# COMP5046
# Natural Language Processing

*Lecture 9: Named Entity Recognition and Coreference Resolution*




**Dr. Caren Han**

*Semester 1, 2021*
*School of Computer Science,*
*University of Sydney*

# 0 LECTURE PLAN

**Lecture 9: Named Entity Recognition and Coreference**

1. Information Extraction
2. Named Entity Recognition (NER) and Evaluation
3. Traditional NER
4. Sequence Model for NER
5. Coreference Resolution
6. Coreference Model
7. Coreference Evaluation
8. Preview

# Information Extraction

THE UNIVERSITY OF SYDNEY

## Information Extraction

*“The task of automatically* ***extracting structured information*** *from unstructured and/or semi-structured machine-readable documents”*



Here are some questions..



- How to allow computation to be done on the unstructured data
- How to extract clear, factual information
- How to put in a semantically precise form that allows further inferences to be made by computer algorithms

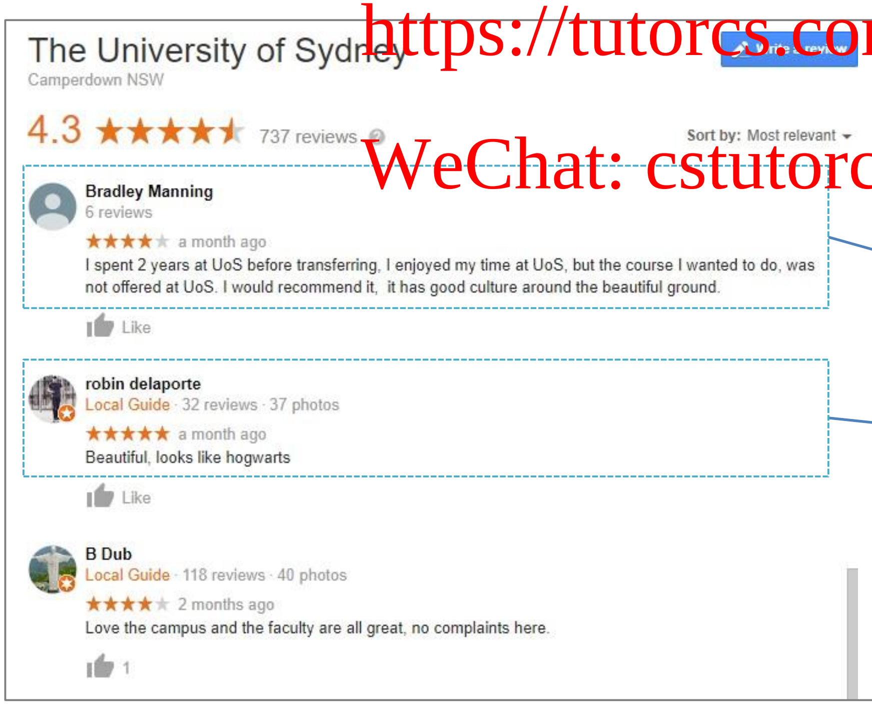
# 1 Information Extraction

## How to extract the structured clear, factual information

- Find and understand limited relevant parts of texts
- Gather information from many pieces of text
- Produce a structured representation of relevant information

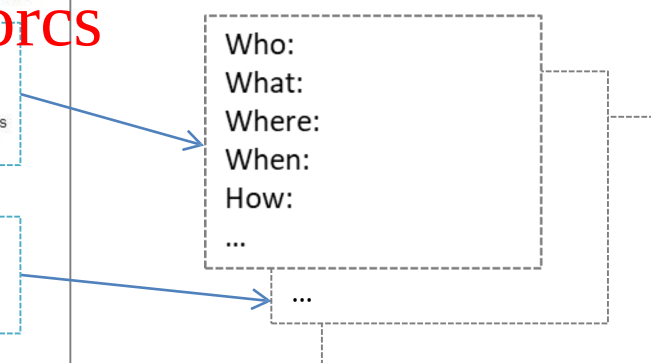
*relations (in the database sense) or a knowledge base*






***"5W1H"***

*who, what, where, when, why, how*



# 1 Information Extraction



## How to extract the structured clear, factual information

- Find and understand limited relevant parts of texts
- Gather information from many pieces of text
- Produce a structured representation of relevant information



***relations (in the database sense) or a knowledge base***






*Extracting*

| Subject | Relation | Object |
|---|---|---|
| Sydney | IS-A | Capital of New South Wales |
| Sydney | IS-A | Australia's largest cities |
| Sydney | KNOWN FOR | Sydney Opera House |
| ... | ... | ... |

*Textual abstract:* ***Summary for human***

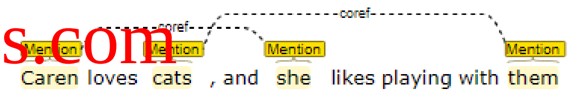
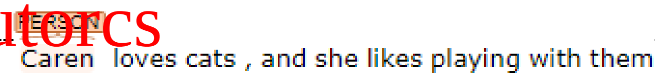
*Structured information:* ***Summary for machine***

# 1 Information Extraction

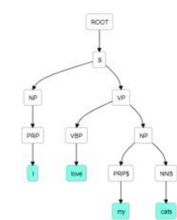
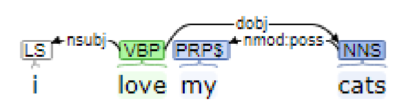
## Information Extraction Pipeline with NLP

**Understanding**

*Application*

| | | |
|---|---|---|
| ***Sentiment Analysis*** | Caren loves cats, and she likes playing with them | **[positive: 90.10%] [neutral: 4.70%] [negative: 5.10%]** |
| ***Coreference Resolution*** | Caren loves cats, and she likes playing with them |  |
| ***Entity Extraction*** | Caren loves cats, and she likes playing with them |  |

*NLP Stack*

| | | |
|---|---|---|
| ***Parsing*** | I love my cats |   |
| ***PoS Tagging*** | I love my cats | **[I/JJ] [love/VBP] [my/PRP] [cats/NNS]** |
| ***Stemming*** | I love my cats | **[I] [love] [my] [cat]** |
| ***Tokenisation*** | I love my cats | **[I] [love] [my] [cats]** |

# Named Entity Recognition (NER)

**What is Named Entity Recognition?**

*"The subtask of information extraction that seeks to locate and classify named entity mentions in unstructured text into pre-defined categories such as the person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc."*



Why recognise Named Entities?

- Named entities can be indexed, linked off, etc.
- Sentiment can be attributed to companies or products
- A lot of relations are associations between named entities
- For question answering, answers are often named entities.

# Named Entity Recognition (NER)



## How to recognize Named Entities?

**Identify** and **classify** names in text

- *The University of Sydney (informally USYD, Sydney, Sydney Uni) is an Australian public research university in Sydney, Australia. Founded in 1850, it was Australia's first university and is regarded as one of the world's leading universities. (Wikipedia, University of Sydney)*



Different types of named entity classes

| Type | Classes |
|---|---|
| 3 class | Location, Person, Organization |
| 4 class | Location, Person, Organization, Misc |
| 7 class | Location, Person, Organization, Money, Percent, Date, Time |

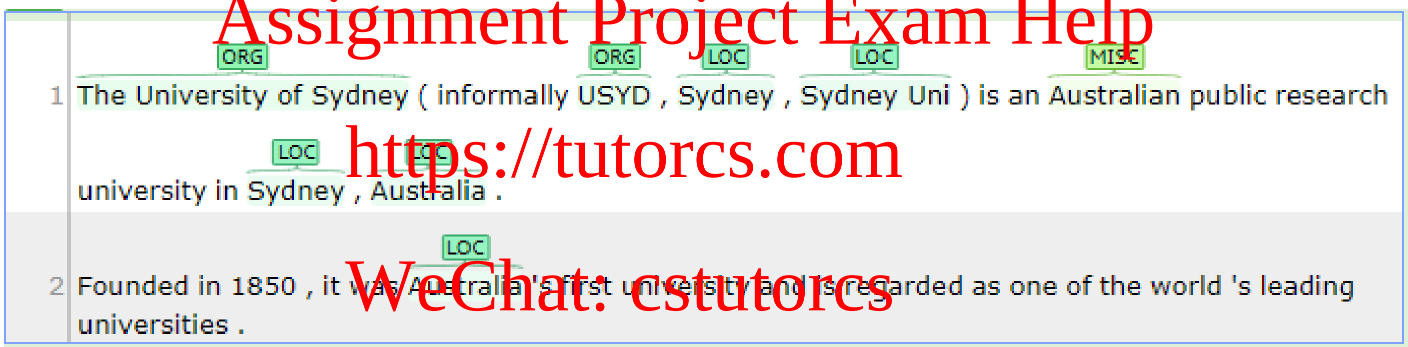
**classes can be different based on annotated dataset*

# 2 Named Entity Recognition (NER)

## How to recognize Named Entities?

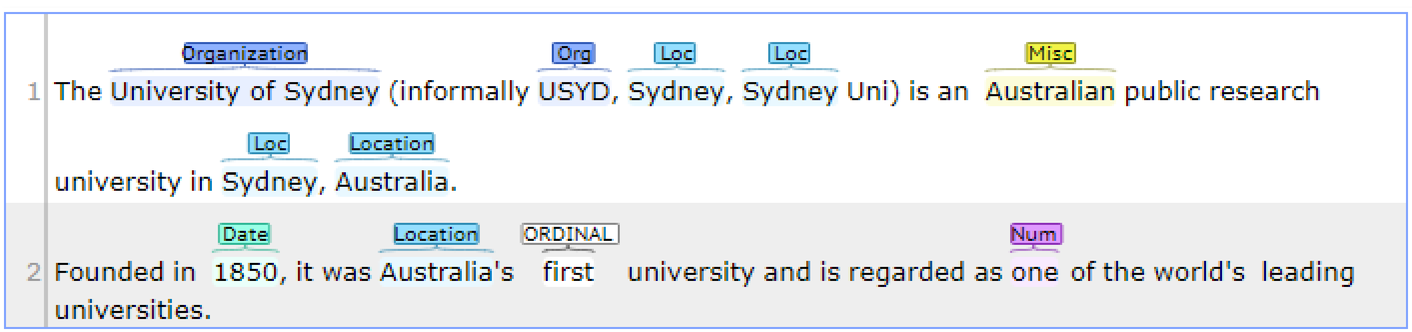
**Identify** and **classify** names in text

***Upenn CogComp-NLP*** *http://macniece.seas.upenn.edu:4004/*






***Stanford CoreNLP 3.9.2*** *http://nlp.stanford.edu:8080/corenlp/process*



# 2 Named Entity Recognition (NER)

**How to evaluate the NER performance?**

The goal: ***predicting entities in a text***

***Standard evaluation is per entity, not per token**

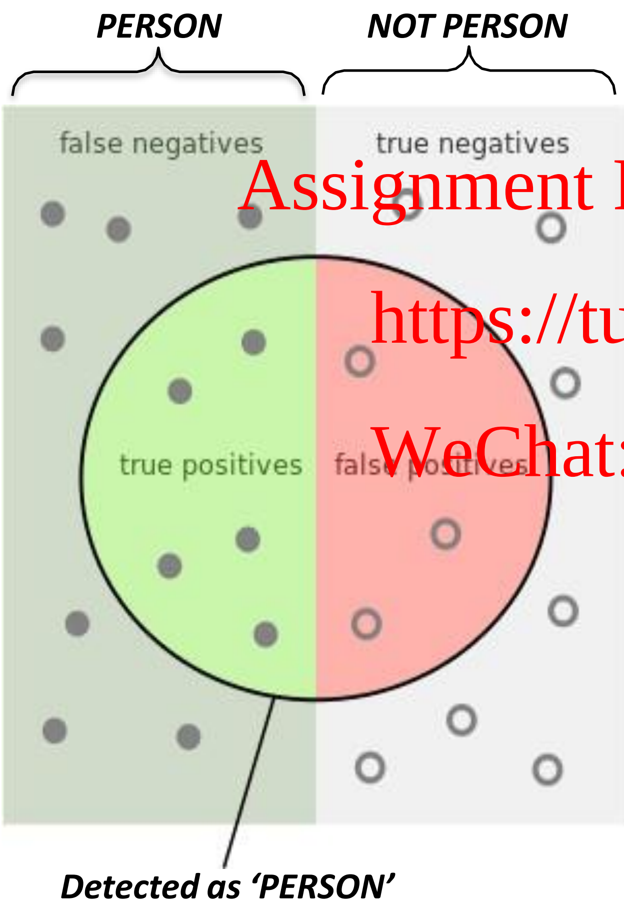


| | Caren | Soyeon | Han | is | working | at | Google | at | Sydney, | Australia |
|---|---|---|---|---|---|---|---|---|---|---|
| ***gold*** | PER | PER | PER | O | O | O | ORG | O | LOC | LOC |
| ***predicted*** | O | O | O | O | O | O | ORG | O | LOC | LOC |

# 2 Named Entity Recognition (NER)



## How to evaluate the NER performance? Precision and recall



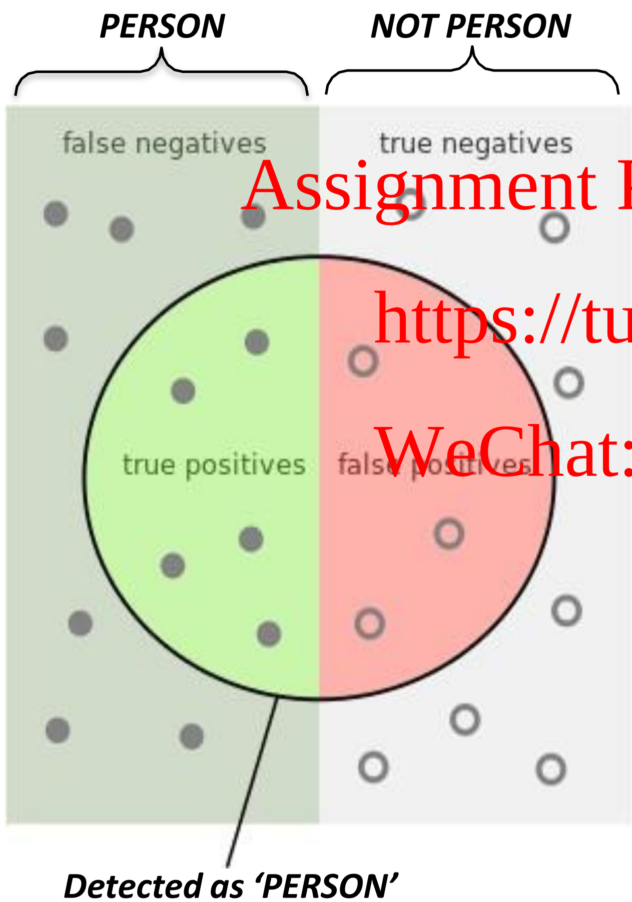



Precision = *Detected as ‘PERSON’ correctly* / *Total number of detected as ‘PERSON’*

Recall = *Detected as ‘PERSON’ correctly* / *Total number of actual ‘PERSON’ entities*

# 2 Named Entity Recognition (NER)

## How to evaluate the NER performance? Precision and recall




True positives: The 'PERSON's that the model detected as 'PERSON'

False positives: The NOT 'PERSON's that the model detected as 'PERSON'

False negatives: The 'PERSON's that the model detected as NOT 'PERSON'

True negatives: The 'NOT PERSON's that the model detected as NOT 'PERSON'

# 2 Named Entity Recognition (NER)

**How to evaluate the NER performance?**

The goal: ***predicting entities in a text***

***Standard evaluation is per entity, not per token**



| | Caren | Soyeon | Han | is | working | at | Google | at | Sydney, | Australia |
|---|---|---|---|---|---|---|---|---|---|---|
| *gold* | PER | PER | PER | O | O | O | ORG | O | LOC | LOC |
| *predicted* | O | O | O | O | O | O | ORG | O | LOC | LOC |



| | correct | not correct |
|---|---|---|
| selected | True Positive (TP) | False Positive (FP) |
| not selected | False Negative (FN) | True Negative (TN) |

# Named Entity Recognition (NER)

**How to evaluate the NER performance?**

The goal: ***predicting entities in a text***

***Standard evaluation is per entity, not per token**



| | Caren | Soyeon | Han | is | working | at | Google | at | Sydney, | Australia |
|---|---|---|---|---|---|---|---|---|---|---|
| ***gold*** | PER | PER | PER | O | O | O | ORG | O | LOC | LOC |
| ***predicted*** | O | O | O | O | O | O | ORG | O | LOC | LOC |



***Precision and Recall are straightforward for text categorization or web search, where there is only one grain size (documents)***

**Quick Exercise: F measure Calculation**

Let's calculate Precision, Recall, and F-measure together!



$P = ??$ $R = ??$ $F_1 = ??$

$$F1 = 2 * \frac{P*R}{P+R}$$



| | correct | not correct |
|---|---|---|
| selected | 2 (TP) | 0 (FP) |
| not selected | 1 (FN) | 0 (TN) |

# 2 Named Entity Recognition (NER)

## Data for learning named entity

- Training counts joint frequencies in a corpus
- The more training data, the better
- Annotated corpora are small and expensive



| Corpora | Source | Size | Class Type |
|---|---|---|---|
| muc-7 | New York Times | 164k tokens | per, org, loc, dates, times, money, percent https://aclweb.org/aclwiki/MUC-7_(State_of_the_art) |
| conll-03 | Reuters | 301k | per, org, loc, misc |
| bbn | Wall Street Journal | 1174k | https://catalog.ldc.upenn.edu/docs/LDC2005T33/BBN-Types-Subtypes.html |

# 2 Named Entity Recognition (NER)

## Data for learning named entity

- Models trained on one corpus perform poorly on others



| ***train*** | ***F-score*** | | |
|---|---|---|---|
| | ***muc*** | ***conll*** | ***bbn*** |
| ***muc*** | 82.3 | 54.9 | 69.3 |
| ***conll*** | 69.9 | 86.9 | 60.2 |
| ***bnn*** | 80.2 | 58.0 | 88.0 |

# 2 Named Entity Recognition (NER)

## CoNLL 2003 NER dataset

- Performance measure: F = 2 * Precision * Recall / (Recall + Precision)



| RANK | MODEL | F1 | EXTRA TRAINING DATA | PAPER | CODE | RESULT | YEAR |
|---|---|---|---|---|---|---|---|
| 1 | LUKE | 94.3 | × | LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention | | | 2020 |
| 2 | ACE + document-context | 94.14 | × | Automated Concatenation of Embeddings for Structured Prediction | | | 2020 |
| 3 | Cross-sentence context (First) | 93.74 | × | Exploring Cross-sentence Contexts for Named Entity Recognition with BERT | | | 2020 |
| 4 | ACE | 93.64 | × | Automated Concatenation of Embeddings for Structured Prediction | | | 2020 |
| 5 | CNN Large + fine-tune | 93.5 | ✓ | Cloze-driven Pretraining of Self-attention Networks | | | 2019 |
| 6 | Biaffine-NER | 93.5 | × | Named Entity Recognition as Dependency Parsing | | | 2020 |
| 7 | GCDT + BERT-L | 93.47 | ✓ | GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling | | | 2019 |
| 8 | I-DARTS + Flair | 93.47 | ✓ | Improved Differentiable Architecture Search for Language Modeling and Named Entity Recognition | | | 2019 |
| 9 | CrossWeigh + Pooled Flair | 93.43 | × | CrossWeigh: Training Named Entity Tagger from Imperfect Annotations | | | 2019 |
| 10 | LSTM-CRF+ELMo+BERT+Flair | 93.38 | ✓ | Neural Architectures for Nested NER through Linearization | | | 2019 |

https://paperswithcode.com/sota/named-entity-recognition-ner-on-conll-2003

# 2 Named Entity Recognition (NER)

## Datasets for NER in English

The following table shows the list of datasets for English entity recognition.

| Dataset | Domain | License | Reference |
|---|---|---|---|
| CONLL 2003 | News | DUA | Sang and Meulder, 2003 |
| NIST-IEER | News | None | NIST 1999 IE-ER |
| MUC-6 | News | LDC | Grishman and Sundheim, 1996 |
| OntoNotes 5 | Various | LDC | Weischedel et al., 2013 |
| BBN | Various | LDC | Weischedel and Brunstein, 2005 |
| GMB-1.0.0 | Various | None | Bos et al., 2017 |
| GUM-3.1.0 | Wiki | Several (*2) | Zeldes, 2016 |
| wikigold | Wikipedia | CC-BY 4.0 | Balasuriya et al., 2009 |
| Ritter | Twitter | None | Ritter et al., 2011 |
| BTC | Twitter | CC-BY 4.0 | Derczynski et al., 2016 |
| WNUT17 | Social media | CC-BY 4.0 | Derczynski et al., 2017 |
| i2b2-2006 | Medical | DUA | Uzuner et al., 2007 |
| i2b2-2014 | Medical | DUA | Stubbs et al., 2015 |
| CADEC | Medical | CSIRO | Karimi et al., 2015 |
| AnEM | Anatomical | CC-BY-SA 3.0 | Ohta et al., 2012 |
| MITRestaurant | Queries | None | Liu et al., 2013a |
| MITMovie | Queries | None | Liu et al., 2013b |
| MalwareTextDB | Malware | None | Lim et al., 2017 |
| re3d | Defense | Several (*1) | DSTL, 2017 |
| SEC-filings | Finance | CC-BY 3.0 | Alvarado et al., 2015 |
| Assembly | Robotics | X | Costa et al., 2017 |



***DUA**: Data Use Agreement*
***LDC**: Linguistic Data Consortium*
***CC-BY 4.0**: Creative Commons Attribution 4.0*

https://github.com/juand-r/entity-recognition-datasets

https://paperswithcode.com/task/named-entity-recognition-ner

# 3 Traditional NER

### Three standard approaches to NER

- Rule-based NER
- Classifier-based NER
- Sequence Model for NER

Traditional Approaches

# 3 Traditional NER

## Rule-based NER

- Entity references have internal and external language cues

  Mr. [per Scott Morrison] flew to [loc Beijing]



- Can recognise names using lists (or gazetteers):
  - Personal titles: Mr, Miss, Dr, President
  - Given names: Scott, David, James
  - Corporate suffixes: & Co., Corp., Ltd.
  - Organisations: Microsoft, IBM, Telstra



- and rules:
  - personal title X ⇒ per
  - X, location ⇒ loc or org
  - travel verb to X ⇒ loc

- Effectively regular expressions, PoS Tagger

# 3 Traditional NER

## Rule-based NER

- Determining which person holds what office in what organization
  - [person] , [office] of [org]
    - Michael Spence, the vice-chancellor and principal of the University of Sydney
  - [org] (named, appointed, etc.) [person] Prep [office]
    - WHO appointed Tedros Adhanom as Director-General



- Determining where an organization is located
  - [org] in [loc]
    - Google headquarters in California
  - [org] [loc] (division, branch, headquarters, etc.)
    - Google London headquarters

# 3 Traditional NER

**Statistical approaches are more portable**

- Learn NER from annotated text
  - weights (≈ rules) calculated from the corpus
  - same machine learner, different language or domain



- Token-by-token classification (with any machine learning)



- Each token may be:



  - not part of an entity (tag o)
  - beginning an entity (tag b-per, b-org, etc.)
  - continuing an entity (tag i-per, i-org, etc.)

- What about N-gram model?

# 3 Traditional NER

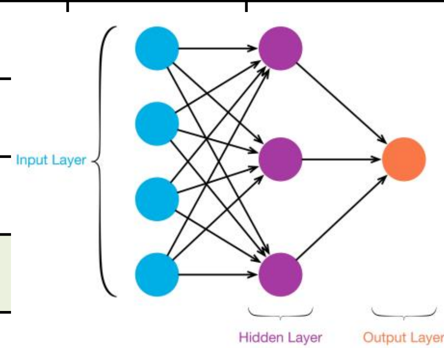
## Various features for statistical NER

| | | | | | | |
|---|---|---|---|---|---|---|
| Unigram | Mr. | Scott | Morrison | flew | to | Beijing |
| Lowercase unigram | mr. | scott | morrison | flew | to | beijing |
| POS tag | nnp | nnp | nnp | vbd | to | nnp |
| length | 3 | 5 | 4 | 4 | 2 | 7 |
| In first-name gazetteer | no | yes | no | no | no | no |
| In location gazetteer | no | no | no | no | no | yes |
| 3-letter suffix | Mr. | ott | son | lew | - | ing |
| 2-letter suffix | r. | tt | on | ew | to | ng |
| 1-letter suffix | . | t | n | w | o | g |
| Tag predictions | O | B-per | I-per | O | O | B-loc |

# 3 Traditional NER

## Various features for statistical NER

| Unigram | Mr. | Scott | Morrison | flew | to | Beijing |
|---|---|---|---|---|---|---|
| Lowercase unigram | mr. | scott | morrison | flew | to | beijing |
| POS tag | nnp | nnp | nnp | vbd | to | nnp |
| length | 3 | 5 | 4 | 4 | 2 | 7 |
| In first-name gazetteer | no | yes | no | no | no | no |
| In location gazetteer | no | no | no | no | no | yes |
| 3-letter suffix | Mr. | | | lew | - | ing |
| 2-letter suffix | r. | | | ew | to | ng |
| 1-letter suffix | . | | | w | o | g |
| Tag predictions | O | | | O | O | B-loc |




Mr. Scott Morrison lives in Sydney ---> ***Predictive Model*** ---> O B-PER I-PER O O B-LOC

# Traditional NER



**Traditional NER Approaches - Pros and Cons**

***Rule-based approaches***

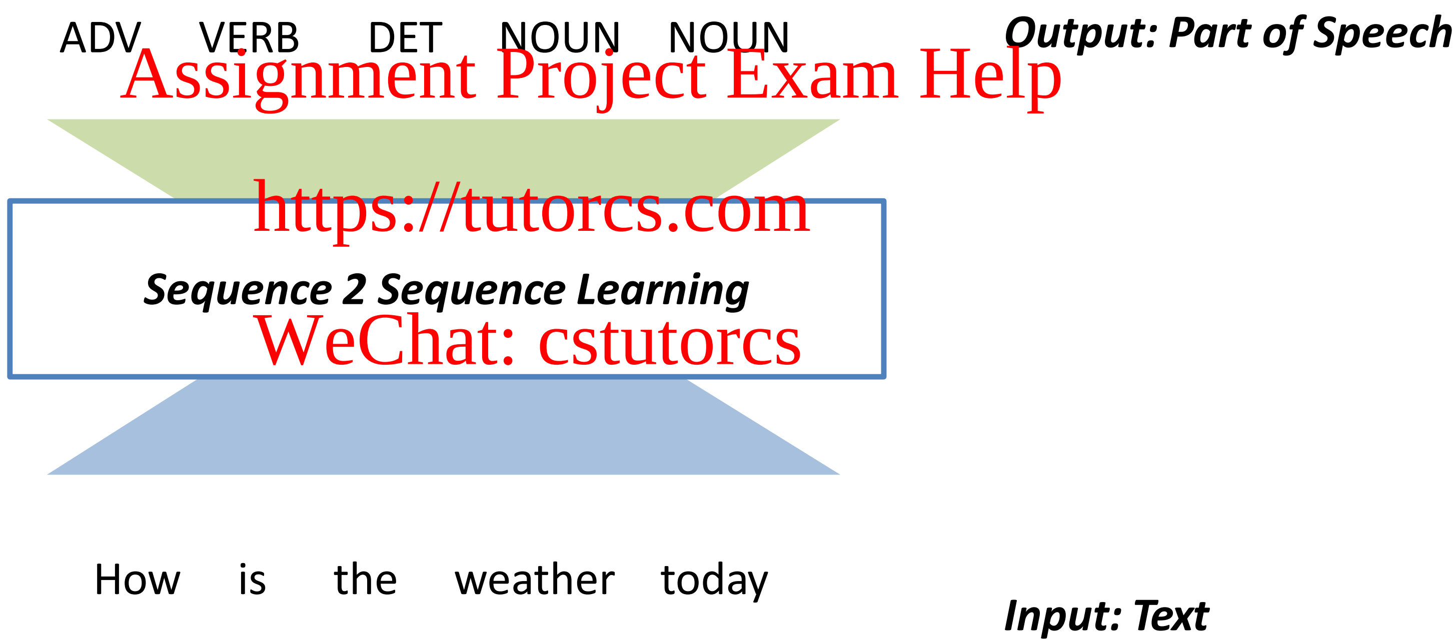
- Can be high-performing and efficient
- Require experts to make rules
- Rely heavily on gazetteers that are always incomplete
- Are not robust to new domains and languages

***Statistical approaches***

- Require (expert-)annotated training data
- May identify unforeseen patterns
- Can still make use of gazetteers
- Are robust for experimentation with new features
- Are largely portable to new languages and domains

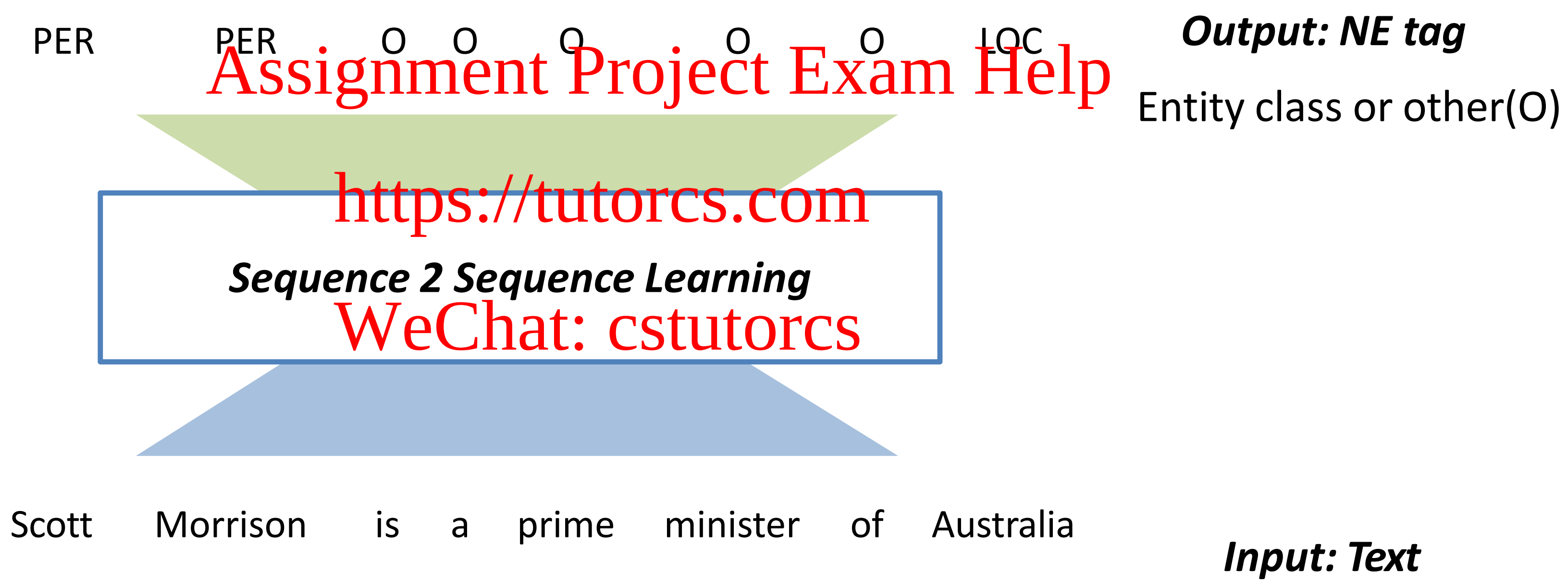
# 4 Sequence Model for NER

### Sequence Model (N to N)



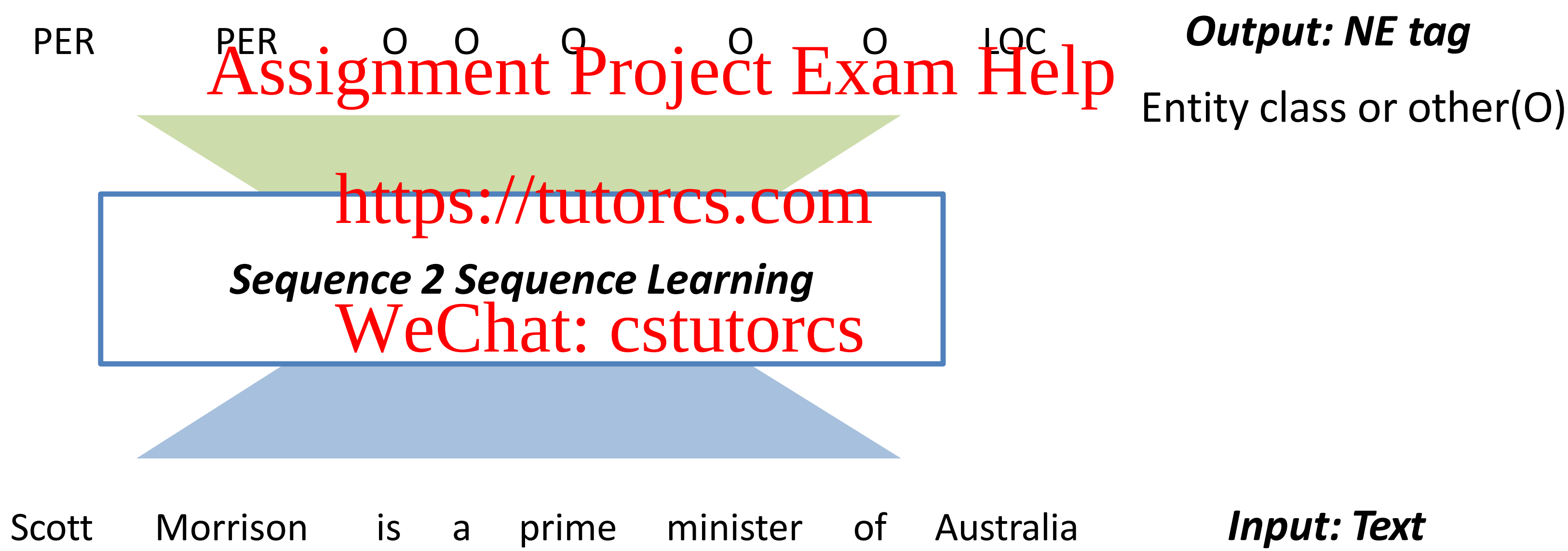
# 4 Sequence Model for NER

## Sequence Model



# 4 Sequence Model for NER

## Encoding classes for sequence labeling






*The IOB (short for inside, outside, beginning) is a common tagging format*

- I- prefix before a tag indicates that the tag is inside a chunk.
- B- prefix before a tag indicates that the tag is the beginning of a chunk.
- An O tag indicates that a token belongs to no chunk (outside).

# 4 Sequence Model for NER

## Encoding classes for sequence labeling

The IO and IOB (inside, outside, beginning) is a common tagging format

| | Josiah | tells | Caren | John | Smith | is | a | student | |
|---|---|---|---|---|---|---|---|---|---|
| IO encoding | PER | O | PER | PER | PER | O | O | O | **n+1** |
| IOB encoding | B-PER | O | B-PER | B-PER | I-PER | O | O | O | **2n+1** |
| | | ***even*** | B-PER | I-PER | I-PER | | | | |



***IO encoding vs IOB encoding***

- *Computation Time?*
- *Efficiency?*

# 4 Sequence Model for NER

## Features for sequence labeling

**Words**

- Current word (essentially like a learned dictionary)
- Previous/next word (context)



**Other kinds of inferred linguistic classification**



- Part-of-speech tags



**Label context**

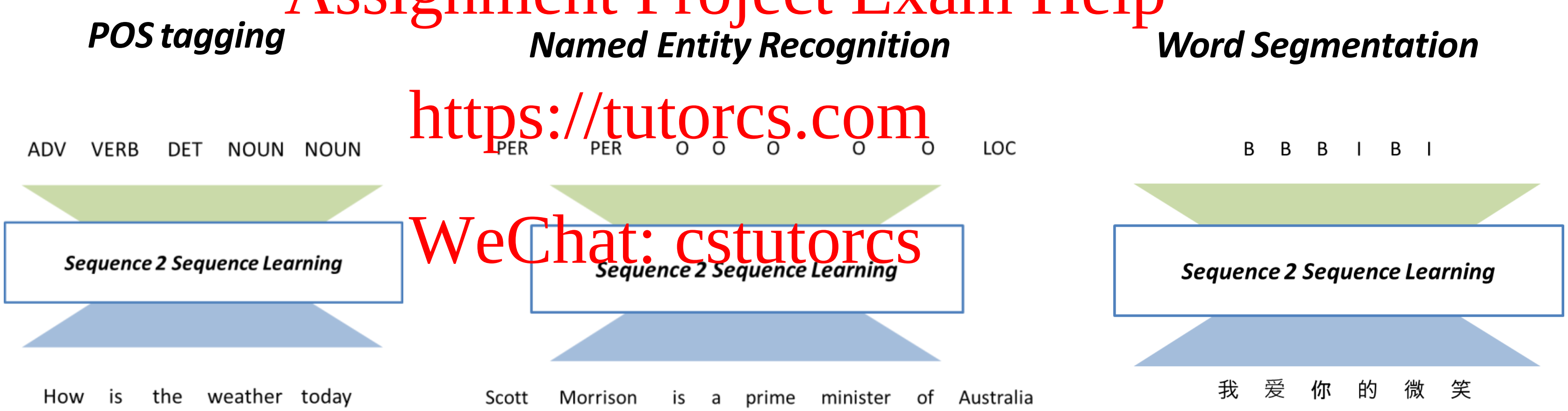
- Previous (and perhaps next) label

# 4 Sequence Model for NER

## N to N Sequence model

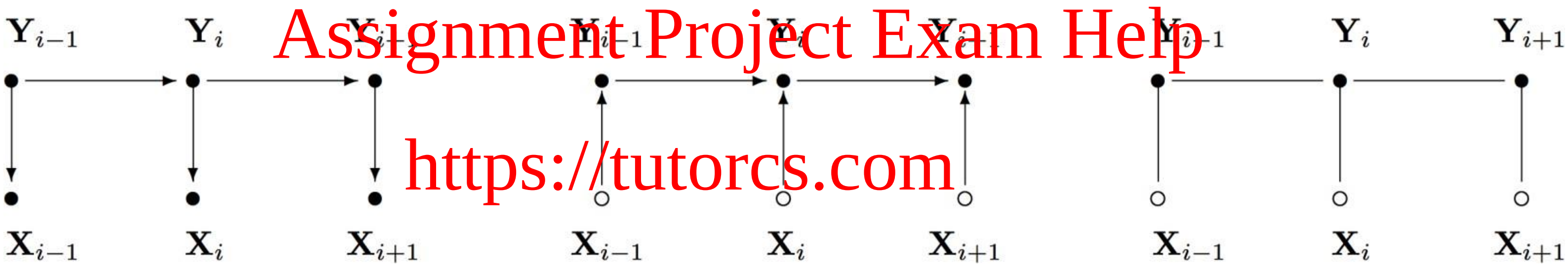
- There are different NLP tasks that used N to N sequence model





# Sequence Model for NER

## Sequence Model (MEMM, CRF)




**HMM** **MEMM** **CRF**

# 4 Sequence Model for NER

## Sequence Inference for NER

- For a Maximum Entropy Markov Model (MEMM), the classifier makes a single decision at a time, conditioned on evidence from observations and previous decisions.



| -3 | -2 | -1 | 0 | +1 |
|---|---|---|---|---|
| Scott | Morrison | lives | in | Australia |
| **NN** | **NN** | **VBZ** | **IN** | **NN** |

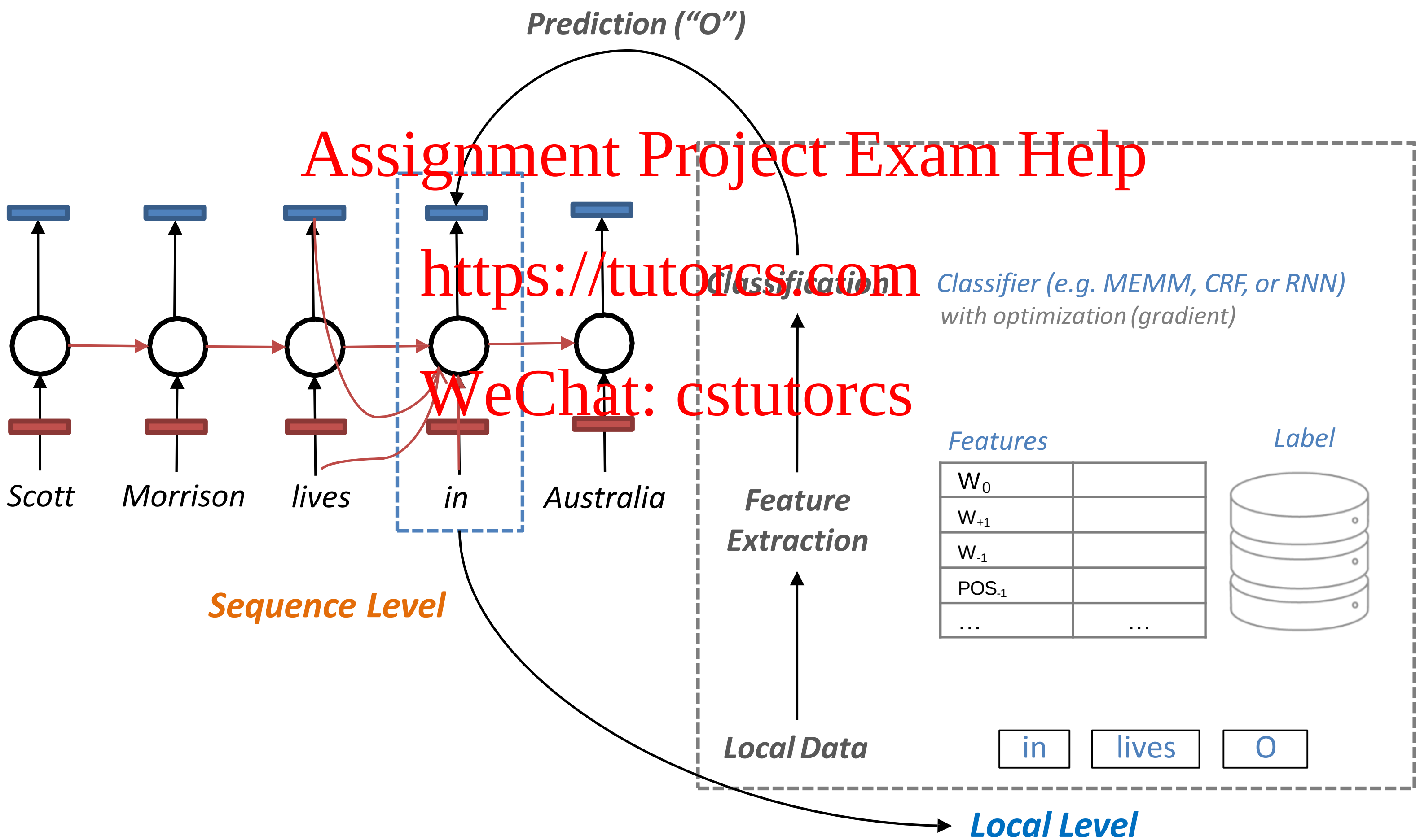
***Features***

| | |
|---|---|
| $W_0$ | in |
| $W_{+1}$ | Australia |
| $W_{-1}$ | lives |
| $POS_{-1}$ | VBZ |
| $POS_{-2}$-$POS_{-1}$ | NN - VBZ |
| hasDigit? | 0 |
| … | … |

(Toutanova et al. 2003, etc.)

# 4 Sequence Model for NER

## Sequence Inference for NER



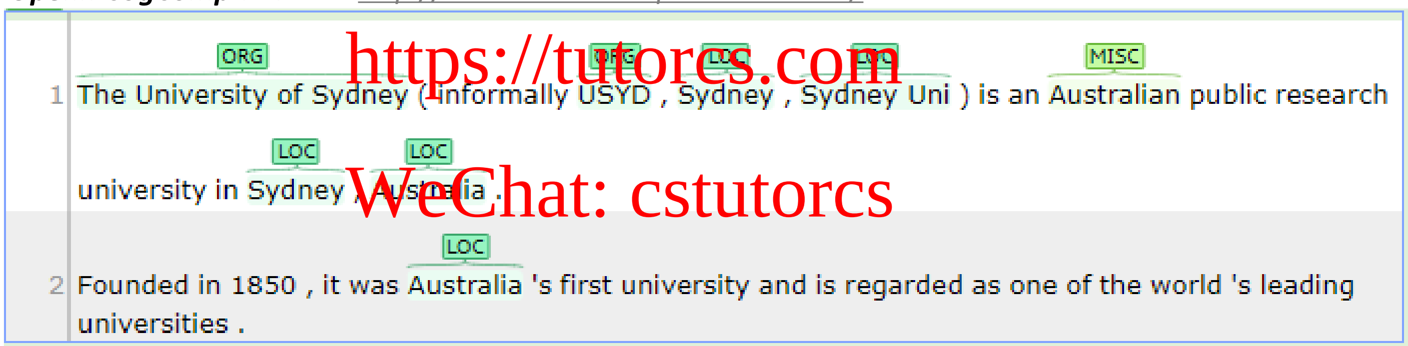
# 4 Sequence Model for NER

## Named Entity Recognition

*The goal: predicting named entity mentions in unstructured text into pre-defined categories  such as the person names, organizations, locations*



***Upenn CogComp-NLP*** *http://macniece.seas.upenn.edu:4004/*





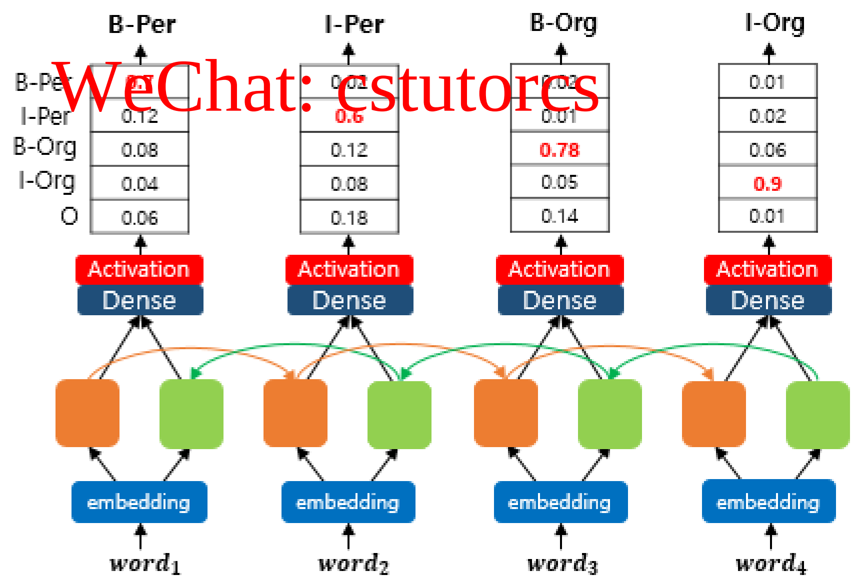

| | Caren | Soyeon | Han | is | working | at | Google | at | Sydney, | Australia |
|---|---|---|---|---|---|---|---|---|---|---|
| ***gold*** | PER | PER | PER | O | O | O | ORG | O | LOC | LOC |
| ***predicted*** | O | O | O | O | O | O | ORG | O | LOC | LOC |

# 4 Sequence Model for NER

### Named Entity Recognition with Bi-LSTM

*We can easily apply Bi-LSTM (N to N Seq2Seq) Model to predict Named Entities*





# Sequence Model for NER



**Named Entity Recognition with Bi-LSTM**

*We can easily apply Bi-LSTM (N to N Seq2Seq) Model to predict Named Entities*

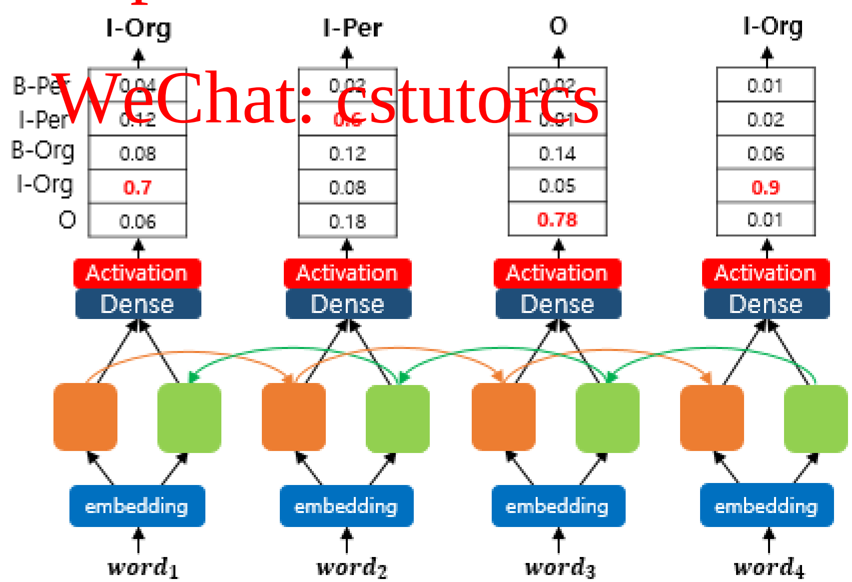
*The model clearly contains incorrect predictions.



'I' cannot appear in the label of the first word. I-Per can only appear after B-Per. I-Org can also appear only after B-Org.







# 4 Sequence Model for NER

## Named Entity Recognition with Bi-LSTM

*We can easily apply Bi-LSTM (N to N Seq2Seq) Model to predict Named Entities*

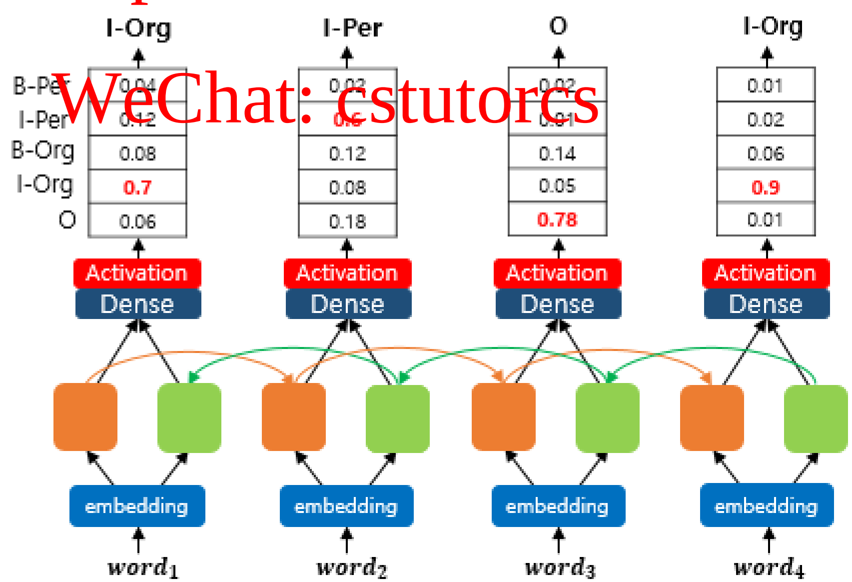


*The model clearly contains incorrect predictions.

‘I’ cannot appear in the label of the first word. I-Per can only appear after B-Per.
I-Org can also appear only after B-Org.







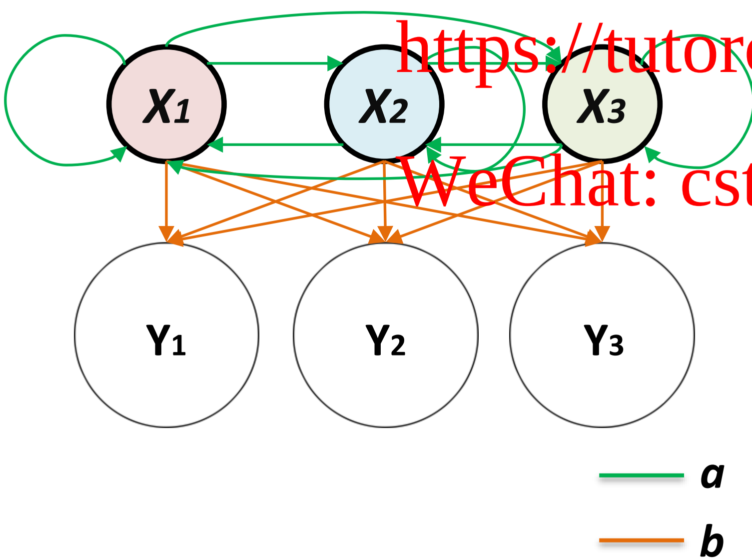

What if we teach the dependency between predicted entity names

# 4 Sequence Model for NER

**Wait? What about HMM?**

*Hidden Markov Models (HMMs) are a class of probabilistic graphical model that allow us to predict a sequence of unknown (hidden) variables from a set of observed variables.*






hidden

***x*** states

***y*** possible observations

***a*** state transition probabilities

***b*** output probabilities

- States are ***hidden***
- ***Observable outcome*** linked to states
- Each state has ***observation probabilities*** to determine the observable event

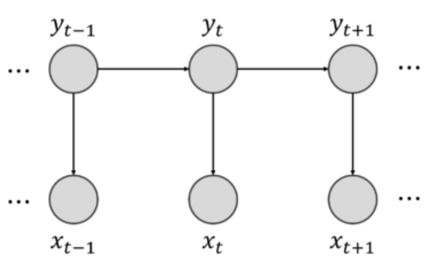
# 4 Sequence Model for NER

## Advanced HMM (MEMM or CRF)

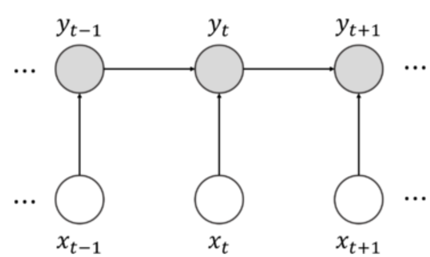
- The CRF model has addressed the labeling bias issue and eliminated unreasonable hypotheses in HMM.
- MEMM adopts local variance normalization, while CRF adopts global variance normalization.



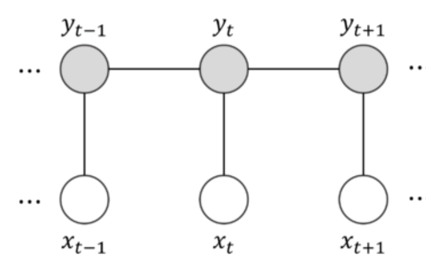
***HMM***

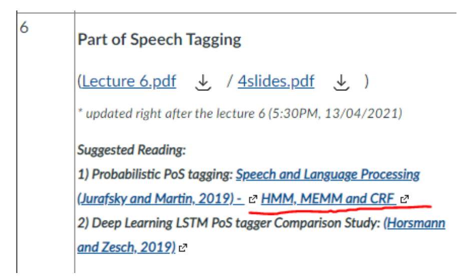



***Maximum-entropy Markov model (MEMM)***




***Conditional random field (CRF)***





6 Part of Speech Tagging

(Lecture 6.pdf / 4slides.pdf )

* updated right after the lecture 6 (5:30PM, 13/04/2021)

*Suggested Reading:*

*1) Probabilistic PoS tagging: Speech and Language Processing (Jurafsky and Martin, 2019) - HMM, MEMM and CRF*

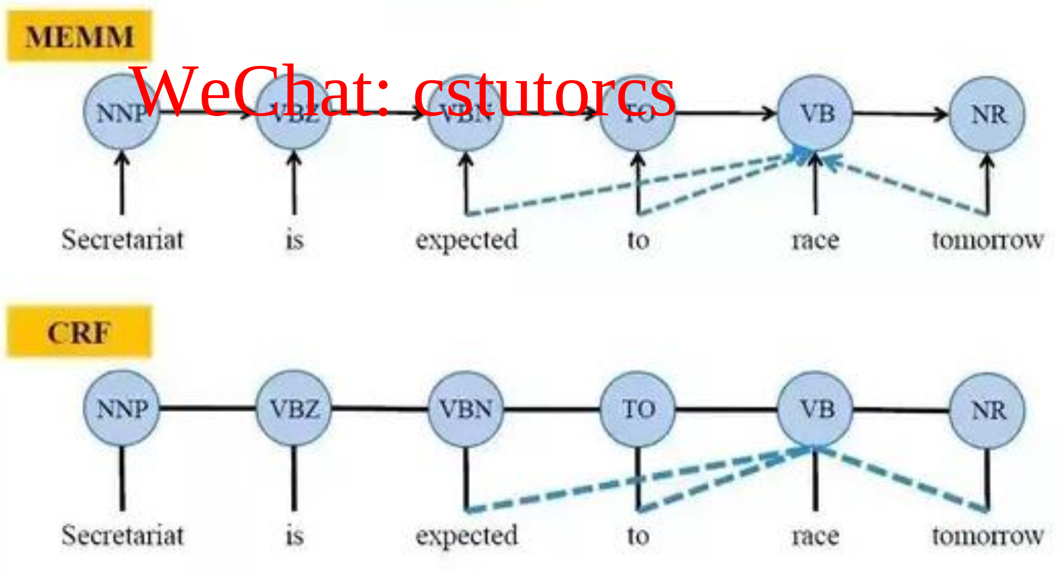
*2) Deep Learning LSTM PoS tagger Comparison Study: (Horsmann and Zesch, 2019)*

# 4 Sequence Model for NER

**Advanced HMM (MEMM or CRF)**

- The CRF model has addressed the labeling bias issue and eliminated unreasonable hypotheses in HMM.
- MEMM adopts local variance normalization, while CRF adopts global variance normalization.



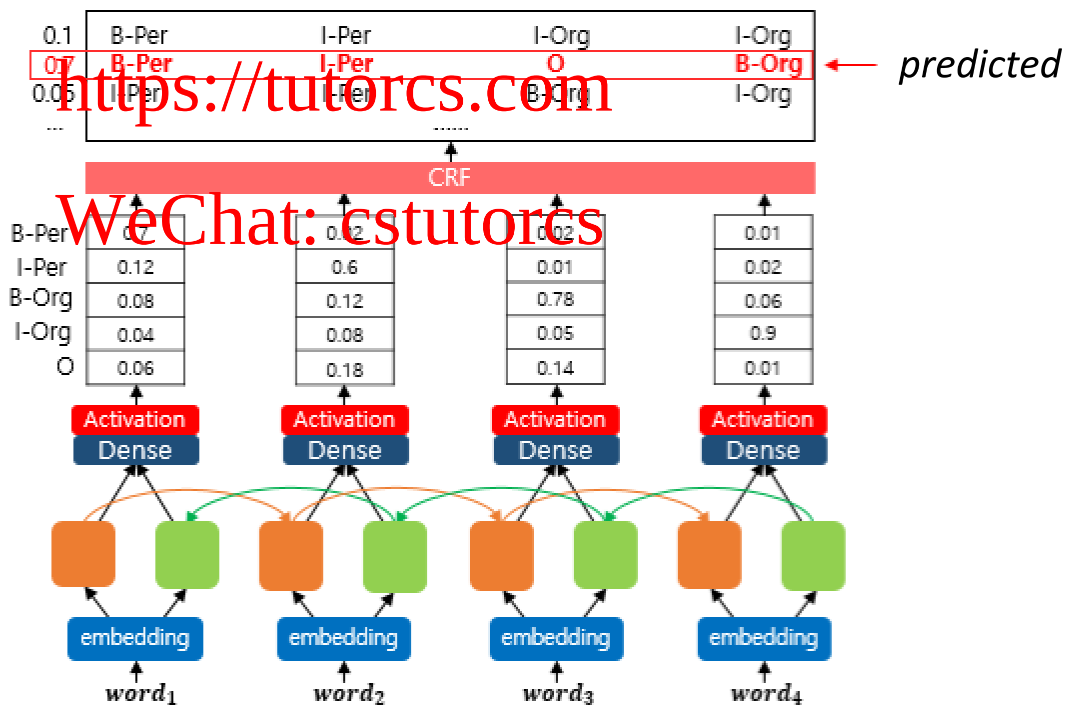


# 4 Sequence Model for NER

**Named Entity Recognition with Bi-LSTM with CRF**

*What if we put CRF on top of the Bi-LSTM model. By adding a CRF layer, the model can handle the dependency between predicted entity names*





# 4 Sequence Model for NER

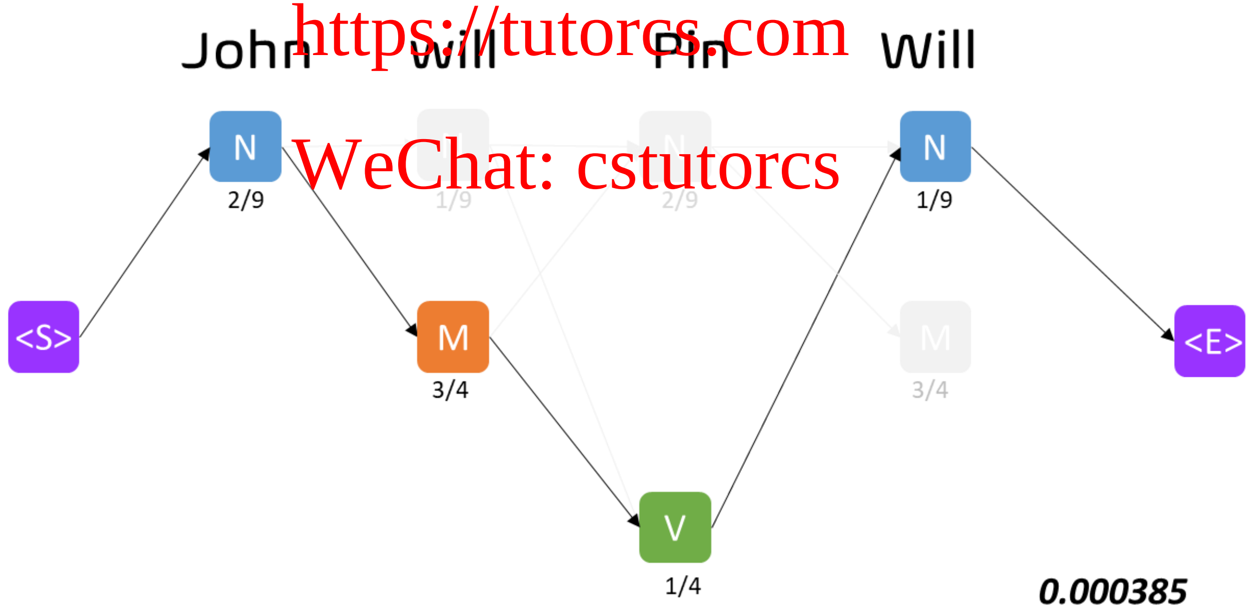
**Remember?**

**POS Tagging: with HMM**

| | N | V | M |
|---|---|---|---|
| Emma | 4/9 | 0 | 0 |
| John | 2/9 | 0 | 0 |
| Will | 1/9 | 0 | 3/4 |
| Pin | 2/9 | 1/4 | 0 |
| Can | 0 | 0 | 1/4 |
| Meet | 0 | 2/4 | 0 |
| Pat | 0 | 1/4 | 0 |

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





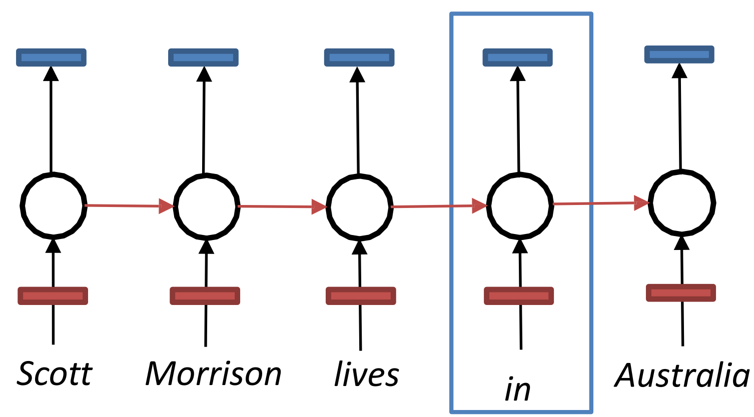

***0.000385***

# 4 Sequence Model for NER

## Greedy Inference

- Greedy inference:
  - We just start at the left, and use our classifier at each position to assign a label
  - The classifier can depend on previous labeling decisions as well as observed data
- Advantages:
  - Fast, no extra memory requirements
  - Very easy to implement
  - With rich features including observations to the right, it may perform quite well
- Disadvantage:
  - Greedy. We make commit errors we cannot recover from



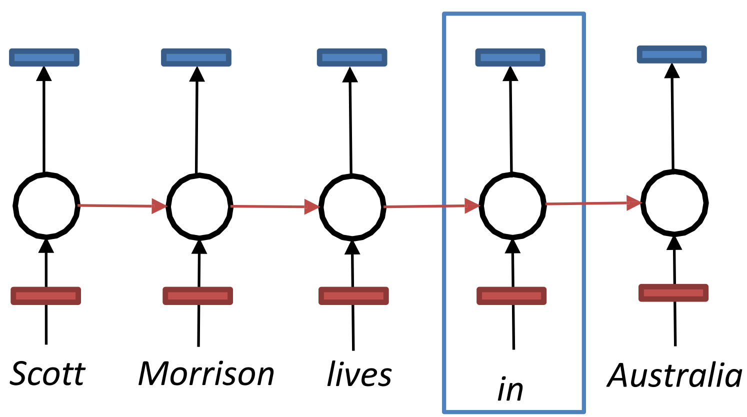


# 4 Sequence Model for NER

## Beam Inference

- Beam inference:
  - At each position keep the top k complete sequences.
  - Extend each sequence in each local way.
  - The extensions compete for the k slots at the next position.
- Advantages:
  - Fast; beam sizes of 3–5 are almost as good as exact inference in many cases.
  - Easy to implement (no dynamic programming required).
- Disadvantage:
  - Inexact: the globally best sequence can fall off the beam.



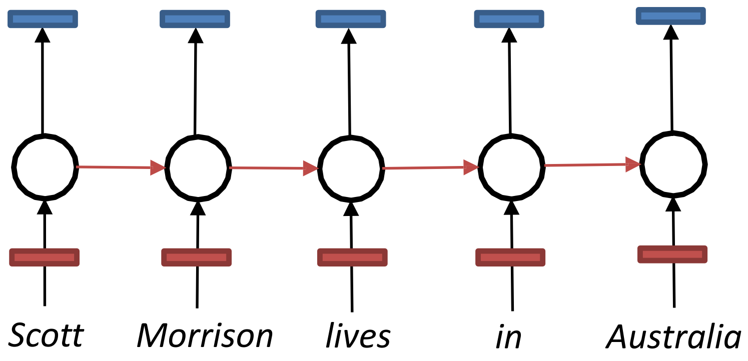


# 4 Sequence Model for NER

## Viterbi Inference

- Viterbi inference:
  - Dynamic programming or memorisation.
  - Requires small window of state influence (e.g., past two states are relevant).
- Advantage:
  - Exact: the global best sequence is returned.
- Disadvantage:
  - Harder to implement long-distance state-state interactions





# 3 Probabilistic Approaches

## Viterbi Algorithm

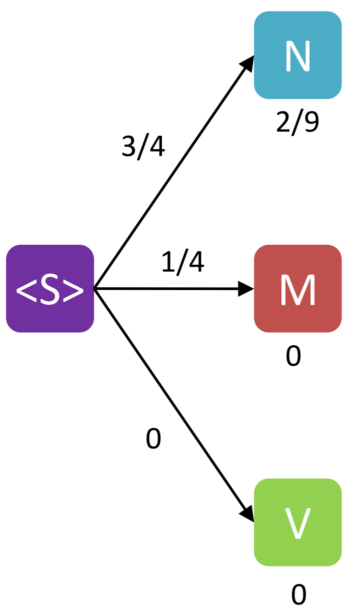
| | N | V | M |
|---|---|---|---|
| Emma | 4/9 | 0 | 0 |
| John | 2/9 | 0 | 0 |
| Will | 1/9 | 0 | 3/4 |
| Pin | 2/9 | 1/4 | 0 |
| Can | 0 | 0 | 1/4 |
| Meet | 0 | 2/4 | 0 |
| Pat | 0 | 1/4 | 0 |

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



John will Pin Will





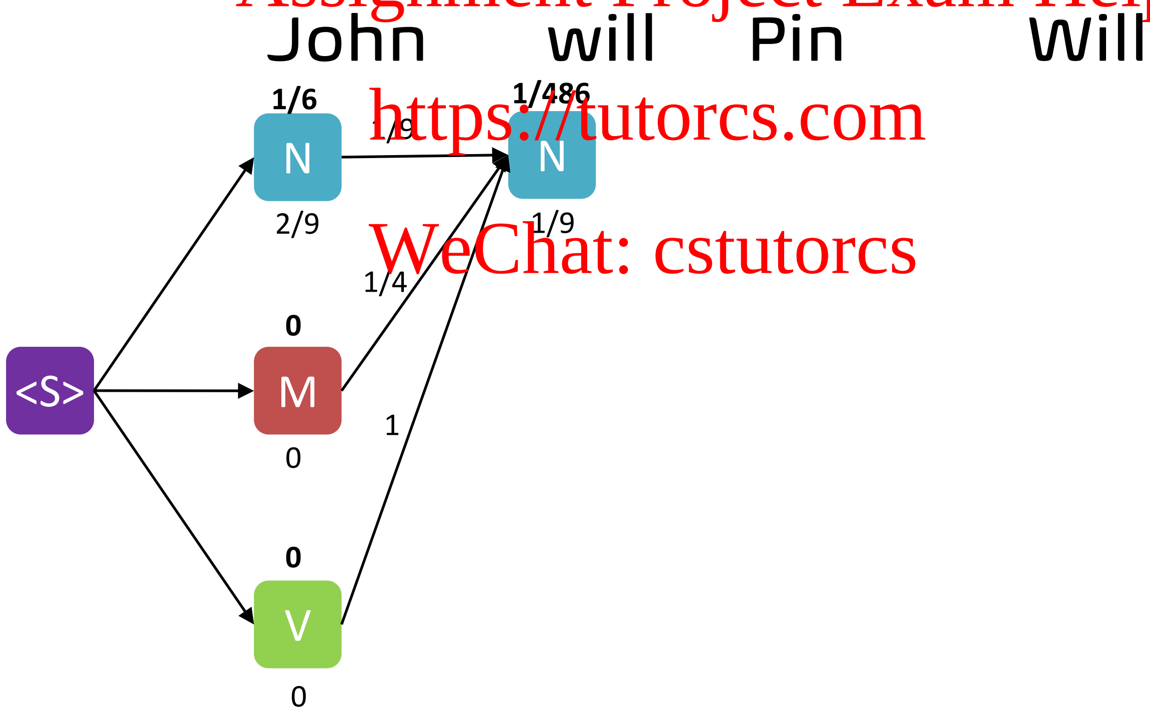
# 3 Probabilistic Approaches

## Viterbi Algorithm

| | N | V | M |
|---|---|---|---|
| Emma | 4/9 | 0 | 0 |
| John | 2/9 | 0 | 0 |
| Will | 1/9 | 0 | 3/4 |
| Pin | 2/9 | 1/4 | 0 |
| Can | 0 | 0 | 1/4 |
| Meet | 0 | 2/4 | 0 |
| Pat | 0 | 1/4 | 0 |

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





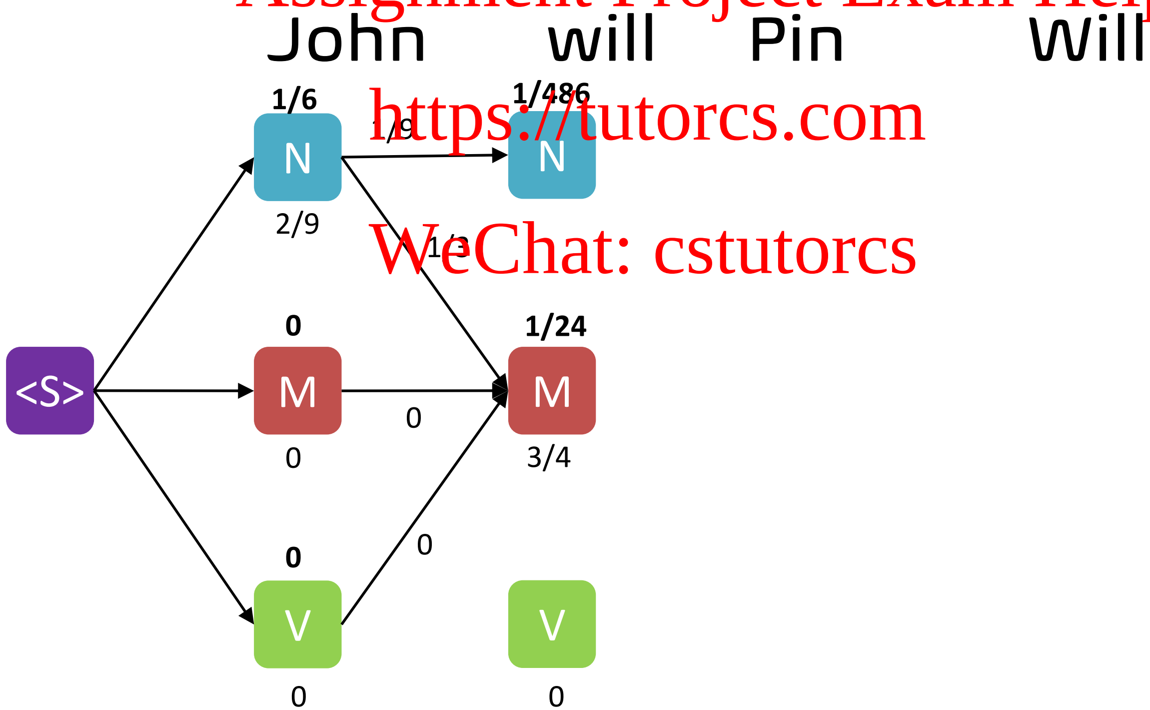
## 3 Probabilistic Approaches

### Viterbi Algorithm

| | N | V | M |
|---|---|---|---|
| Emma | 4/9 | 0 | 0 |
| John | 2/9 | 0 | 0 |
| Will | 1/9 | 0 | 3/4 |
| Pin | 2/9 | 1/4 | 0 |
| Can | 0 | 0 | 1/4 |
| Meet | 0 | 2/4 | 0 |
| Pat | 0 | 1/4 | 0 |

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





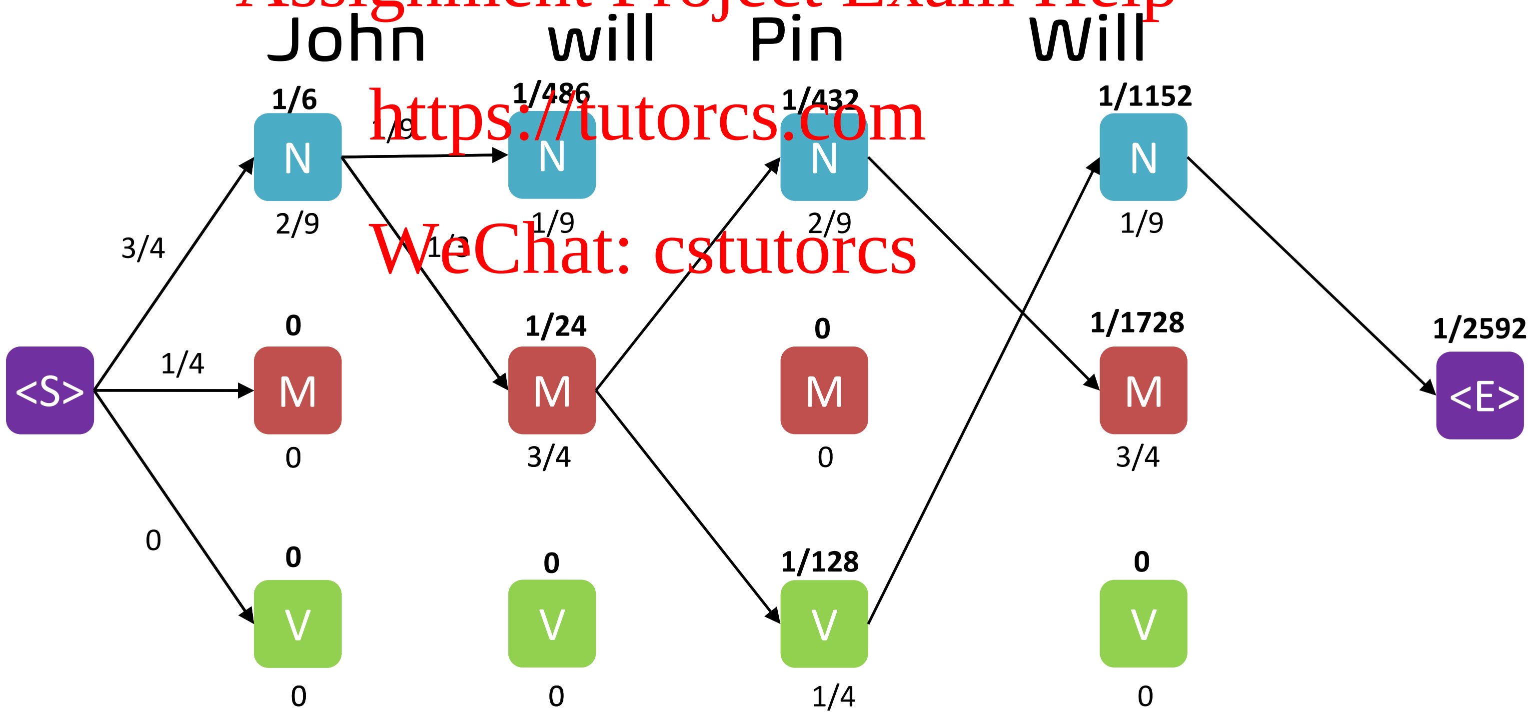
# Probabilistic Approaches

## Viterbi Algorithm

| | N | V | M |
|---|---|---|---|
| Emma | 4/9 | 0 | 0 |
| John | 2/9 | 0 | 0 |
| Will | 1/9 | 0 | 3/4 |
| Pin | 2/9 | 1/4 | 0 |
| Can | 0 | 0 | 1/4 |
| Meet | 0 | 2/4 | 0 |
| Pat | 0 | 1/4 | 0 |

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





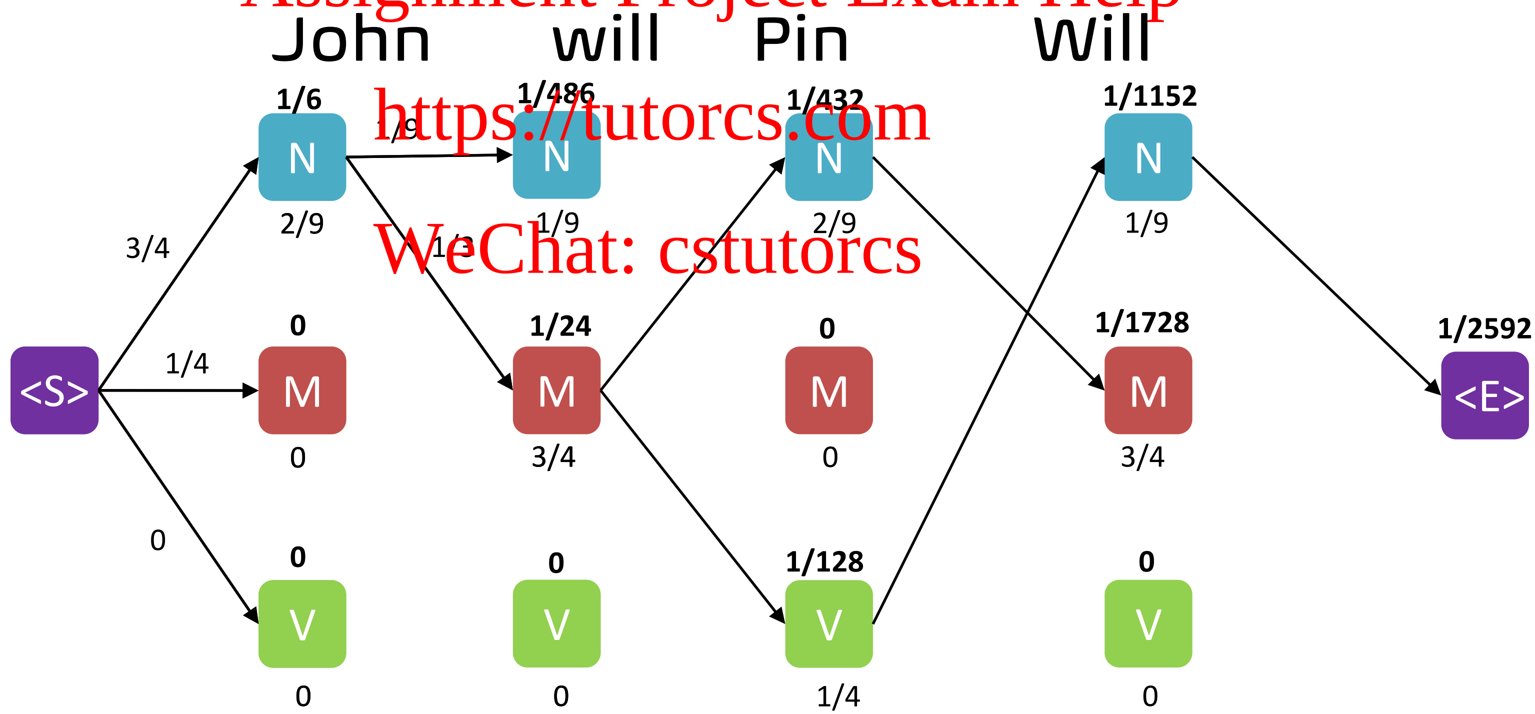
THE UNIVERSITY OF SYDNEY

## Viterbi Algorithm

| | N | V | M |
|---|---|---|---|
| Emma | 4/9 | 0 | 0 |
| John | 2/9 | 0 | 0 |
| Will | 1/9 | 0 | 3/4 |
| Pin | 2/9 | 1/4 | 0 |
| Can | 0 | 0 | 1/4 |
| Meet | 0 | 2/4 | 0 |
| Pat | 0 | 1/4 | 0 |

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



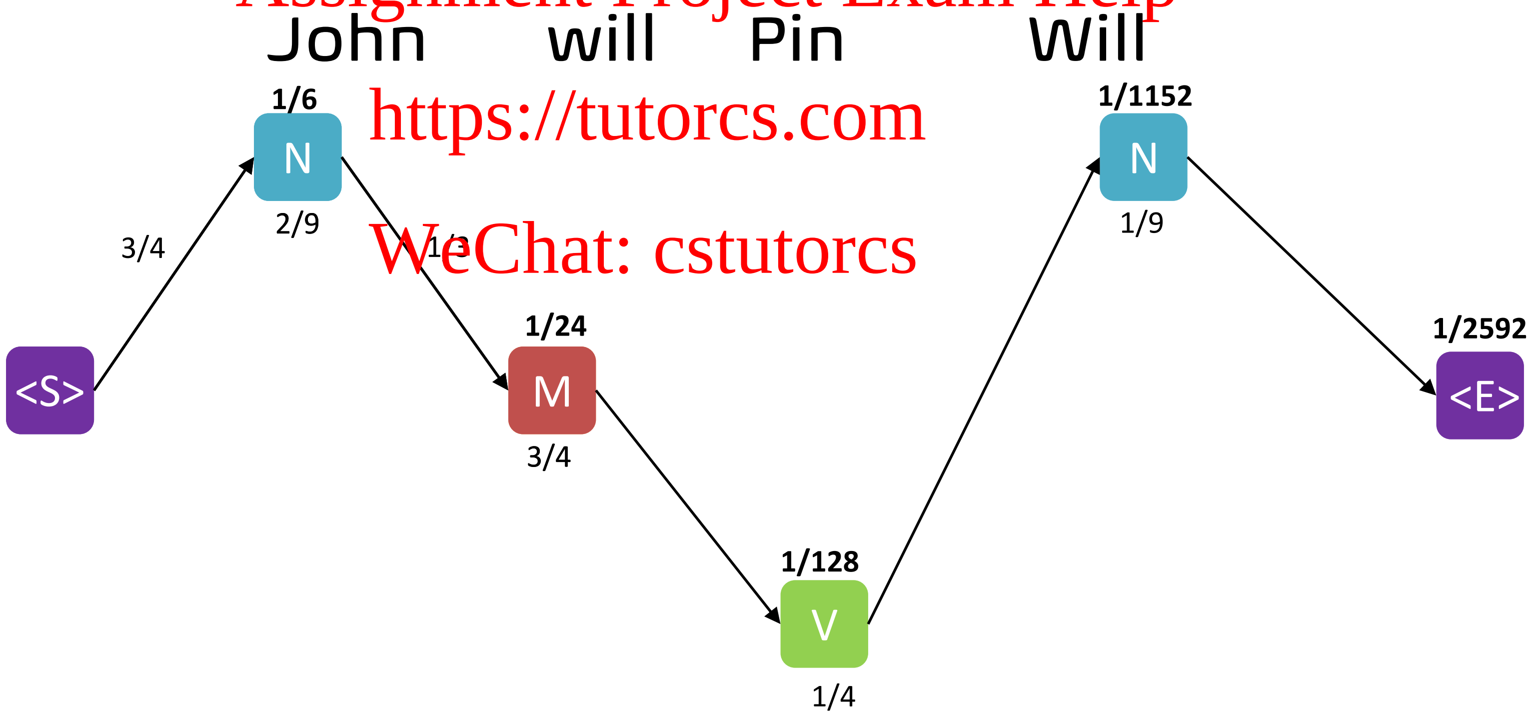


# Viterbi Algorithm

| | N | V | M |
|---|---|---|---|
| Emma | 4/9 | 0 | 0 |
| John | 2/9 | 0 | 0 |
| Will | 1/9 | 0 | 3/4 |
| Pin | 2/9 | 1/4 | 0 |
| Can | 0 | 0 | 1/4 |
| Meet | 0 | 2/4 | 0 |
| Pat | 0 | 1/4 | 0 |

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





## Viterbi Algorithm

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

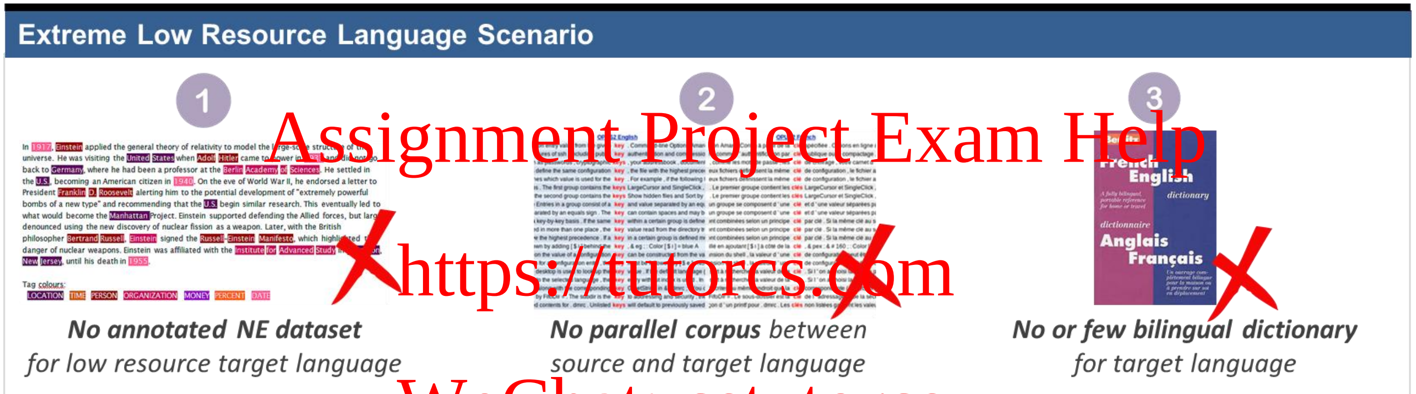


create a path probability matrix $viterbi[N,T]$
**for** each state $s$ **from** 1 **to** $N$ **do** ; initialization step
    $viterbi[s,1] \leftarrow \pi_s * b_s(o_1)$
    $backpointer[s,1] \leftarrow 0$
**for** each time step $t$ **from** 2 **to** $T$ **do** ; recursion step
  **for** each state $s$ **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* ← the path starting at state *bestpathpointer*, that follows backpointer[] to states back in time
**return** *bestpath*, *bestpathprob*

# 4 Sequence Model for NER

**What if there is a language that do not have any annotation?**

*NER in Low Resource Language*






***Current State of the Art model:*** *Han et al. 2019 from Usyd NLP Research Group*

| TASK | DATASET | MODEL | METRIC NAME | METRIC VALUE | GLOBAL RANK | COMPARE |
|---|---|---|---|---|---|---|
| Low Resource Named Entity Recognition | CONLL 2003 Dutch | Low Resource Named Entity Recognition using Contextual Word Representation and Neural Cross-Lingual Knowledge Transfer | F1 score | 75.10 | # 1 | See all |
| Low Resource Named Entity Recognition | CONLL 2003 German | Low Resource Named Entity Recognition using Contextual Word Representation and Neural Cross-Lingual Knowledge Transfer | F1 score | 58.63 | # 1 | See all |
| Low Resource Named Entity Recognition | Conll 2003 Spanish | Low Resource Named Entity Recognition using Contextual Word Representation and Neural Cross-Lingual Knowledge Transfer | F1 score | 75.34 | # 1 | See all |
| Low Resource Named Entity Recognition | Uyghur Unsequestered Set | Low Resource Named Entity Recognition using Contextual Word Representation and Neural Cross-Lingual Knowledge Transfer | F1 score | 42.88 | # 1 | See all |

## NER and Coreference Resolution

NER only produces a list of entities in a text.

- “I voted for **Scott** because he was most aligned with my values”



***Then, How to trace it?***



***Coreference Resolution*** is the task of finding all expressions that refer to the same entity in a text



- “**I** voted for **Scott** because **he** was most aligned with **my** values”
  - **Scott ← he**
  - **I ← my**

5

# Coreference Resolution

## What is Coreference Resolution?

Finding all mentions that refer to the same entity



Donald Trump said he considered nominating Ivanka Trump to be president of the World Bank because "she is very good with numbers,"





# 5 Coreference Resolution

### What is Coreference Resolution?

Finding all mentions that refer to the same entity



**Donald Trump** said **he** considered nominating Ivanka Trump to be president of the World Bank because "she is very good with numbers,"



# 5 Coreference Resolution

**What is Coreference Resolution?**

Finding all mentions that refer to the same entity



Donald said he considered nominating **Ivanka Trump** to be **president of the World Bank** because "**she** is very good with numbers,"



# 5 Coreference Resolution

**How to conduct Coreference Resolution?**

**1. Detect the mentions**

**** Mention: span of text referring to same entity***



- Pronouns

e.g. I, your, it, she, him, etc.

- Named entities

e.g. people, places, organisation etc.

- Noun phrases

e.g. a cat, a big fat dog, etc.

# 5 Coreference Resolution

**The difficulty in coreference resolution**

1. Detect the mentions

**** Mention: span of text referring to same entity***



**Tricky mentions…**

- **It** was very interesting
- **No staff**
- **The best university in Australia**

***How to handle this tricky mentions?*** ***Classifiers!***

# Coreference Resolution

**How to conduct Coreference Resolution?**

***1. Detect the mentions***

**Donald Trump** said **he** considered nominating **Ivanka Trump** to be **president of the World Bank** because “**she** is very good with numbers,”



***2. Cluster the mentions***

**Donald Trump** said **he** considered nominating **Ivanka Trump** to be **president of the World Bank** because “**she** is very good with numbers,”

# 5 Coreference Resolution

**How to cluster the mentions and find the coreference**

**Coreference**

It occurs when two or more expressions in a text refer to the same person or thing.



- **"Donald Trump** is a president of the United States. **Trump** was born and raised in the New York City borough of Queens"



**Anaphora**



The use of a word referring back to a word used earlier in a text or conversation. Mostly noun phrases

- a word (anaphor) refers to another word (antecedent)
- "**Donald Trump** is a president of the United States. Before entering politics, **he** was a businessman and television personality"

*antecedent* *anaphor*

# 5 Coreference Resolution

## Coreference vs Anaphora

*Coreference*

Donald Trump



Trump





*Anaphora*

Donald Trump

he

# Coreference Resolution

**Not all anaphoric relations are coreferential**

**1. Not all noun phrases have reference**

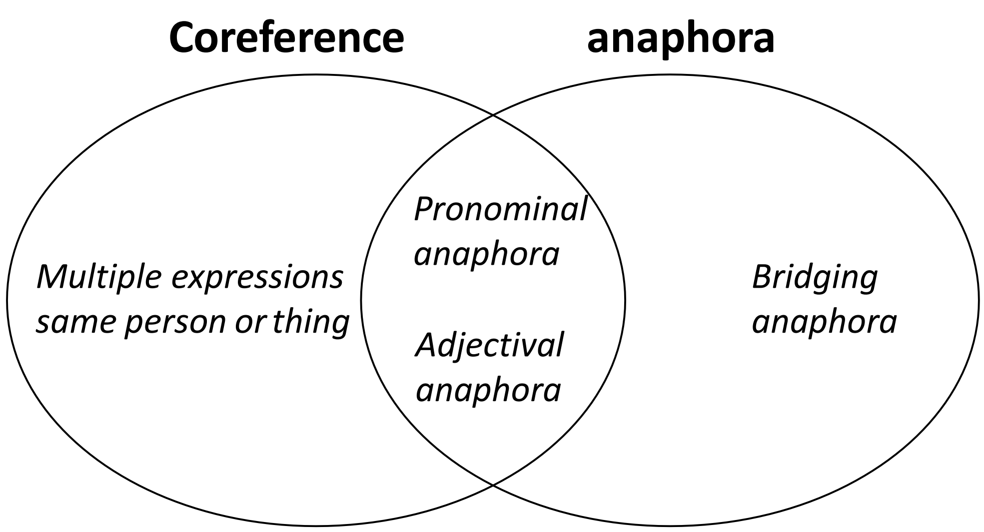
- Every student like his speech
- No student like his speech



**2. Not all anaphoric relations are co-referential** (bridging anaphora)

- I attended **the meeting** yesterday. **The presentation** was awesome!






**cataphora**

I almost stepped on **it**.
It was a big **snake**…

# Coreference Model

### How to Cluster Mentions?

After detecting this all mentions in a text, we need to cluster them!



***Ivanka***

***Donald***

***he***

***her***

***she***

**Ivanka** was happy that **Donald** said **he** considered nominating **her** because **she** is very good with numbers

# Coreference Model



**How to Cluster Mentions?**

After detecting this all mentions in a text, we need to cluster them!



*Ivanka*

*Donald*

*he*

*her*

*she*

**Ivanka** was happy that **Donald** said **he** considered nominating **her** because **she** is very good with numbers

*Gold cluster 1*

*Gold cluster 2*

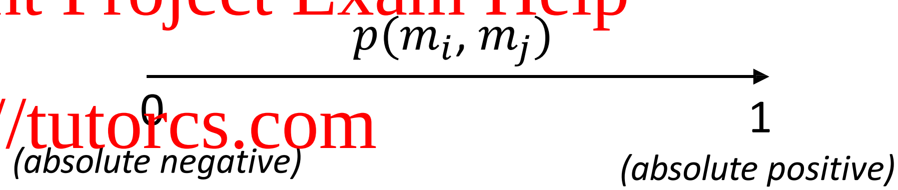
# Coreference Model



## How to Cluster Mentions?

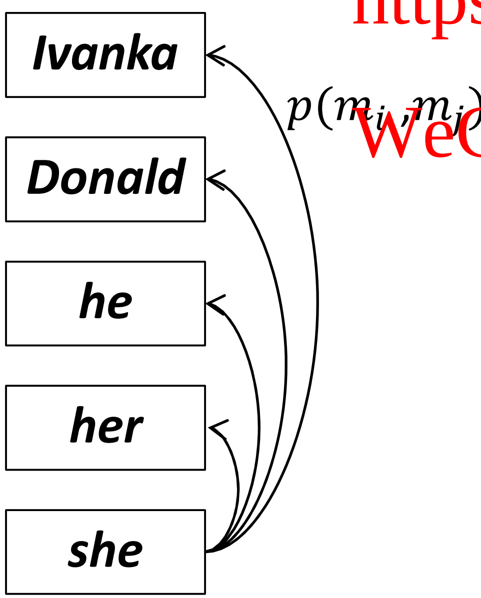
- Train a binary classifier that assigns every pair of mentions a probability of being coreferent: $p(m_i, m_j)$










**Ivanka** was happy that **Donald** said **he** considered nominating **her** because **she** is very good with numbers

# 6 Coreference Model

THE UNIVERSITY OF SYDNEY

## Mention Pair Training

- N mentions in a document
- $y_{ij} = 1$ if mentions $m_i$ and $m_j$ are coreferent, -1 if otherwise
- Just train with regular cross-entropy loss (looks a bit different because it is binary classification)



$$J = -\sum_{i=2}^{N} \sum_{j=1}^{i} y_{ij} \underline{\log p(m_j, m_i)}$$

Coreferent mentions pairs should get high probability, others should get low probability

Iterate through mentions

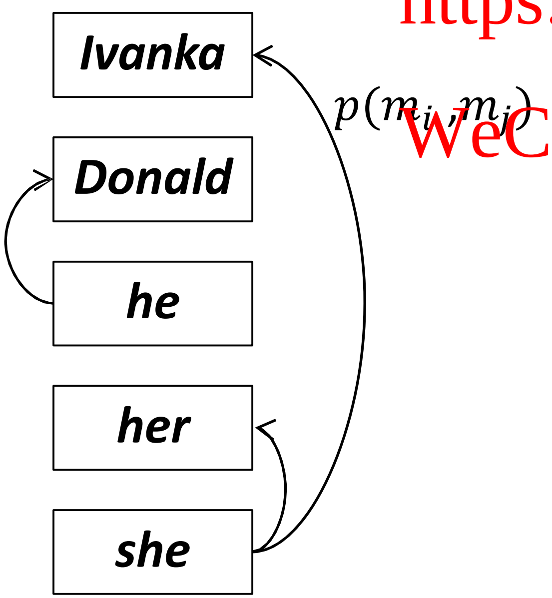
Iterate through candidate antecedents (previously occurring mentions)

# 6 Coreference Model

## Mention Pair Testing

- Coreference resolution is a clustering task, but we are only scoring pairs of mentions… what to do?
- Pick some threshold (e.g., 0.5) and add coreference links between mention pairs where p(mi, mj) is above the threshold





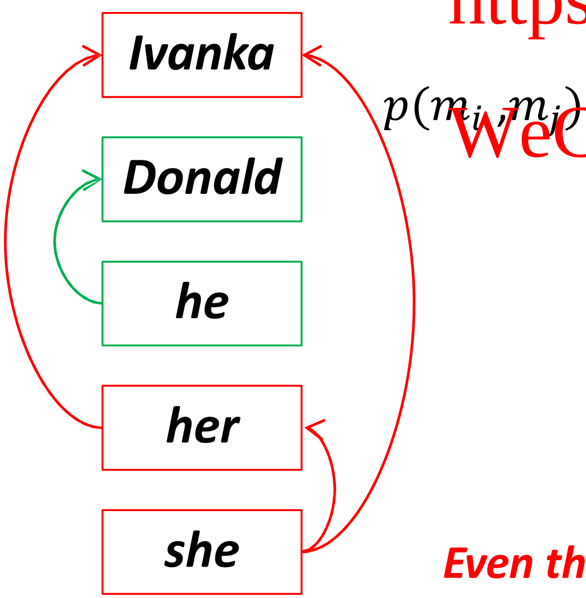

**Ivanka** was happy that **Donald** said **he** considered nominating **her** because **she** is very good with numbers

# 6 Coreference Model

## Mention Pair Testing

- Pick some threshold (e.g., 0.5) and add coreference links between mention pairs where $p(m_i, m_j)$ is above the threshold
- Take the transitive closure to get the clustering






**Ivanka** was happy that **Donald** said **he** considered nominating **her** because **she** is very good with numbers

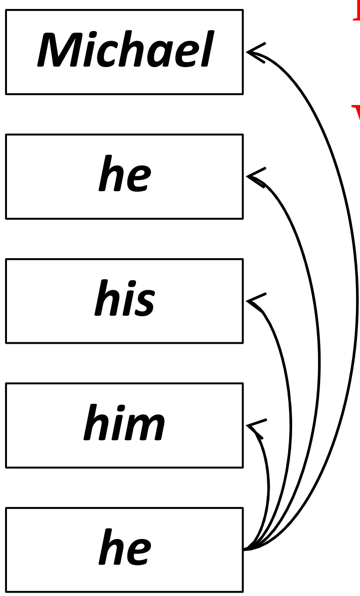
***Even though the model did not predict this coreference link, Ivanka and her are coreferent due to transitivity***

# 6 Coreference Model

## Mention Pair Testing: Issue

- Assume that we have a **long document** with the following mentions
- Michael… he … his … him … <several paragraphs>
- … won the game because he …






Many mentions only have one clear antecedent but we are asking the model to predict all of them

**Alternative solution:** instead train the model to predict only one antecedent for each mention

***Mention Ranking***

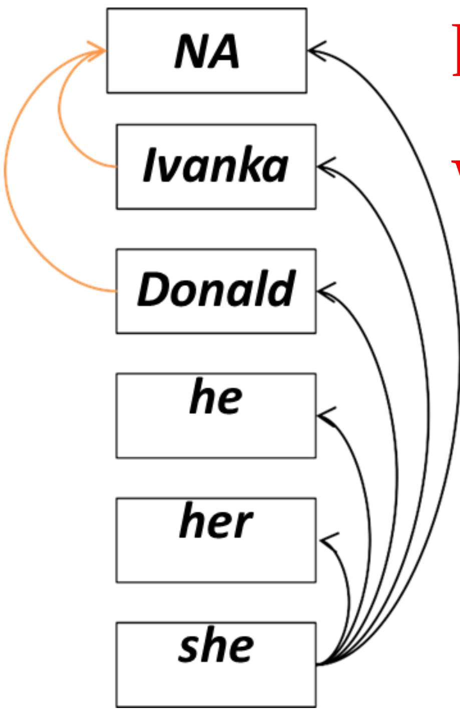
# Coreference Model



## Mention Ranking

- Assign each mention its highest scoring candidate antecedent according to the model
- Dummy NA mention allows model to decline linking the current mention to anything ("singleton" or "first" mention)





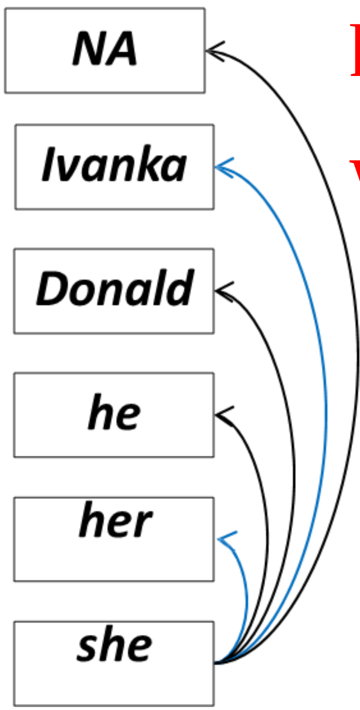

What can be the best antecedent for **she**?

# 6 Coreference Model

## Mention Ranking

- Assign each mention its highest scoring candidate antecedent according to the model
- Dummy NA mention allows model to decline linking the current mention to anything ("singleton" or "first" mention)






What can be the best antecedent for she?

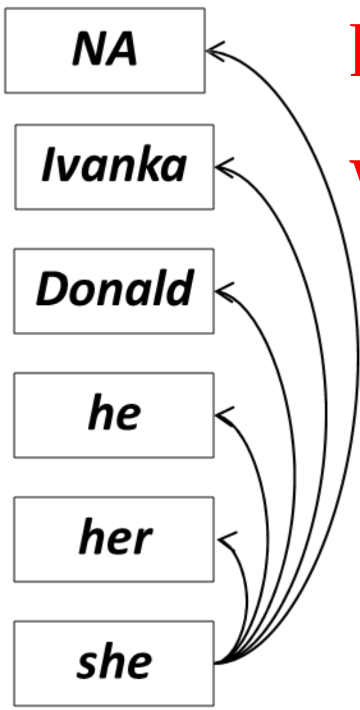
Positive examples: model has to assign a high probability to either one (but not necessarily both)

# 6 Coreference Model

## Mention Ranking

- Assign each mention its highest scoring candidate antecedent according to the model
- Dummy NA mention allows model to decline linking the current mention to anything ("singleton" or "first" mention)






What can be the best antecedent for she?

Apply a **softmax** over the scores for candidate antecedents so probabilities sum to 1

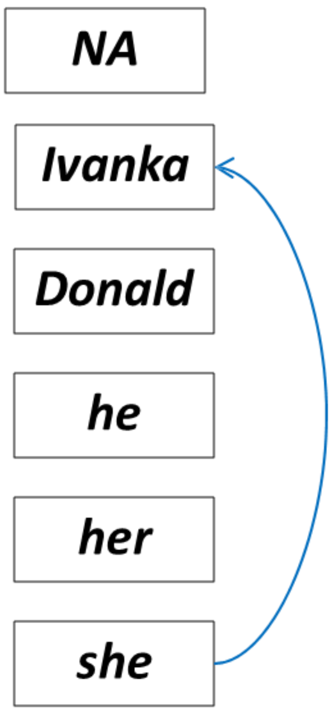
- p(NA, she) = 0.1
- p(Ivanka, she) = 0.5
- p(Donald, she) = 0.1
- p(he, she) = 0.1
- p(her, she) = 0.2

# Coreference Model

## Mention Ranking

- Assign each mention its highest scoring candidate antecedent according to the model
- Dummy NA mention allows model to decline linking the current mention to anything ("singleton" or "first" mention)






What can be the best antecedent for she?

Apply a **softmax** over the scores for candidate antecedents so probabilities sum to 1

- p(NA, she) = 0.1
- **p(Ivanka, she) = 0.5**
- p(Donald, she) = 0.1
- p(he, she) = 0.1
- p(her, she) = 0.2

*only add highest scoring coreference link*

# Coreference Model

## Coreference Models: Training

- The current mention $m_j$ should be linked to any one of the candidate antecedents it's coreferent with.
- Mathematically, maximize this probability:



$$\sum_{j=1}^{i-1} \mathbb{1}(y_{ij} = 1)\, p(m_i, m_j)$$

Iterate through candidate antecedents (previously occurring mentions)

For ones that are coreferent to $m_j$...

...we want the model to assign a high probability

# Coreference Model



## Coreference Models: Training

- The current mention $m_j$ should be linked to any one of the candidate antecedents it's coreferent with.
- Mathematically, maximize this probability:



$$\sum_{j=1}^{i-1} \mathbb{1}(y_{ij} = 1)\, p(m_i, m_j)$$

Iterate through candidate antecedents (previously occurring mentions)

For ones that are coreferent to $m_j$…

…we want the model to assign a high probability

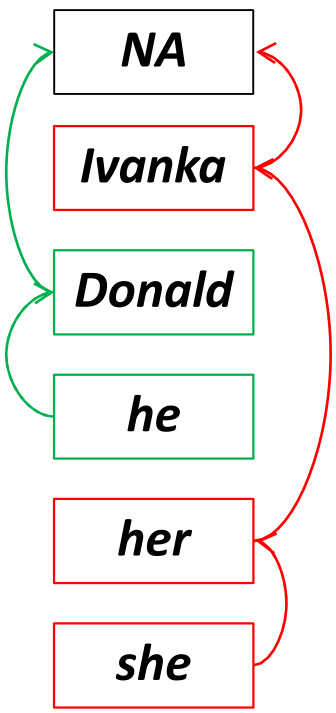
*The model could produce 0.9 probability for one of the correct antecedents and low probability for everything else*

# Coreference Model

## Mention Ranking Models: Test Time

- Similar to mention-pair model except each mention is assigned only one antecedent








How do we compute the probabilities?



- Non-neural statistical classifier
- Simple neural network
- More advanced model using LSTMs, attention

# 6 Coreference Model

**How do we compute the probabilities?**
**End to End Model (Lee at al., 2017)**

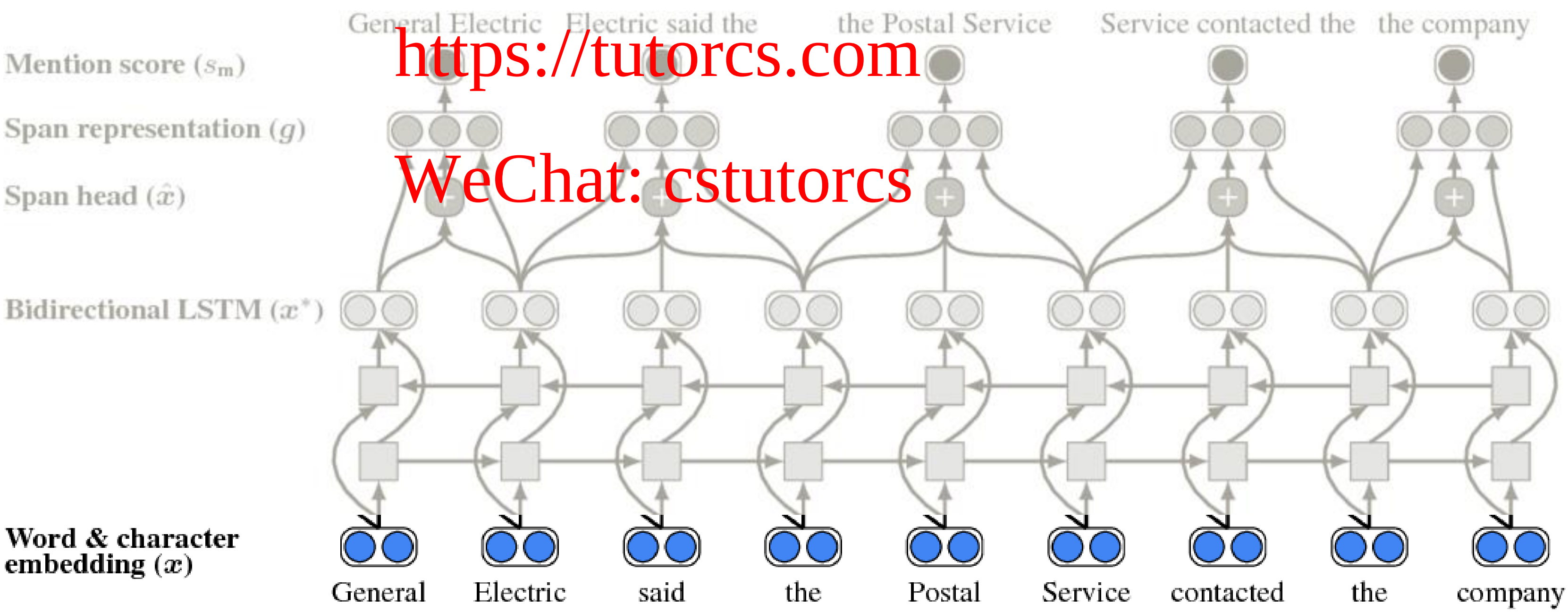
- Current state-of-the-art model for coreference resolution (before 2019)
- Mention ranking model
- Improvements over simple feed-forward NN
  - Use an LSTM
  - Use attention (will learn about this in Lecture 10)
  - Do mention detection and coreference end-to-end
    - No mention detection step

# Coreference Model

**End to End Model (Lee at al., 2017)**

- First embed the words in the document using a word embedding matrix and a character-level embedding



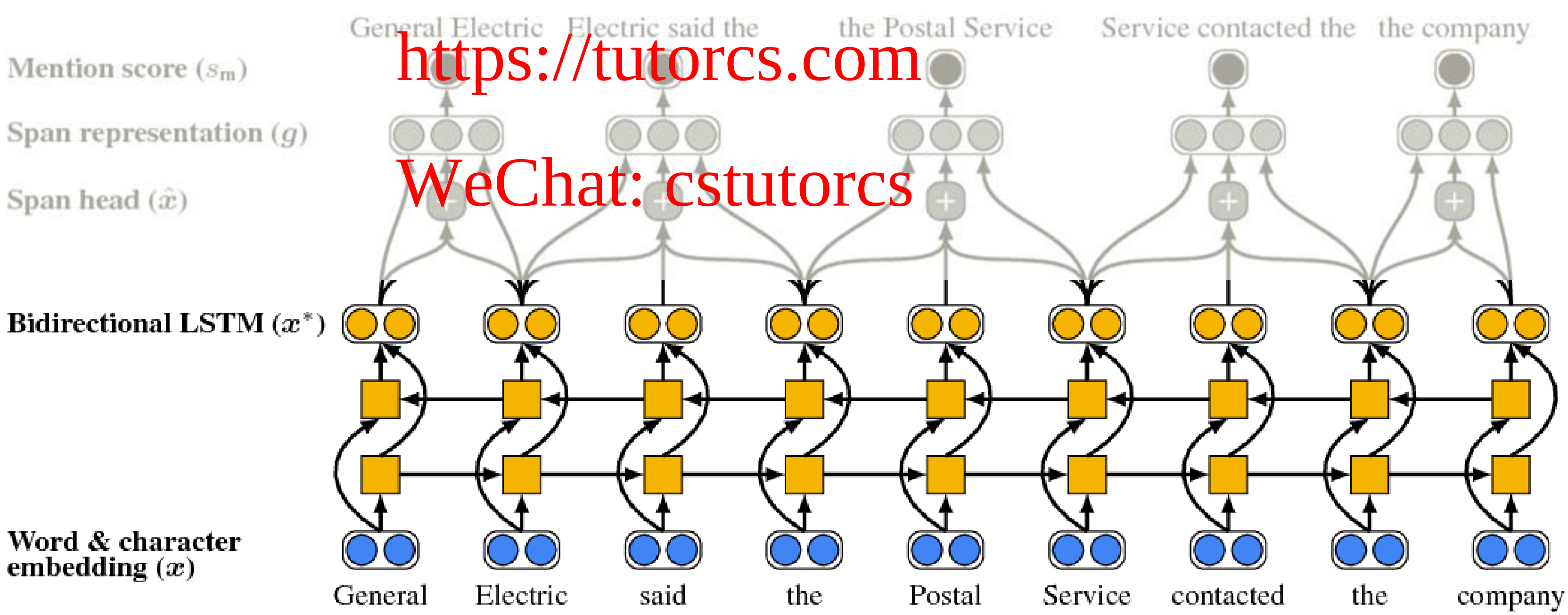


# 6 Coreference Model

**End to End Model (Lee at al., 2017)**

- Then run a bidirectional LSTM over the document



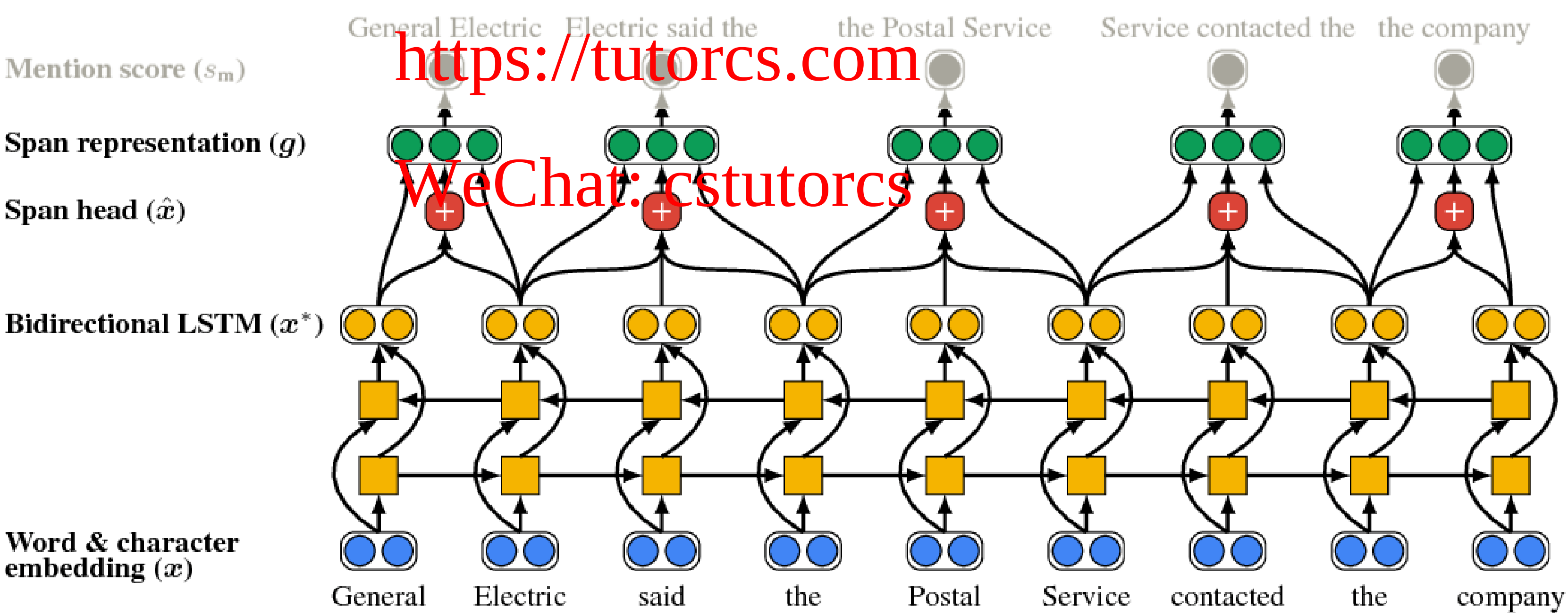


6 Coreference Model

**End to End Model (Lee at al., 2017)**

- *Next, represent each span of text i going from START(i) to END(i) as a vector*



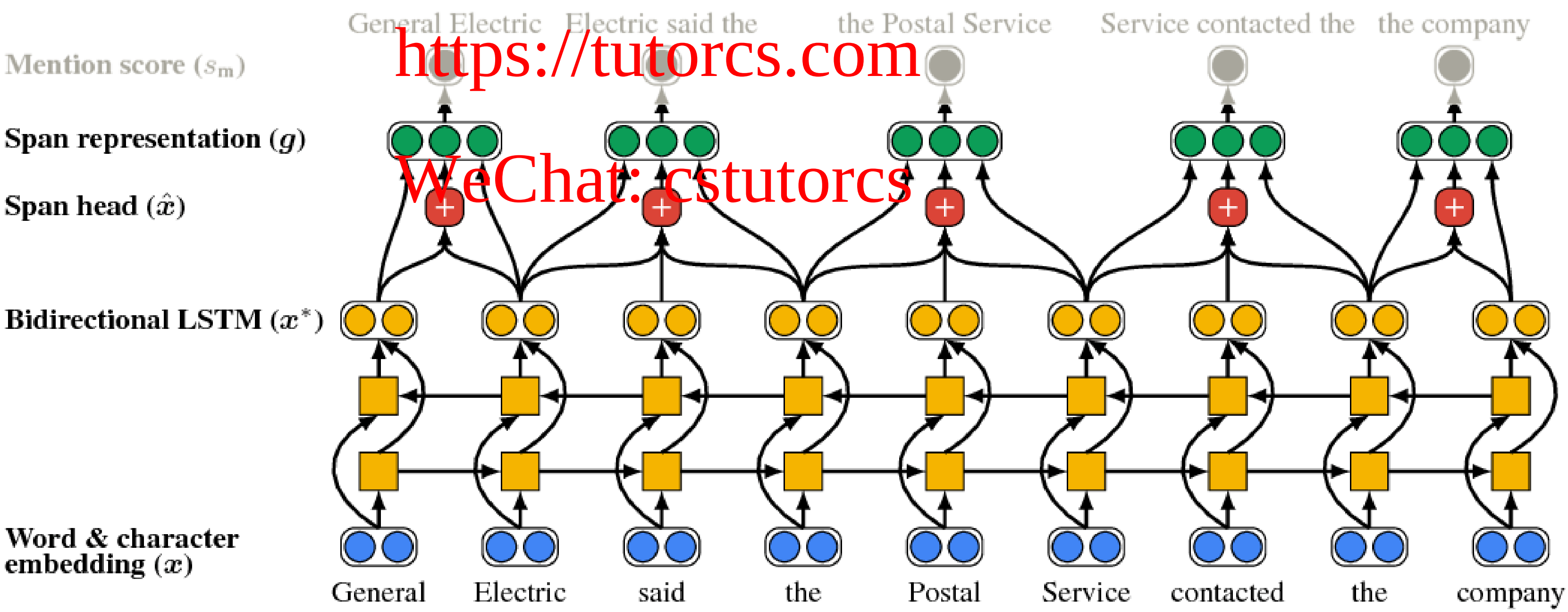


# Coreference Model

**End to End Model (Lee at al., 2017)**

- *Next, represent each span of text i going from START(i) to END(i) as a vector*





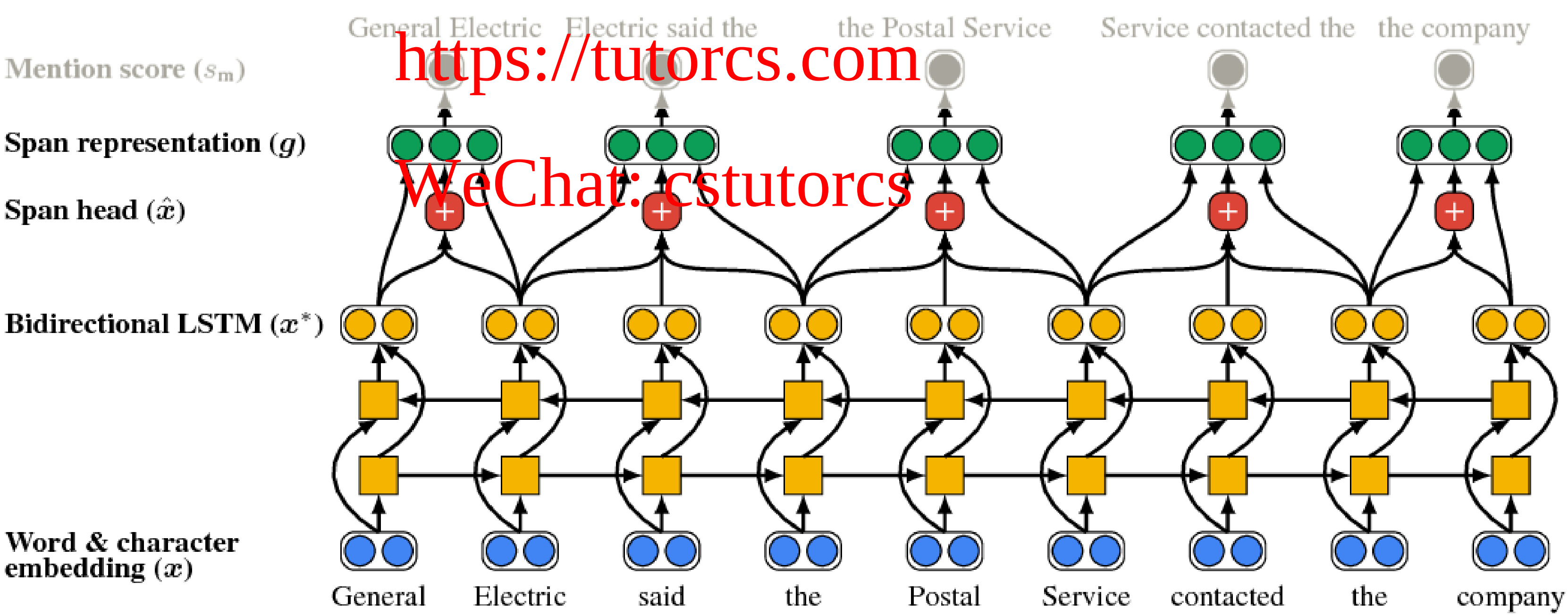

*General, General Electric, General Electric said, … Electric, Electrid said,… will all get its own vector representation*

# 6 Coreference Model

**End to End Model (Lee at al., 2017)**

- *Next, represent each span of text i going from START(i) to END(i) as a vector*







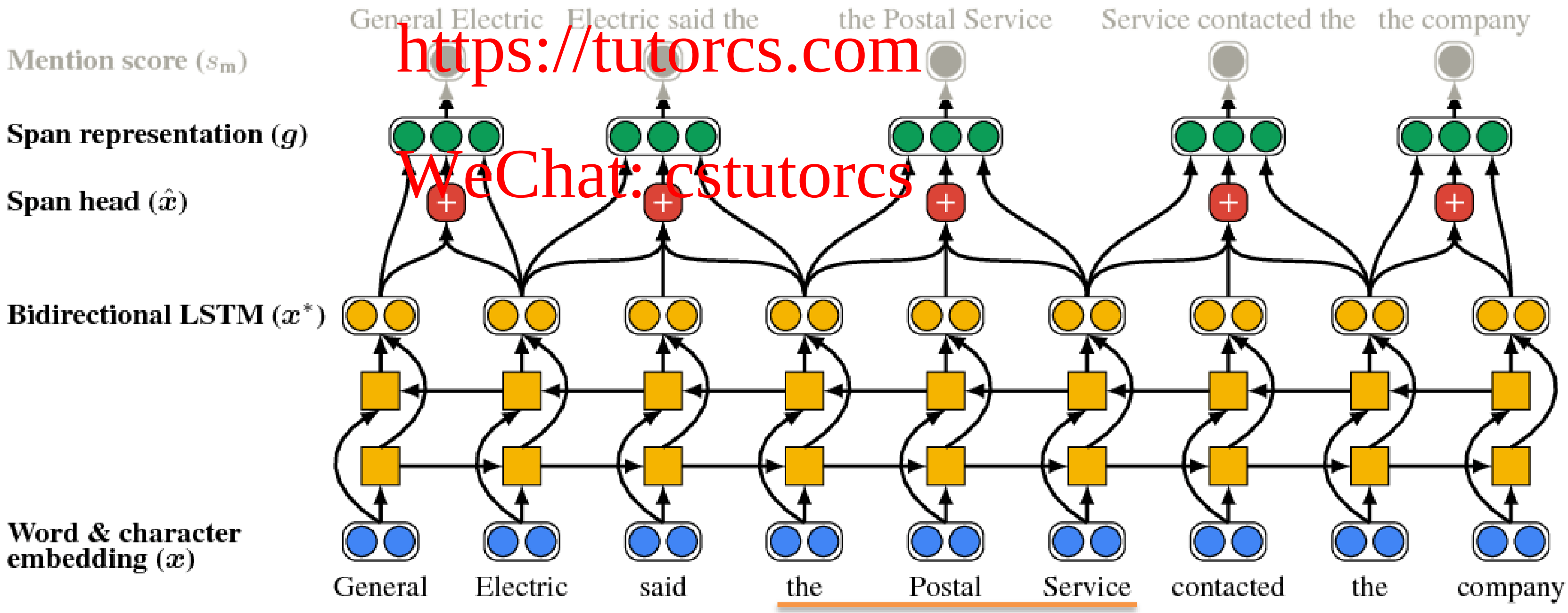

*Span Representation:* $\boldsymbol{g}_i = [\boldsymbol{x}^*_{\mathrm{START}(i)}, \boldsymbol{x}^*_{\mathrm{END}(i)}, \hat{\boldsymbol{x}}_i, \phi(i)]$

# 6 Coreference Model

## End to End Model (Lee at al., 2017)

- *Next, represent each span of text i going from START(i) to END(i) as a vector. For example, for "the postal service"*





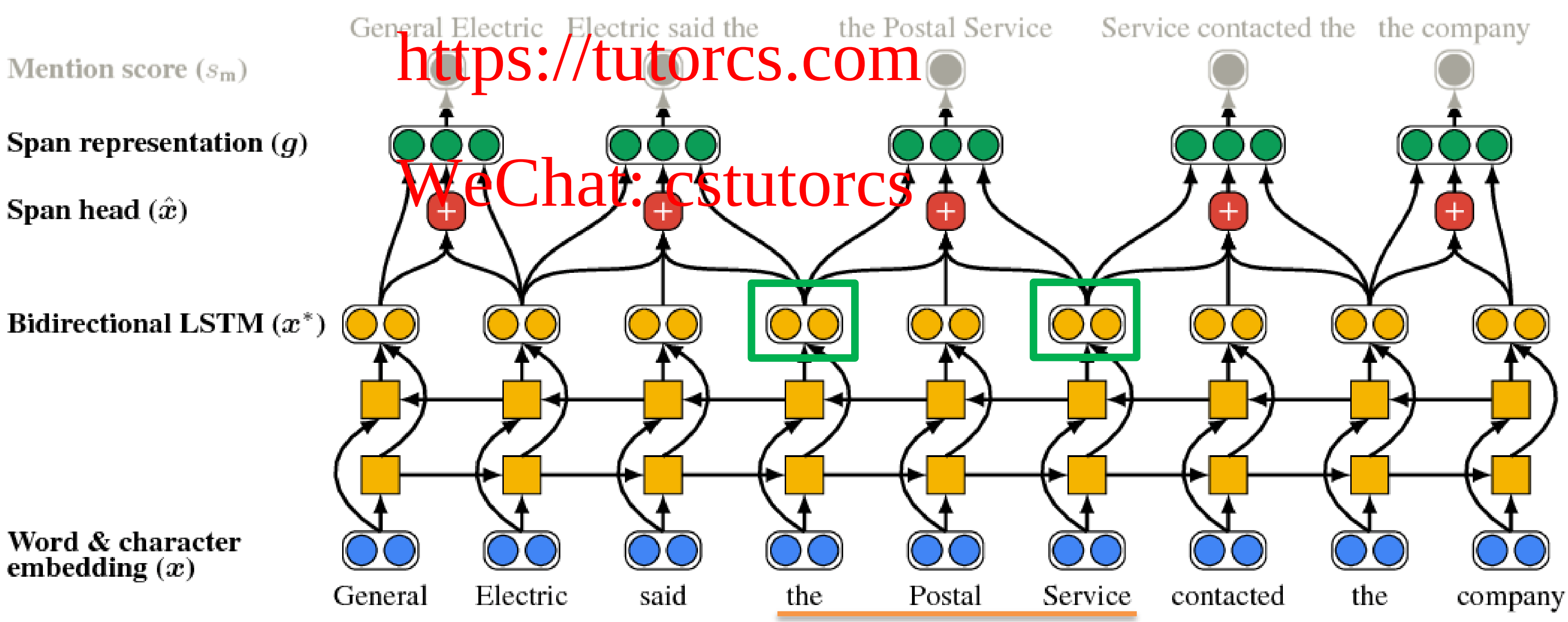

*Span Representation:* $\boldsymbol{g}_i = [\boldsymbol{x}^*_{\text{START}(i)}, \boldsymbol{x}^*_{\text{END}(i)}, \hat{\boldsymbol{x}}_i, \phi(i)]$

# 6 Coreference Model

### End to End Model (Lee at al., 2017)

- *Next, represent each span of text i going from START(i) to END(i) as a vector. For example, for "the postal service"*








*Span Representation:* $\boldsymbol{g}_i = [\underbrace{\boldsymbol{x}^*_{\mathrm{START}(i)}}, \underbrace{\boldsymbol{x}^*_{\mathrm{END}(i)}}, \hat{\boldsymbol{x}}_i, \phi(i)]$

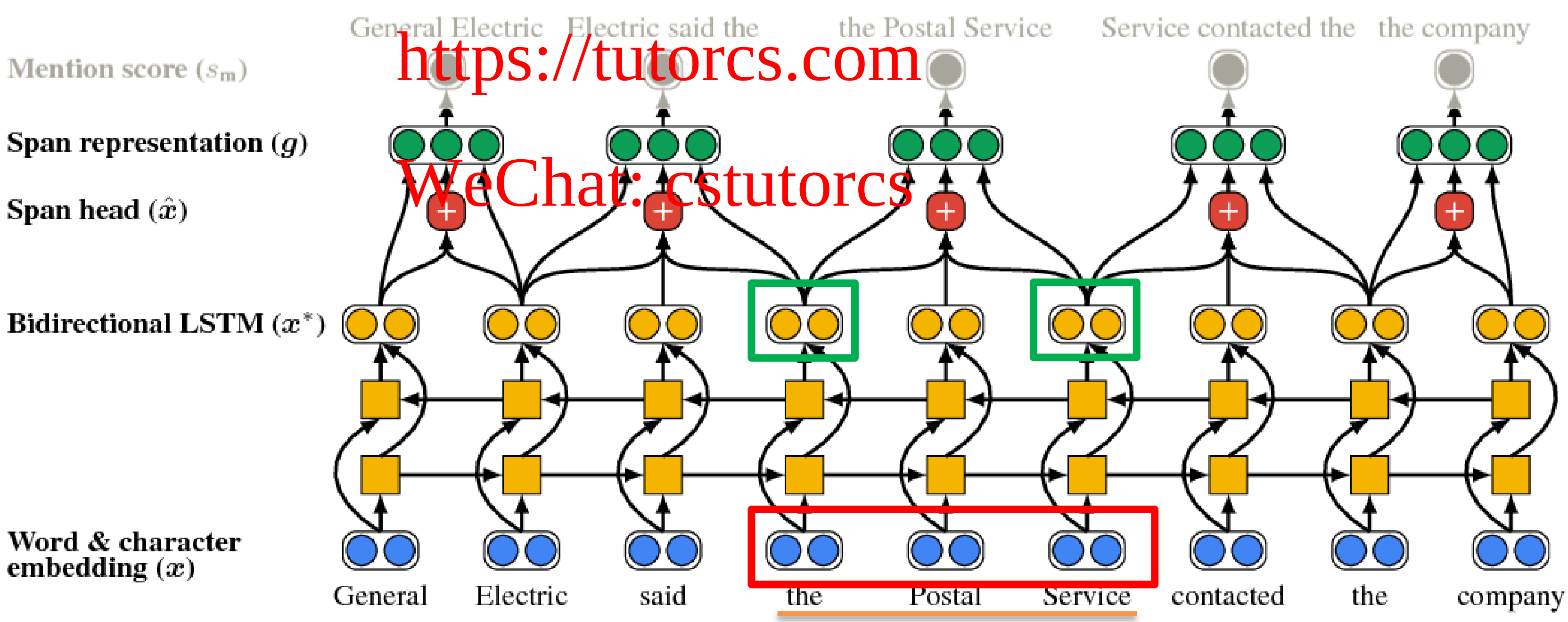
***Bi-LSTM hidden states for span's start and end***

# 6 Coreference Model

## End to End Model (Lee at al., 2017)

- *Next, represent each span of text i going from START(i) to END(i) as a vector. For example, for “the postal service”*








*Span Representation:* $\boldsymbol{g}_i = [\underline{\boldsymbol{x}^*_{\mathrm{START}(i)}}, \underline{\boldsymbol{x}^*_{\mathrm{END}(i)}}, \underline{\hat{\boldsymbol{x}}_i}, \phi(i)]$

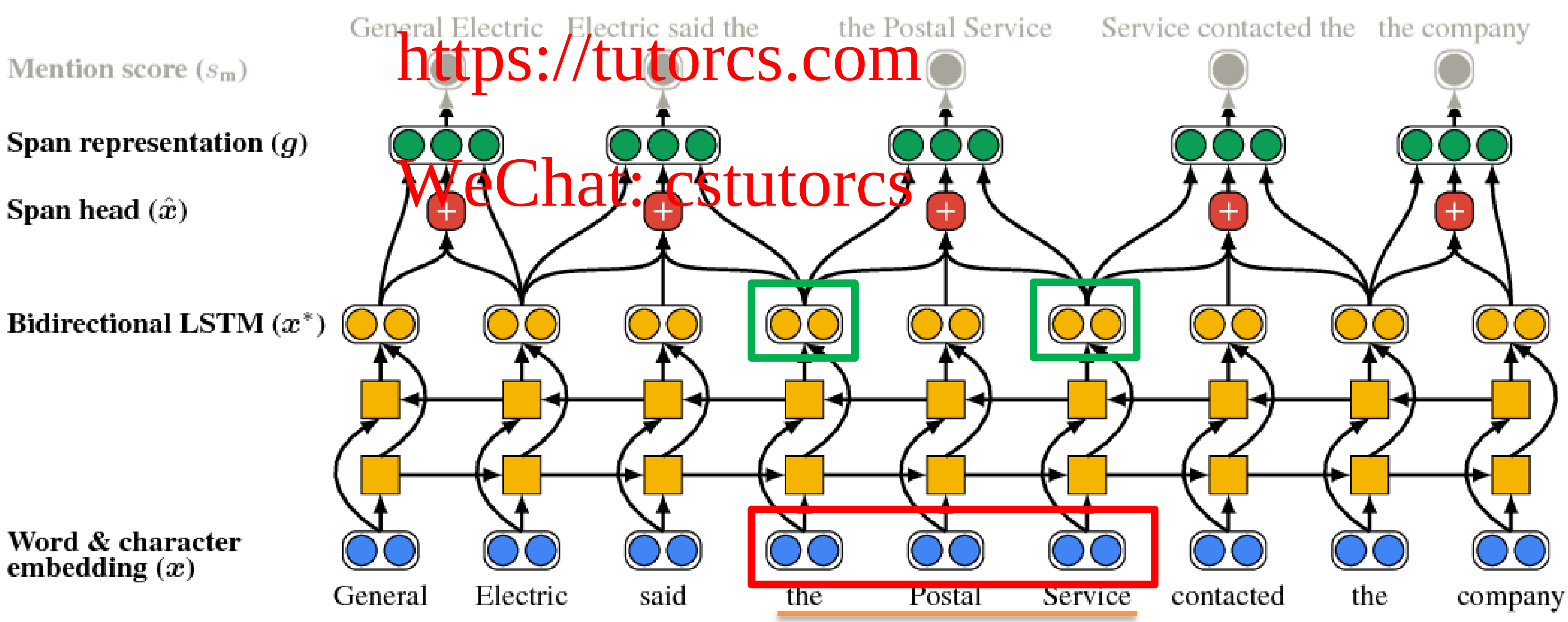
***Attention-based representation (Lecture 10)***

# 6 Coreference Model

## End to End Model (Lee at al., 2017)

- *Next, represent each span of text i going from START(i) to END(i) as a vector. For example, for “the postal service”*








*Span Representation:* $\boldsymbol{g}_i = [\boldsymbol{x}^*_{\text{START}(i)}, \boldsymbol{x}^*_{\text{END}(i)}, \hat{\boldsymbol{x}}_i, \phi(i)]$

***Additional features***

# Coreference Model

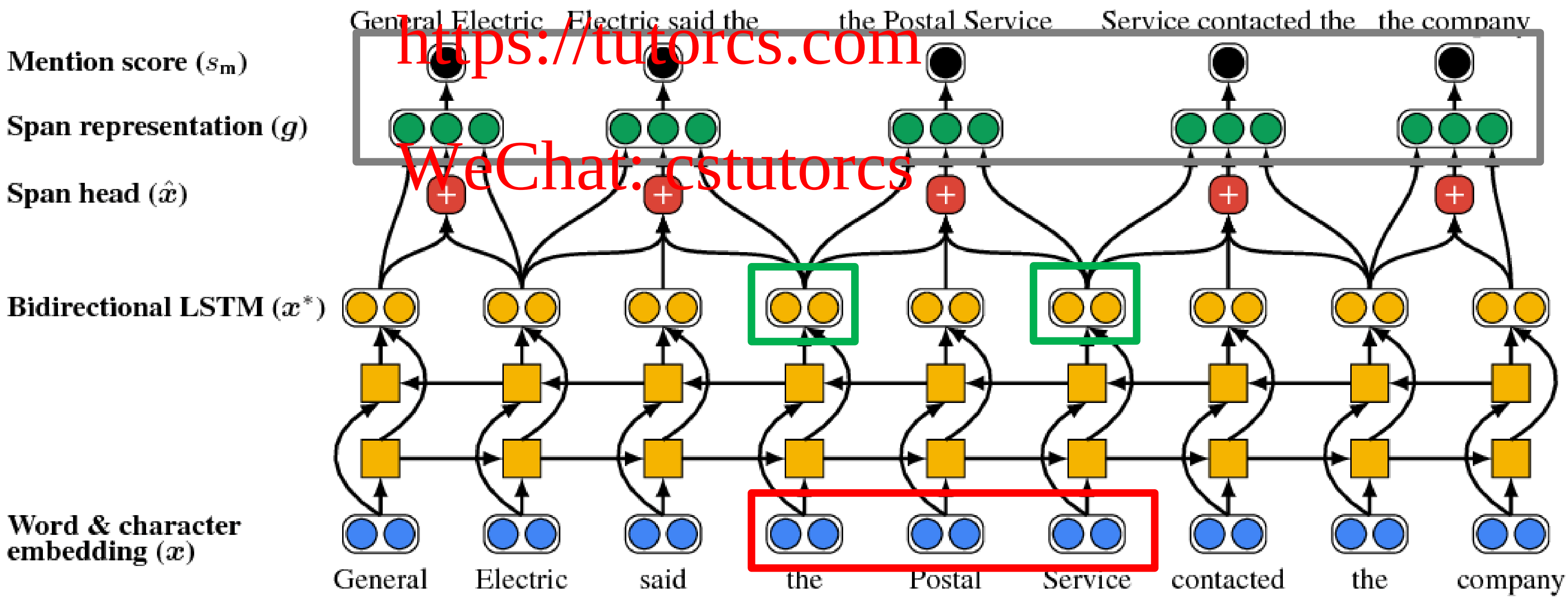
## End to End Model (Lee at al., 2017)

- *Next, represent each span of text i going from START(i) to END(i) as a vector.*

$$s(i, j) = s_{\mathrm{m}}(i) + s_{\mathrm{m}}(j) + s_{\mathrm{a}}(i, j)$$

***Are spans i and j conference mentions?*** *Is i a mention?* *Is j a mention?* ***Do they look coreferent?***





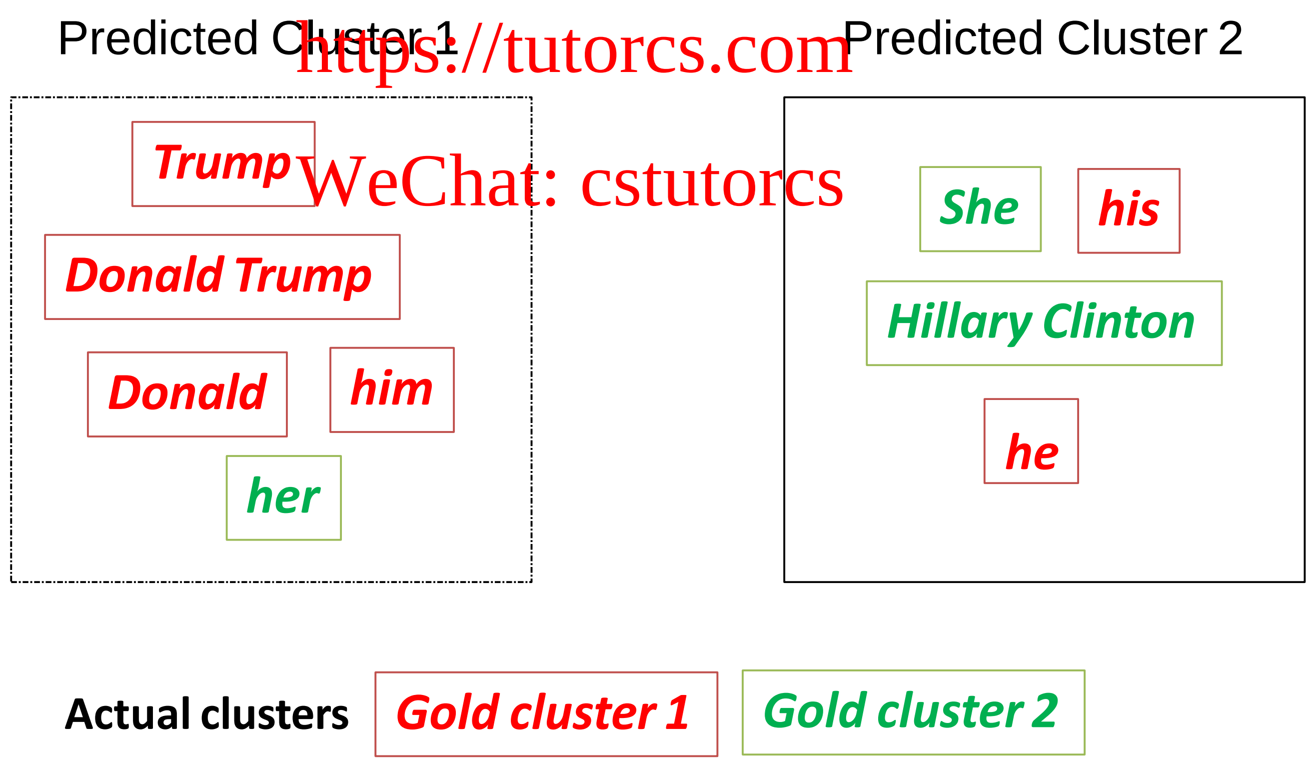
# 7 Coreference Evaluation



**How to evaluate coreference?**

There are different types of metrics available for evaluating coreference, such as B-CUBED, MUC, CEAF, LEA, BLANC, or Often report the average over a few different metrics





# 7 Coreference Evaluation

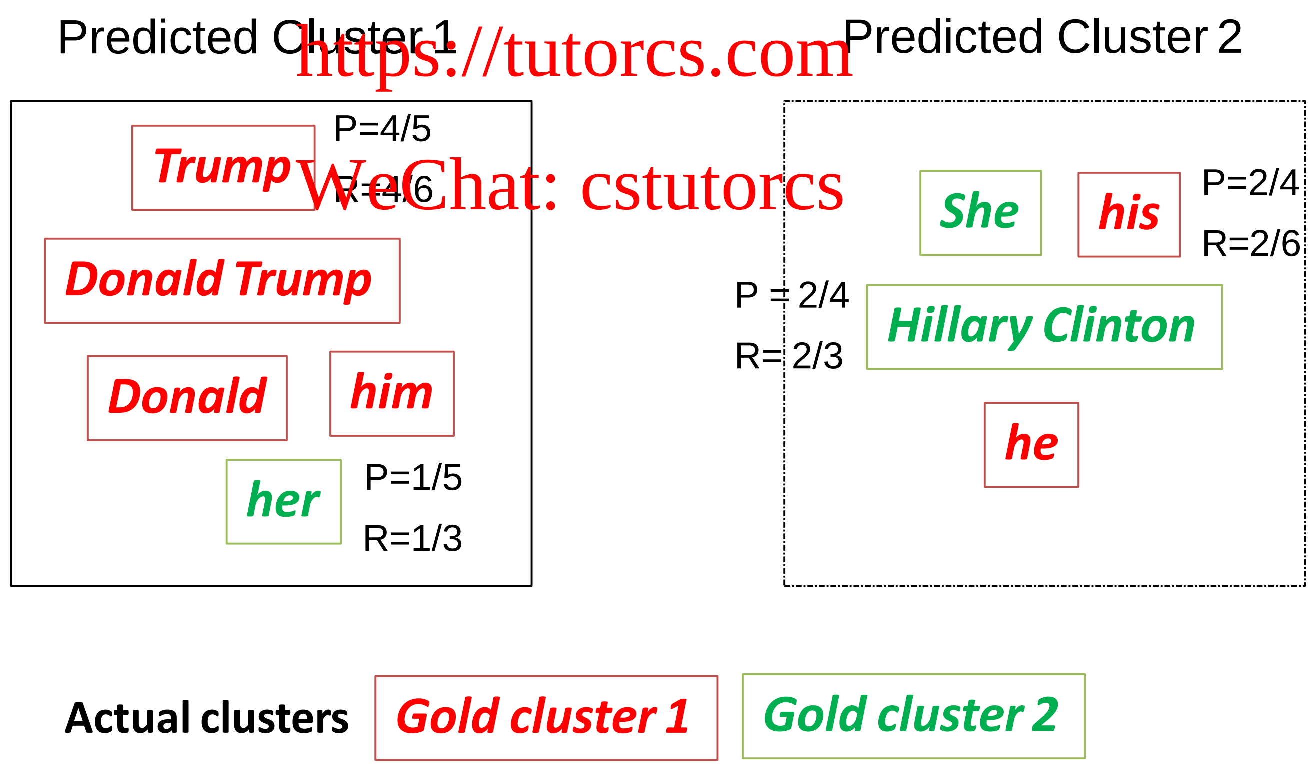


**How to evaluate coreference?**

Let’s evaluate with B-CUBED metrics

- Compute **P**recision and **R**ecall for each mention.





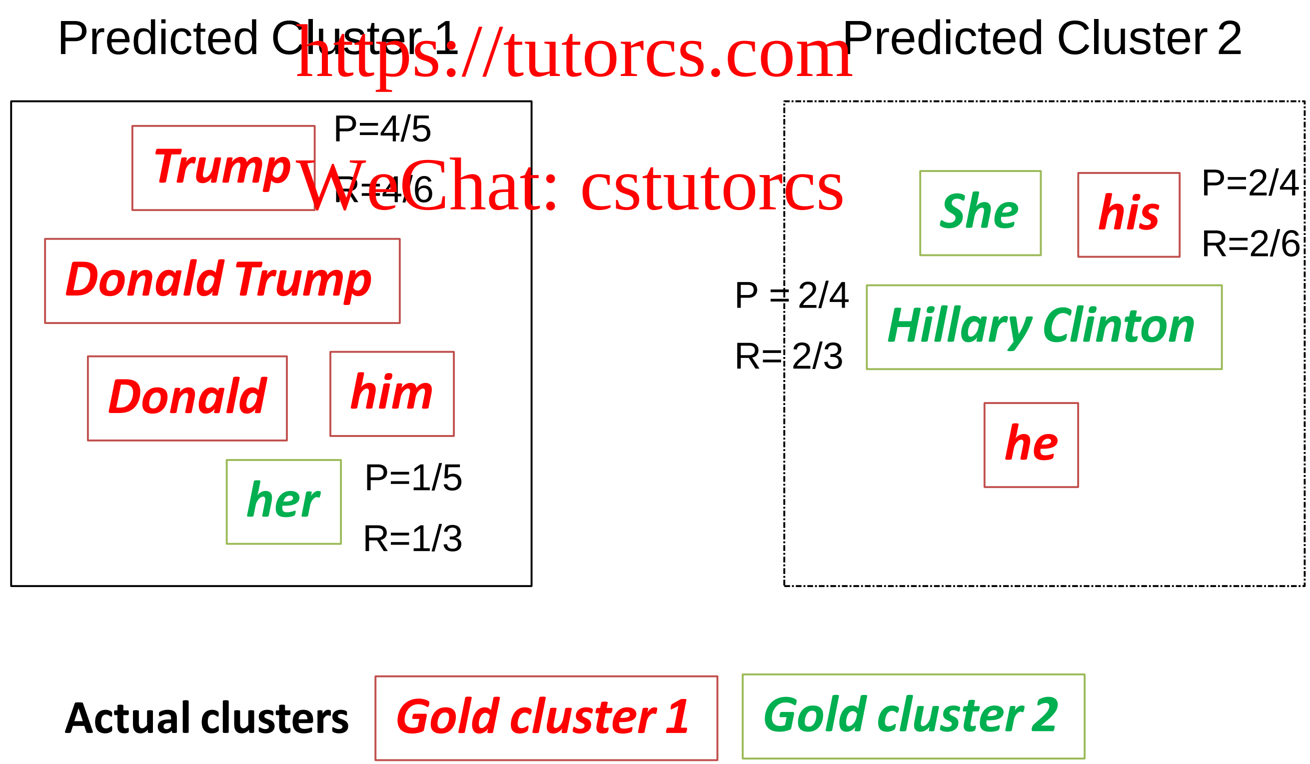
# 7 Coreference Evaluation

**How to evaluate coreference?**

Let’s evaluate with B-CUBED metrics

- Compute precision and recall for each mention.
- Average the individual Ps and Rs





# Coreference Evaluation

## Performance Comparison

OntoNotes dataset: ~3000 documents labeled by humans

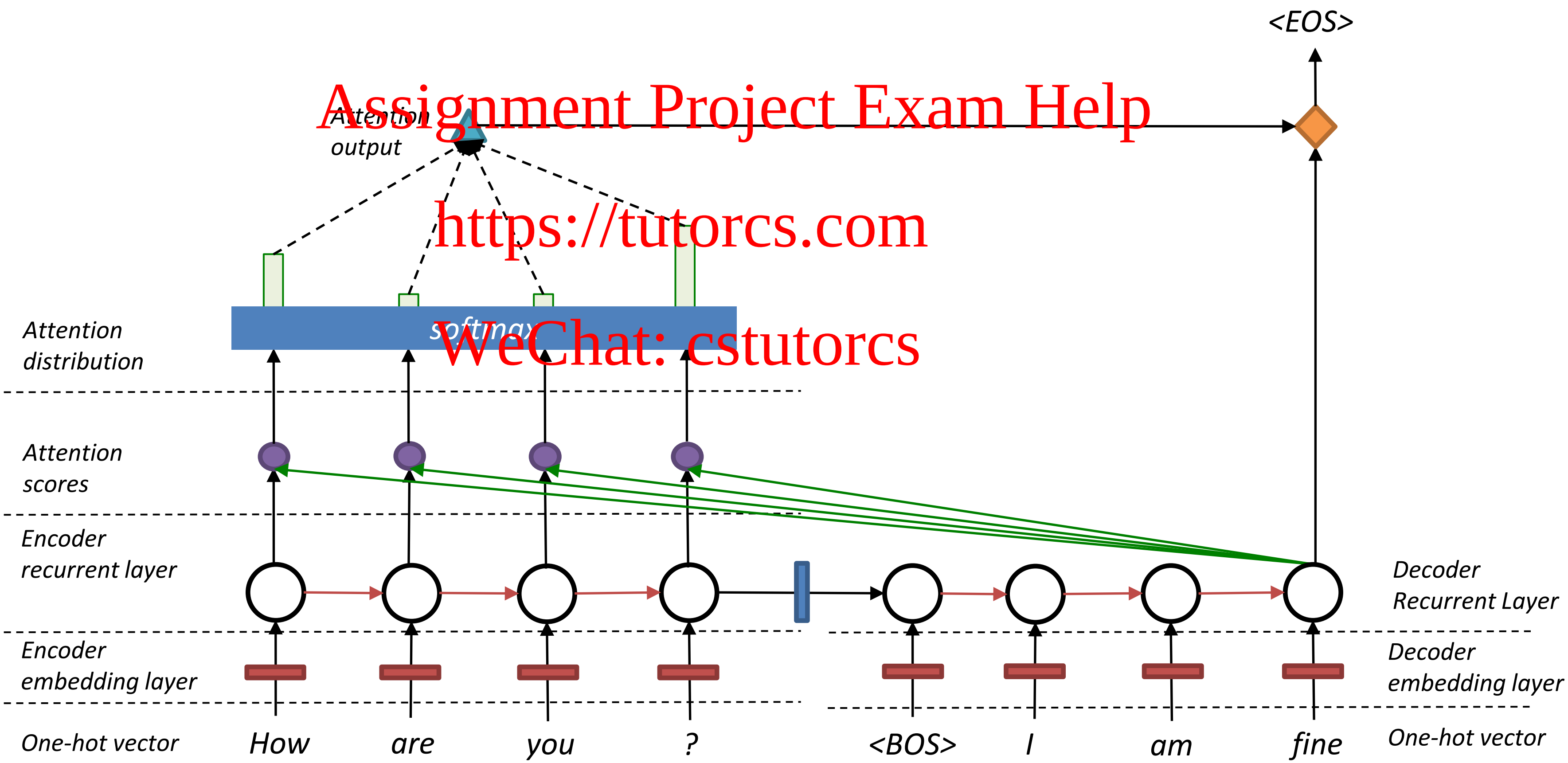
- English and Chinese data



| Model | Approach | English | Chinese |
|---|---|---|---|
| Lee et al. (2010) | Rule-based system | ~55 | ~50 |
| Chen & Ng (2012) [CoNLL 2012 Chinese winner] | Non-neural machine learning models | 54.5 | 57.6 |
| Fernandes (2012) [CoNLL 2012 English winner] | | 60.7 | 51.6 |
| Wiseman et al. (2015) | Neural mention ranker | 63.3 | — |
| Lee et al. (2017) | Neural mention ranker (end-to-end style) | 67.2 | -- |
| UsydNLP (2019) | Neural mention ranker with lemma cross validation | 74.87 | -- |

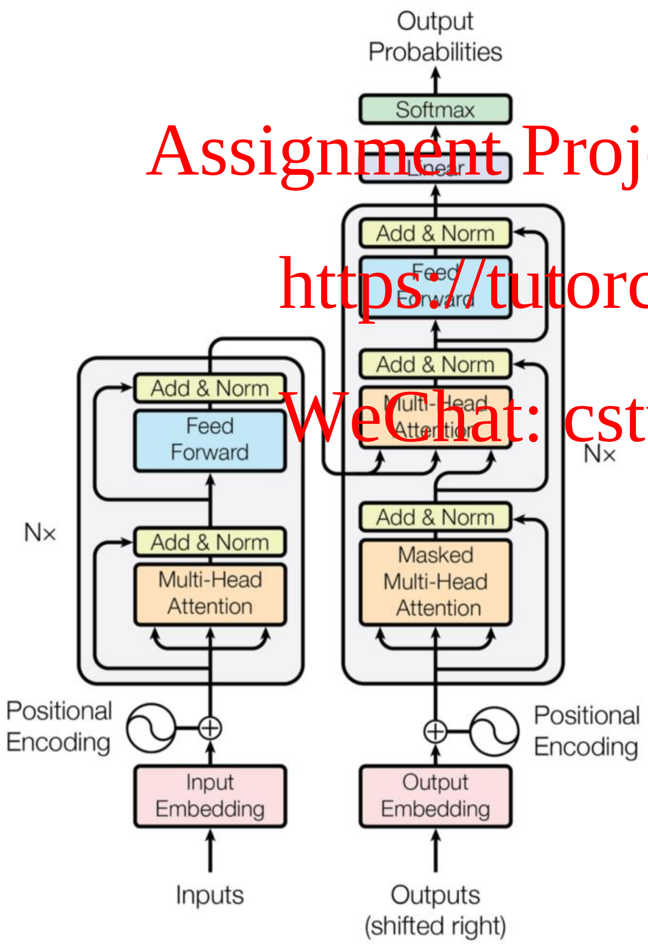
# Preview: Week 10

## Attention and Reading Comprehension



# 0 Preview: Week 11

## Transformer and Machine Translation



# / Reference

## Reference for this lecture


- Deng, L., & Liu, Y. (Eds.). (2018). Deep Learning in Natural Language Processing. Springer.
- Rao, D., & McMahan, B. (2019). Natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning. " O'Reilly Media, Inc.".
- Manning, C. D., Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing. MIT press.
- Manning, C 2018, Natural Language Processing with Deep Learning, lecture notes, Stanford University
- Li, J., Galley, M., Brockett, C., Gao, J., & Dolan, B. (2015). A diversity-promoting objective function for neural conversation models. arXiv preprint arXiv:1510.03055.
- Jiang, S., & de Rijke, M. (2018). Why are Sequence-to-Sequence Models So Dull? Understanding the Low-Diversity Problem of Chatbots. arXiv preprint arXiv:1809.01941.
- Liu, C. W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., & Pineau, J. (2016). How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. arXiv preprint arXiv:1603.08023.