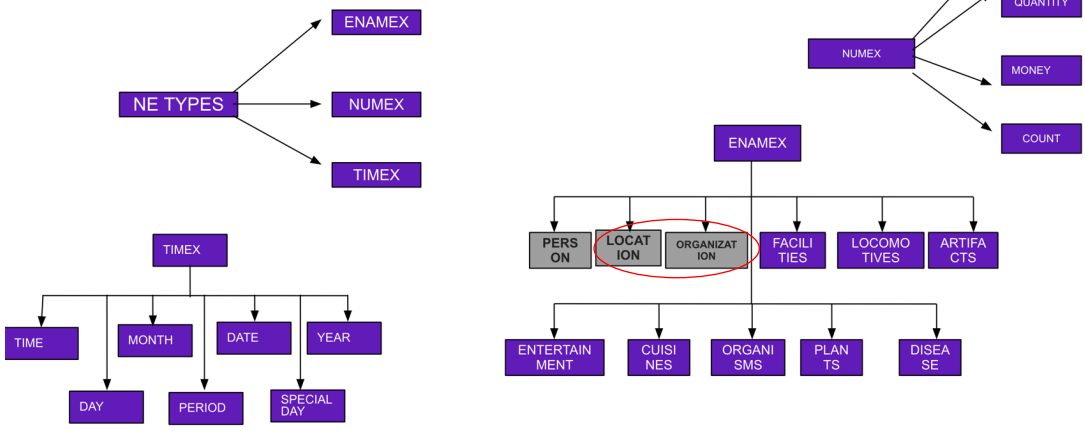
ORG/LOC/… Type of entity; ORG: Organisation, LOC: Location….

Commonwealth	Bank	is	located	in	the	Commonwealth	of	Australia	.
B	I	O	O	O	O	B	I	I	O
B-ORG	**I-ORG**	**O**	**O**	**O**	**O**	**B-LOC**	**I-LOC**	**I-LOC**	**O**

15

So, what are the possible entity types?

MUC Tagset for NER



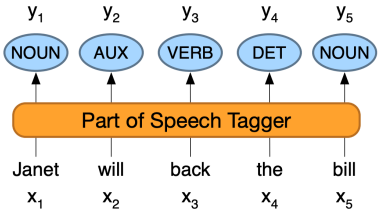
Nancy Chinchor. MUC-6 Named Entity Task Definition (Version 2.1). MUC-6. Columbia, Maryland. 1995.

16

Mmm.. okay… let's look at them again!

POS Tagging

You	are	my	sunshine	,	my	only	sunshine	.
PRP	VBP	PRP$	NN	,	PRP$	JJ	NN	.



Named Entity Recognition

Commonwealth	Bank	is	located	in	the	Commonwealth	of	Australia	.
B-ORG	I-ORG	O	O	O	O	B-LOC	I-LOC	I-LOC	O

17

Your turn! (1/2)

Can you write the POS tags for the words in the following sentence?

Trust	me	,	it	gets	better	!
VB	PRP	,	PRP	VBZ	JJR	!

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

18

Your turn! (2/2)

Can you write the NER tags for the words in the following sentence?

Fifty people injured on LATAM Airlines flight from Sydney to Auckland

X-Y:

X:
B: Beginning
I: Inside
O: Outside

Y:
ORG: Organisation
PER: Person
LOC: Location
MON: Money
NUM: Number
DIS: Distance

Fifty	people	injured	on	LATAM	Airlines	flight	from	Sydney	to	Auckland
B-NUM	O	O	O	B-ORG	I-ORG	O	O	B-LOC	O	B-LOC

Question for you: If there's a B, I and O, why isn't there an E (End)?

Ambiguity Resolution in POS tagging

Ambiguity Resolution: Select one among many possible tags for each word

$Y^* = \text{argmax } P(Y|X)$ for input sentence X and tag sequence Y

bank → VB, NN

Words have a small set of possible tags. $P(y_i|x_i)$

I went to the bank → ~~VB~~, NN

$P(y_i|y_{i-1}, y_{i-2}, y_{i-3,\ldots})$

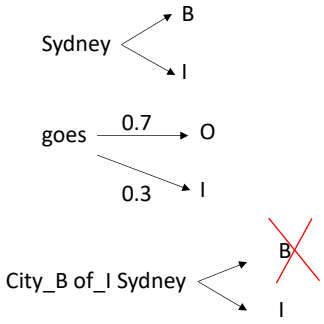
The POS tag of a word depends on words (and their tags) "around" it.

I went to the ____ → JJ, JJR…, NN, NNS…..

Ambiguity Resolution in NER

Ambiguity Resolution: Select one among three possible tags for each word; Select one among many entity types

Y* = argmax P(Y|X) for input sentence X and tag sequence Y



Capitalisation can be an underlying feature in some languages. $P(y_i|x_i)$

A word bears its own NER likelihood.

The NER tag of a word depends on words "around" it. $P(y_i|y_{i-1}, y_{i-2}, y_{i-3,....})$

Fundamental components of sequence tagging

Each tag is determined by….

1) Information about the word itself $P(y_i|x_i)$

2) ***Information*** about neighbouring tags $P(y_i|y_{i-1}, y_{i-2}, y_{i-3,....})$

$P(y_i|y_{i-1}, y_{i-2}, y_{i-3,....}....y_{i+1}, y_{i+2}, y_{i+3,....})$???

Why POS Tagging & NER?

POS tagging and NER can help several downstream tasks:

Sentiment Analysis: Focus on adjectives (Append POS embeddings to word embeddings)

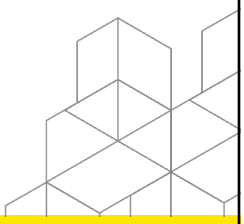
Information Retrieval: Create an index of documents with certain named entities (e.g. Documents that mention “Sydney Mardi Gras” - not the same as “Sydney”)

Question-answering: “Who heads Sydney Uni?”

Information Extraction

POS tagging performance nearly the same for statistical and neural models.

The two represent sequence labeling/tagging problems in NLP



UNSW SYDNEY

23

Obvious approach 1: Use rules

Lexicons: WordNet

Gazetteers: Lists of companies and cities; Wikipedia index

Look up words in the lexicon to select subset of tags

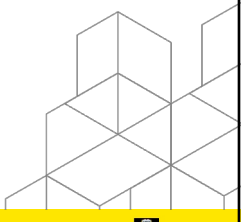
Add rules about context in order to select a tag

E.g.

lexicon says: “bank” can be a noun or a verb

Rule says: if the preceding word is ”a”, “an” or “the”, then bank is a noun.

Read more: Brill (1992)

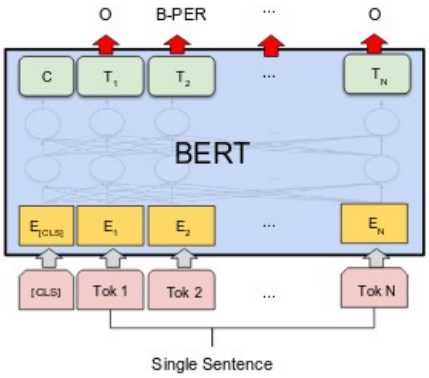


Brill, Eric. "A simple rule-based part of speech tagger." *Speech and Natural Language: ACL*. 1992.

UNSW SYDNEY

25

Obvious approach 2: Use BERT for token classification

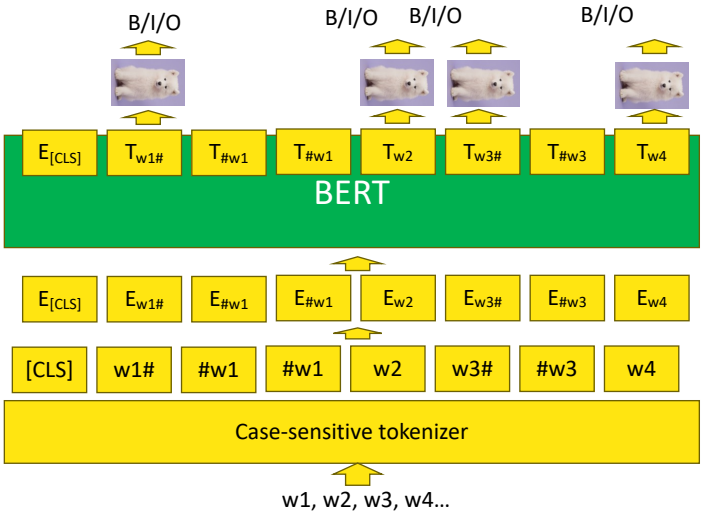


First of all, contrast this with what we saw for sentiment classification!

What changes would we need to the BERT model to use it for sequence tagging tasks such as NER?

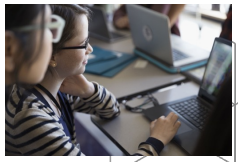
- **Case-sensitive** tokenizer
- Representation of the first subword token to predict the NER label

NER as multi-class classification over tokens



Final layer representations

Tokens



Demo time!

Also see: https://github.com/entbappy/NLP-Projects-Notebooks/blob/master/Fine-Tuning-BERT-for-NER.ipynb

POS Tagging: Back to 2011

"taggers now clearly exceed human performance on the task"

Table 4. Frequency of different POS tagging error types.

Class	Frequency
1. Lexicon gap	4.5%
2. Unknown word	4.5%
3. Could plausibly get right	16.0%
4. Difficult linguistics	19.5%
5. Underspecified/unclear	12.0%
6. Inconsistent/no standard	28.0%
7. Gold standard wrong	15.5%

Early availability of labeled datasets… 1991

Christopher Manning. 2011. "Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?" In Proceedings of CICLING.

28

UNSW SYDNEY | Australia's Global University

Part 1

HMM & CRF

Markov chain models: https://web.stanford.edu/~jurafsky/slp3/A.pdf
HMM-based tagging: Kupiec, Julian. "Robust part-of-speech tagging using a hidden Markov model." Computer speech & language 6.3 (1992): 225-242.
CRF-based tagging: Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." *In ICML*.

29

Datasets

POS Tagged Datasets

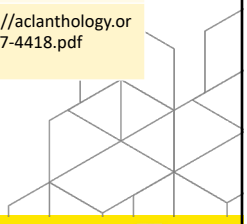
Name	Domain	Link
Penn TreeBank	News articles; English*	https://catalog.ldc.upenn.edu/LDC99T42
UD Dataset	Wikipedia, books, news articles; 60 languages*	https://universaldependencies.org/

NER Tagged Datasets

Name	Domain	Link
CONLL-2023	News articles; English and German*; 4 types of NER tags	https://aclanthology.org/W03-0419.pdf
OntoNotes	English, Arabic, Chinese*; 18 types of NER tags	https://catalog.ldc.upenn.edu/docs/LDC2013T19/OntoNotes-Release-5.0.pdf
Emerging Entities 2017	Train/test dataset for unseen entities; WNUT; English reddits*	https://aclanthology.org/W17-4418.pdf

Bender Rule: *A research convention in NLP: State the language of the dataset; English is not the default.*

A good resource to learn more about datasets: https://nlpprogress.com/

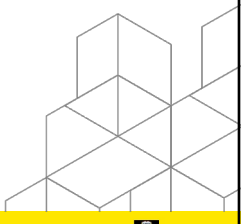


Objective: Compute the best tag sequence

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

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

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



Question: What is the POS tag of the next word in the sentence?

$$T^* = \arg\max_T (P(T|W))$$
$$= \arg\max_T \left[\frac{P(T).P(W|T)}{W} \right]$$
$$= \arg\max_T P(T).P(W|T)$$

The current POS tag is a determiner (DT) (i.e., 'a', 'an', or 'the').

-> The next POS tag can be an adjective or a noun.

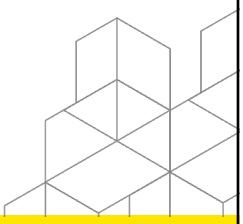
The current POS tag is a verb.

-> The next POS tag can be a determiner….

$$P(T) = P(\hat{}\, t_0 t_1 t_2 \ldots t_{n-2} t_{n-1} t_n .)$$
$$= P(\hat{}\,)P(t_0|\hat{}\,)P(t_1|t_0)P(t_2|t_1)\ldots P(t_1|t_{n-1})P(.|t_{n-1})$$
$$= P(t_0|\hat{}\,)P(t_1|t_0)P(t_2|t_1)\ldots P(t_n|t_{n-1})P(.|t_n)$$
$$= \prod_{i=0}^{n+1} P(t_i|t_{i-1})$$

Transition probability
Transition from one tag to the next

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



UNSW SYDNEY

32

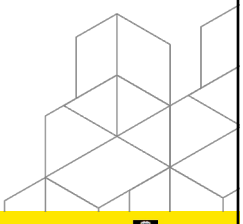
Question: How will the weather be tomorrow?

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

Potential Answer 1: I cannot determine for sure. It is impossible to know the future with certainty.

Potential Answer 2: Give me 2 hours. I will collect the weather data for the past 2 years and build a time series model.

Potential Answer 3: It's winter. It has been cold today. So, it is likely that the weather will be cold.

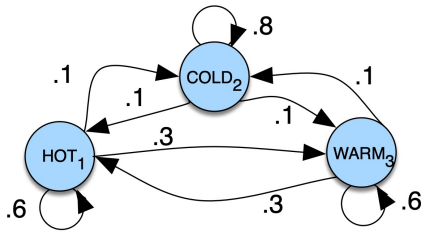


UNSW SYDNEY

33

Markov graph for weather

Edges: Transition matrix



(a)

What is the likelihood of a word, given a tag?

$$P(W|T) = P(\hat{}\, w_0 w_1 w_2 \ldots w_{n-2} w_{n-1} w_n . | \hat{}\, t_0 t_1 t_2 \ldots t_{n-2} t_{n-1} t_n .)$$
$$= P(\hat{}\,|\hat{}\,) P(w_0|t_0) P(w_1|t_1) P(w_2|t_2) \ldots P(w_n|t_n) P(.|.)$$
$$= P(\hat{}\,|\hat{}\,) P(w_0|t_0) P(w_1|t_1) P(w_2|t_2) \ldots P(w_n|t)$$
$$= \prod_{i=0}^{n+1} P(w_{i-1}|t_{i-1})$$

$$T^* = \arg\max_T (P(T|W))$$
$$= \arg\max_T \left[\frac{P(T).P(W|T)}{W} \right]$$
$$= \arg\max_T P(T).P(W|T)$$

… a word depends only on its own tag.

$$T^* = \arg\max_T \prod_{i=0}^{n+1} [P(t_i|t_{i-1}) P(w_{i-1}|t_{i-1})]$$
$$where \quad T : \hat{}\, t_0 t_1 t_2 \ldots t_{n-2} t_{n-2} t_{n-1} t_n .$$

Observation likelihood
Generation of a word, conditional on a tag

Interpreting the two terms…

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

$$P(w_1 \ldots w_n | t_1 \ldots t_n) \approx \prod_{i=1}^{n} P(w_i | t_i) \qquad P(t_1 \ldots t_n) \approx \prod_{i=1}^{n} P(t_i | t_{i-1})$$

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(t_1 \ldots t_n | w_1 \ldots w_n) \approx \underset{t_1 \ldots t_n}{\operatorname{argmax}} \prod_{i=1}^{n} \overbrace{P(w_i | t_i)}^{\text{emission}} \overbrace{P(t_i | t_{i-1})}^{\text{transition}}$$

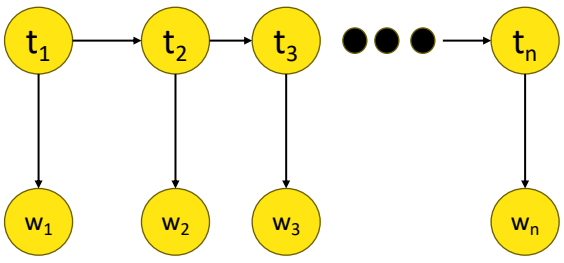
Remember 'two components'?

Hidden Markov Model

Used to model time series data

In the case of POS tagging or NER:

- Tags are hidden states arranged in a Markov chain
- Words are observations that depend only on the tag for the word



Viterbi algorithm

Dynamic programming algorithm that obtains the best sequence for a given word sequence.
In the case of HMM: best state sequence that emits the given word sequence.

Referred to as "Viterbi decoding"

'decoding' –(origin of the term)-> 'Decoder' in Transformer

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t (drawn from a vocabulary $V = v_1, v_2, \ldots, v_V$) being generated from a state q_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, $\sum_{i=1}^{n} \pi_i = 1$

UNSW SYDNEY

38

Markov graph for POS states + Observation likelihood

Observation likelihood

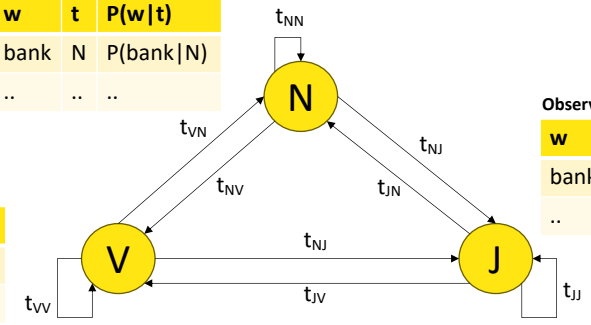
w	t	P(w|t)
bank	N	P(bank|N)
..	..	..

Observation likelihood

w	t	P(w|t)
bank	V	P(bank|V)
..	..	..

Observation likelihood

w	t	P(w|t)
bank	J	P(bank|J)
..	..	..



N: Noun
V: Verb
J: Adjective

t_{NN} = P(ti="N"|ti-1="N")
t_{NJ} = P(ti="J"|ti-1="N")

... Markov Assumption has made this possible.

UNSW SYDNEY

39

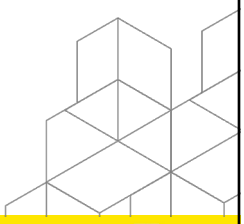
Example

The people dance.

Assume only three POS tags:
DT: Determiner
N: Noun
V: Verb

dance:
They **dance** well.
I liked the **dance** performance.

people:
People the meadow with flowers.
People love a good performance.

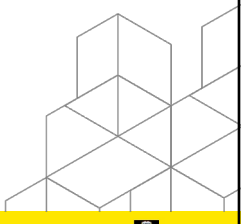


Pseudocode

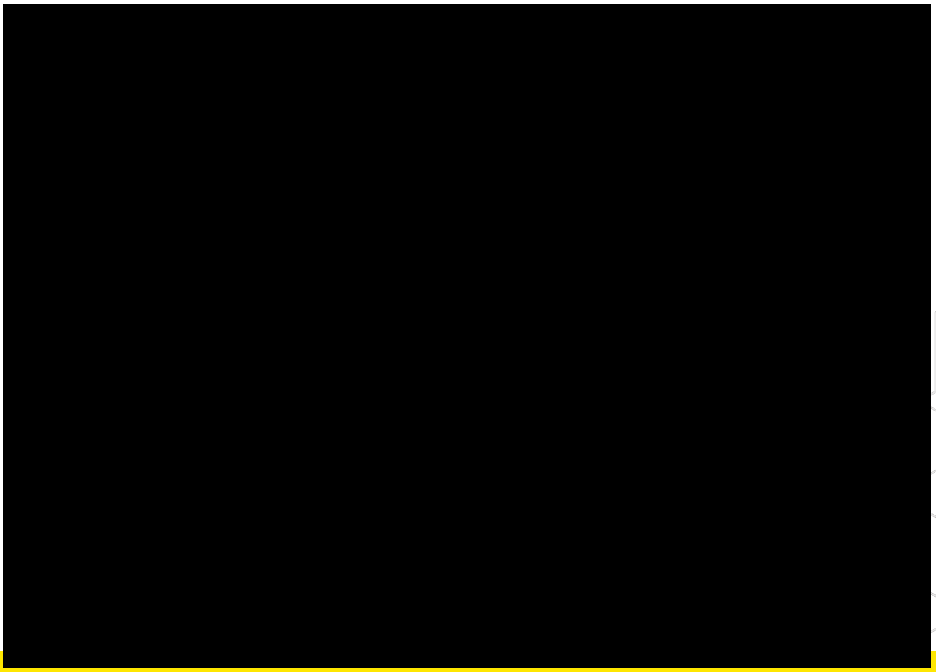
$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t (drawn from a vocabulary $V = v_1, v_2, ..., v_V$) 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, $\sum_{i=1}^{n} \pi_i = 1$

function VITERBI(*observations* of len T,*state-graph* of len N) **returns** *best-path*, *path-prob*

create a path probability matrix *viterbi[N,T]*
for each state s **from** 1 **to** N **do** ; initialization step
  $viterbi[s,1] \leftarrow \pi_s * b_s(o_1)$
  $backpointer[s,1] \leftarrow 0$
for each time step t **from** 2 **to** T **do** ; recursion step
  for each state s **from** 1 **to** N **do**
    $viterbi[s,t] \leftarrow \max_{s'=1}^{N} viterbi[s', t-1] * a_{s',s} * b_s(o_t)$
    $backpointer[s,t] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s', t-1] * a_{s',s} * b_s(o_t)$
$bestpathprob \leftarrow \max_{s=1}^{N} viterbi[s,T]$; termination step
$bestpathpointer \leftarrow \operatorname{argmax}_{s=1}^{N} viterbi[s,T]$; termination step
$bestpath \leftarrow$ the path starting at state *bestpathpointer*, that follows backpointer[] to states back in time
return *bestpath*, *bestpathprob*



Pseudocode from Jurafsky-Martin.



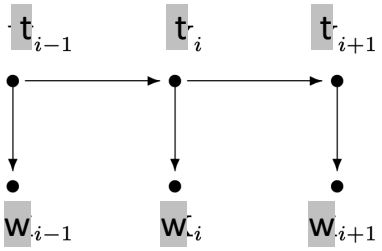
42

HMM is a generative model

$$T^* = \arg\max_T \prod_{i=0}^{n+1} [P(t_i \mid t_{i-1}) P(w_{i-1} \mid t_{i-1})]$$

$$where \quad T : \hat{} t_0 t_1 t_2 \ldots t_{n-2} t_{n-2} t_{n-1} t_n .$$

The model estimates the generation of a word, given a tag



To compute $P(w_i|t_i)$,

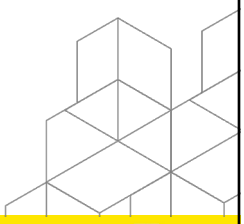
we need sufficiently large number of examples!

This model does not factor word features! (e.g.: a word ending in 'ed' might be a past tense verb)

43

Enter: Discriminative Models

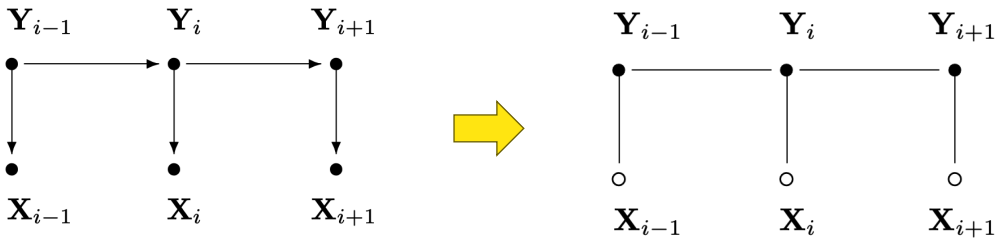
Generative Models	Discriminative Models
Model the joint distribution	Model the conditional distribution
Hidden Markov Models	Conditional Random Fields
Depends on large volume of data to be tractable	Can leverage feature independence



HMM -> CRF

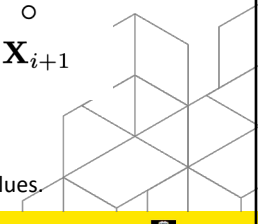
In CRF, words can be represented as features

Notice no 'chain' over the tags: What does this mean for the model?



HMM CRF

Filled circles indicate inferred values (via decoding or similar); Hollow circles indicate observed values.



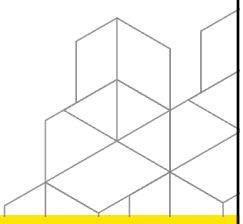
CRF: "Finite state model with unnormalized transition probabilities"

Definition. *Let $G = (V, E)$ be a graph such that $\mathbf{Y} = (\mathbf{Y}_v)_{v \in V}$, so that $\mathbf{Y}$ is indexed by the vertices of G. Then $(\mathbf{X}, \mathbf{Y})$ is a conditional random field in case, when conditioned on $\mathbf{X}$, the random variables $\mathbf{Y}_v$ obey the Markov property with respect to the graph: $p(\mathbf{Y}_v \mid \mathbf{X}, \mathbf{Y}_w, w \neq v) = p(\mathbf{Y}_v \mid \mathbf{X}, \mathbf{Y}_w, w \sim v)$, where $w \sim v$ means that w and v are neighbors in G.*

input Xs and output Ys are connected in a graph.
X: Words; Y: POS tags

POS tags are only dependent on those of neighbouring tags.

G is a simple chain or line: G = (V = {1, 2,...m}, E = {(i, i + 1)}).



UNSW SYDNEY

46

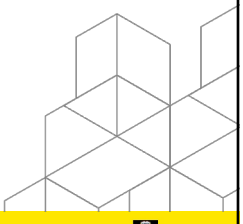
Theorem of random fields

$$p_\theta(\mathbf{y} \mid \mathbf{x}) \propto \exp\left(\sum_{e \in E, k} \lambda_k \, f_k(e, \mathbf{y}|_e, \mathbf{x}) + \sum_{v \in V, k} \mu_k \, g_k(v, \mathbf{y}|_v, \mathbf{x})\right)$$

edges

vertices

features combining the arguments



UNSW SYDNEY

47

The argmax for CRF

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

all possible tag sequences

$$= \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right)$$

features based on word and tag sequences

$$= \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \exp\left(\sum_{k=1}^{K} w_k \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i)\right)$$

... but features will only use y_{i-1} and y_i (Neighbourhood tags)

$$= \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \sum_{k=1}^{K} w_k \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i)$$

Therefore, a human designer will select the features

$$= \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \sum_{i=1}^{n} \sum_{k=1}^{K} w_k f_k(y_{i-1}, y_i, X, i)$$

...and the training algorithm will learn the corresponding weights.

UNSW SYDNEY

48

Features in a CRF

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

X: People danced

Feature 1 (Current Suffix): Last two letters of the current word

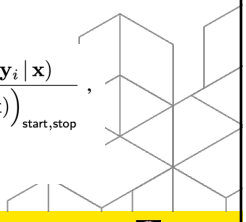
xi : “danced” f1: suffix(x) = “ed”

Feature 2 (Current Suffix): Last two letters of the current word and the POS of the previous word

xi : “danced”, f1: suffix(x) = “ed”,
yi-1: NNS prev(tag) = ”NNS”

$$p_\theta(\mathbf{y} \mid \mathbf{x}) = \frac{\prod_{i=1}^{n+1} M_i(\mathbf{y}_{i-1}, \mathbf{y}_i \mid \mathbf{x})}{\left(\prod_{i=1}^{n+1} M_i(\mathbf{x})\right)_{\text{start,stop}}},$$

where $\mathbf{y}_0 = \text{start}$ and $\mathbf{y}_{n+1} = \text{stop}$.



UNSW SYDNEY

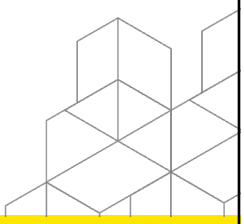
49

Some more features

4.2.1 Spelling features

We extract the following features for a given word in addition to the lower case word features.

- whether start with a capital letter
- whether has all capital letters
- whether has all lower case letters
- whether has non initial capital letters
- whether mix with letters and digits
- whether has punctuation
- letter prefixes and suffixes (with window size of 2 to 5)
- whether has apostrophe end ('s)
- letters only, for example, I. B. M. to IBM
- non-letters only, for example, A. T. &T. to ..&
- word pattern feature, with capital letters, lower case letters, and digits mapped to 'A', 'a' and '0' respectively, for example, D56y-3 to A00a-0
- word pattern summarization feature, similar to word pattern feature but with consecutive identical characters removed. For example, D56y-3 to A0a-0



UNSW SYDNEY

50

MAP inference of CRF using modified Viterbi

function VITERBI(*observations* of len T,*state-graph* of len N) **returns** *best-path*, *path-prob*

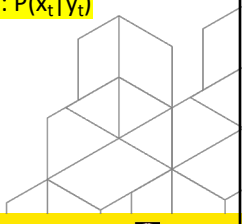
create a path probability matrix *viterbi[N,T]*
for each state s **from** 1 **to** N **do** ; initialization step
  $viterbi[s,1] \leftarrow \pi_s * b_s(o_1)$
  $backpointer[s,1] \leftarrow 0$
for each time step t **from** 2 **to** T **do** ; recursion step
  for each state s **from** 1 **to** N **do**
    $viterbi[s,t] \leftarrow \max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
    $backpointer[s,t] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
$bestpathprob \leftarrow \max_{s=1}^{N} viterbi[s,T]$; termination step
$bestpathpointer \leftarrow \operatorname{argmax}_{s=1}^{N} viterbi[s,T]$; termination step
$bestpath \leftarrow$ the path starting at state *bestpathpointer*, that follows backpointer[] to states back in time
return *bestpath*, *bestpathprob*

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i) \sum_{k=1}^{K} w_k f_k(y_{t-1}, y_t, X, t) \quad 1 \le j \le N, 1 < t \le T$$

Why?

$a_{s',s}$: $P(y_t | y_{t-1})$
$bs(o_t)$: $P(x_t | y_t)$

https://www.cs.columbia.edu/~mcollins/fb.pdf



UNSW SYDNEY

51

A key advantage of CRF-based tagging is its ability to use features

Arbitrary combinations of relevant tags and words in a sentence

e.g. “The” is followed by either an adjective or a noun can be encoded as a feature

Allow model unseen words (because some features may still get detected)

Demo time!

52

UNSW SYDNEY | Australia's Global University

Part 2

CRF -> BiLSTM+CRF

Huang, Zhiheng, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging." *arXiv preprint arXiv:1508.01991* (2015).
Lample, Guillaume, et al. "Neural Architectures for Named Entity Recognition." *Proceedings of NAACL-HLT*. 2016.
Ma, Xuezhe, and Eduard Hovy. "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF." *ACL* 2016.

53

Can we eliminate the need for features?

Neural models replaced human feature engineering

Can sequence tagging be done using neural models? -> Hybrid sequence tagging models

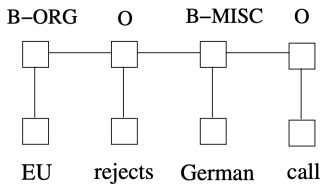


Figure 5: A CRF network.

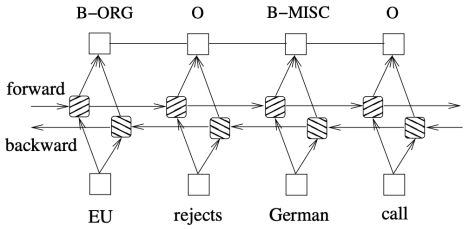


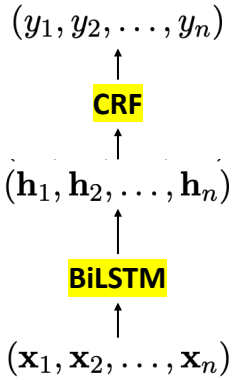
Figure 7: A BI-LSTM-CRF model.

CNNs/RNNs are also possible in the place of BiLSTM.

Combining BiLSTMs with CRF

Bidirectional LSTM encodes hidden state corresponding to every word position.

CRF uses current hidden state as the input representation.



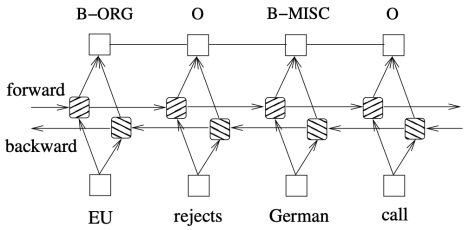


Figure 7: A BI-LSTM-CRF model.

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \qquad \mathbf{y} = (y_1, y_2, \ldots, y_n)$$

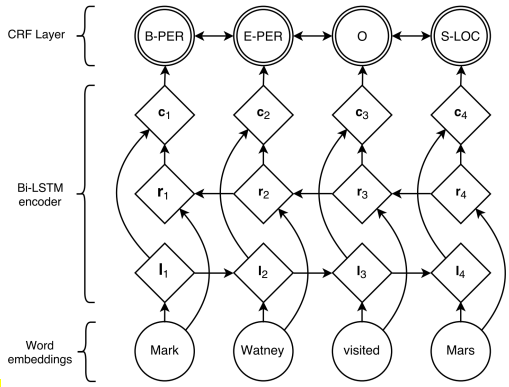
$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{s(\mathbf{X},\mathbf{y})}}{\sum_{\tilde{\mathbf{y}} \in \mathbf{Y}_\mathbf{X}} e^{s(\mathbf{X},\tilde{\mathbf{y}})}}$$

$$s(\mathbf{X}, \mathbf{y}) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i}$$

Transition score

Observation score

The two components, again!

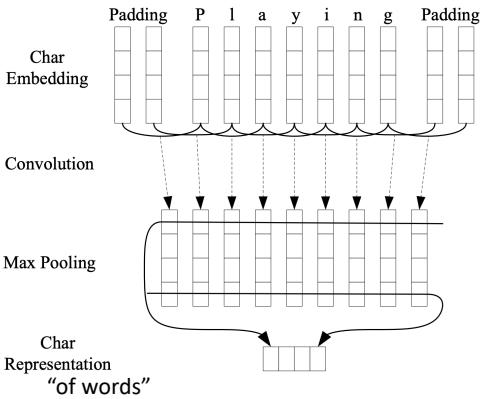


Dot product between c_i and y_i

… embedding dimensions act as features of the observed word!

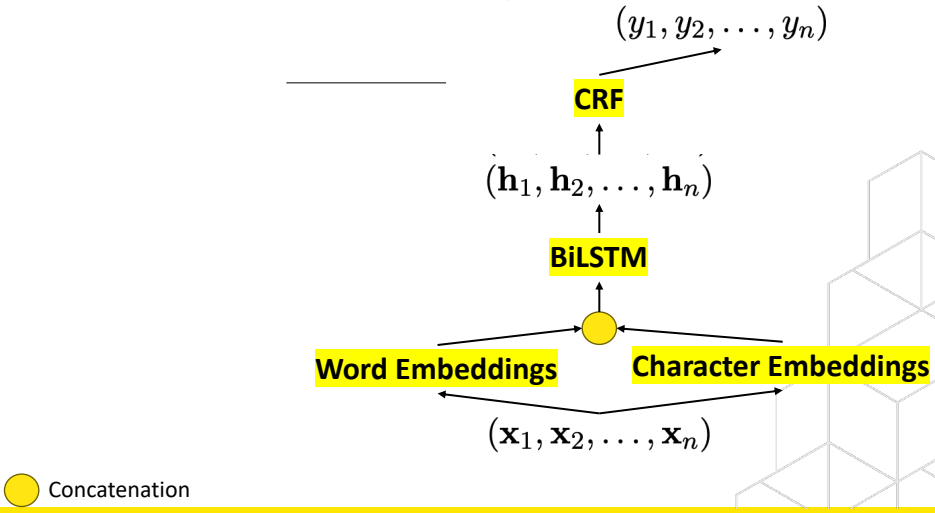
56

….but for POS and NER, sub-word information is valuable!



57

The "complete" diagram



58

Training algorithm: Forward-backward algorithm

Algorithm 1 Bidirectional LSTM CRF model training procedure

```
for each epoch do
    for each batch do
        1) bidirectional LSTM-CRF model forward pass:
            forward pass for forward state LSTM
            forward pass for backward state LSTM
        2) CRF layer forward and backward pass
        3) bidirectional LSTM-CRF model backward pass:
            backward pass for forward state LSTM
            backward pass for backward state LSTM
        4) update parameters
    end for
end for
```

59

https://pytorch.org/tutorials/beginner/nlp/advanced_tutorial.html

Demo time!

60

UNSW SYDNEY | Australia's Global University

Part 3

Special cases of POS Tagging & NER

Owoputi, Olutobi, et al. "Improved part-of-speech tagging for online conversational text with word clusters." Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies. 2013.
Wang, Yu, et al. "Nested named entity recognition: a survey." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16.6 (2022): 1-29.
Nishida, Kosuke, Naoki Yoshinaga, and Kyosuke Nishida. "Self-Adaptive Named Entity Recognition by Retrieving Unstructured Knowledge." EACL 2023.

61

Motivation

Special cases of POS Tagging

Social media text may not obey the rules of grammatical sentences

"Went to the uni today.. spent the day zzzing in the lounge… loolz"

Subject may be dropped

Creative words ("zzzing")

Special cases of NER

Domain-specific named entities

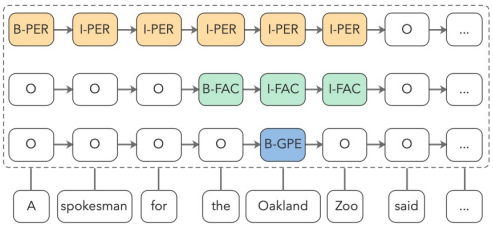
AIMS:To clarify the correlation between chronic sleep restriction (CSR) and sporadic Alzheimer disease (AD), we determined in wild-type mice the impact of CSR on cognitive performance, beta-amyloid (Aβ) peptides, and its feed-forward regulators regarding AD pathogenesis. METHODS:Sixteen nine-month-old C57BL/6 male mice were equally divided into the CSR and control groups. CSR was achieved by application of a slowly rotating drum for 2 months. The Morris water maze test was used to assess cognitive impairment. The concentrations of Aβ peptides, amyloid precursor protein (APP) and β-secretase 1 (BACE1), and the mRNA levels of BACE1 and BACE1-antisense (BACE1-AS) were measured. RESULTS:Following CSR, impairments of spatial learning and memory consolidation were observed in the mice, accompanied by Aβ plaque deposition and an increased Aβ concentration in the prefrontal and temporal lobe cortex. CSR also upregulated the β-secretase-induced cleavage of APP by increasing the protein and mRNA levels of BACE1, particularly the BACE1-AS. CONCLUSIONS:This study shows that a CSR accelerates AD pathogenesis in wild-type mice. An upregulation of the BACE1 pathway appears to participate in both cortical Aβ plaque deposition and memory impairment caused by CSR. BACE1-AS is likely activated to initiate a cascade of events that lead to AD pathogenesis. Our study provides, therefore, a molecular mechanism that links CSR to sporadic AD.

Example article from Europe PMC

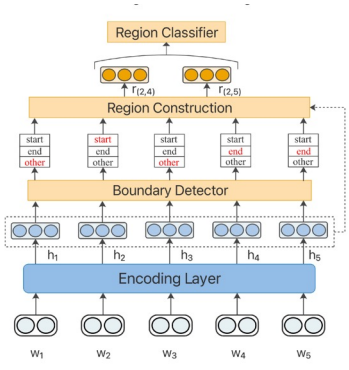
Nested NER: Two approaches

Example: "COVID-19 Moderna vaccine" -> "COVID-19" -> infection; "COVID-19 Moderna Vaccine" -> vaccine

Different tagging task



Region classification



POS tagging of social media text

ikr smh he asked fir yo last name so he can add u on fb lolol

What are some issues you see with POS tagging such text?

Can we use the grammatical set of POS tags?

Can you use specialized features to learn unseen words?

Word clustering

N	common noun
O	pronoun (personal/WH; not possessive)
^	proper noun
S	nominal + possessive
Z	proper noun + possessive
V	verb including copula, auxiliaries
L	nominal + verbal (e.g. *i'm*), verbal + nominal (*let's*)
M	proper noun + verbal
A	adjective
R	adverb
!	interjection
D	determiner
P	pre- or postposition, or subordinating conjunction
&	coordinating conjunction
T	verb particle
X	existential *there*, predeterminers
Y	X + verbal
#	hashtag (indicates topic/category for tweet)
@	at-mention (indicates a user as a recipient of a tweet)
~	discourse marker, indications of continuation across multiple tweets
U	URL or email address
E	emoticon
$	numeral
,	punctuation
G	other abbreviations, foreign words, possessive endings, symbols, garbage

64

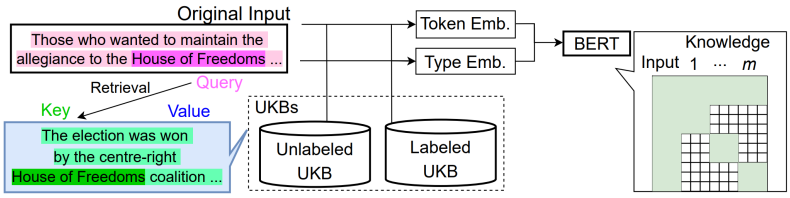
NER for unknown/emerging entities

Self-adaptive NER: Retrieval-augmented language models

Apply NER as per an NER model

If the type of an entity is unknown or the model is 'underconfident',

retrieve information about it using an external source.



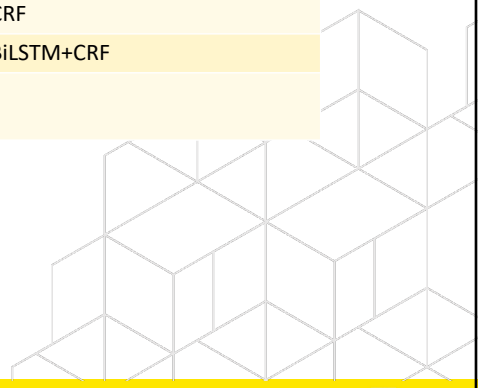
65

Summary

Part	Key Concepts	Demos
Two components of tagging	Tagsets for POS tagging and NER; BERT-based tagging; transition and observation	BERT
HMM + CRF	Generative and discriminative models	CRF
CRF->BiLSTM+CRF	Combining CRF with BiLSTM	BiLSTM+CRF
Specials cases of POS Tagging & NER	Emerging entities, domain-specific NER, nested NER	-

POS Tagging & NER represent sequence tagging tasks in NLP
Can you think of other sequence tagging tasks?

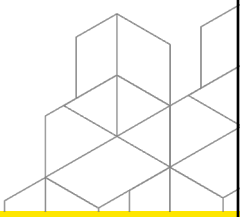
How do NLP models with input and output both as sequences work?
-> Week 8: Machine translation

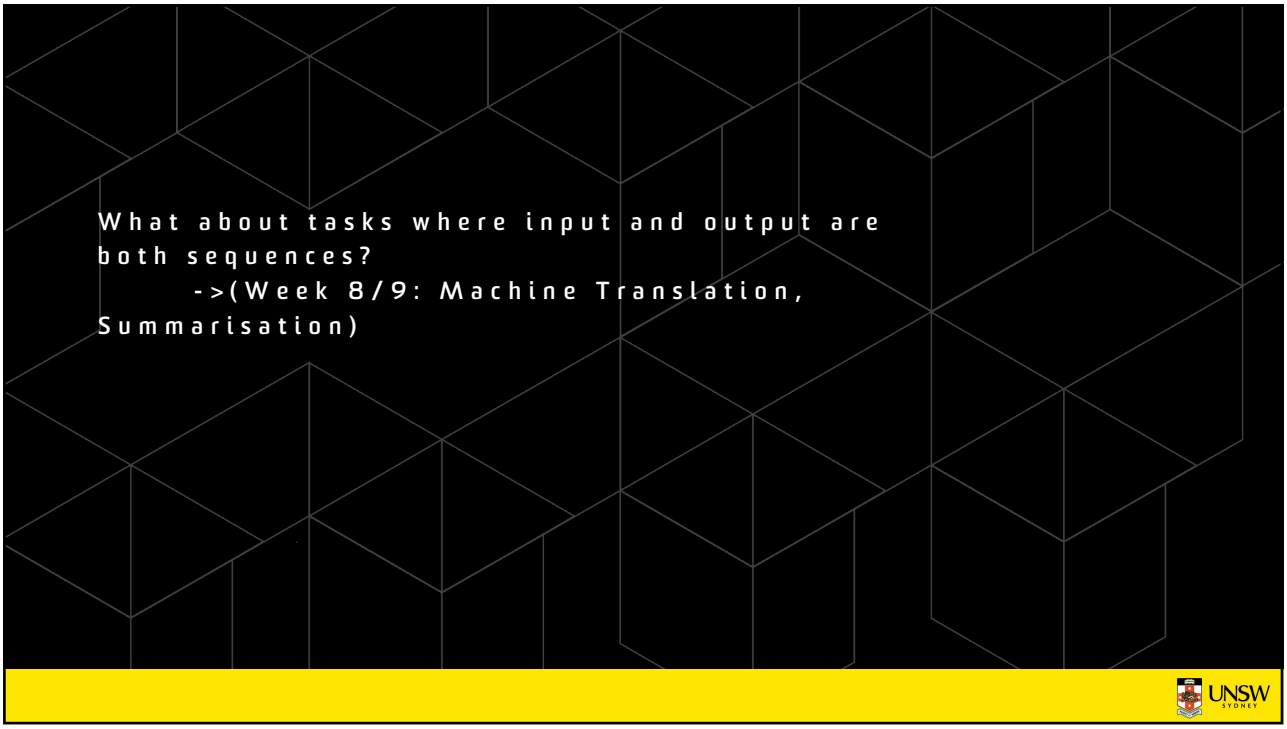


Advanced/Optional Reading

Yadav, Vikas, and Steven Bethard. "A Survey on Recent Advances in Named Entity Recognition from Deep Learning models." *Proceedings of the 27th International Conference on Computational Linguistics*. 2018.

Ehrmann, Maud, et al. "Named entity recognition and classification in historical documents: A survey." *ACM Computing Surveys* 56.2 (2023): 1-47.




What about tasks where input and output are both sequences?
->(Week 8/9: Machine Translation, Summarisation)
UNSW SYDNEY

68