



**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  
SINGAPORE

**EE6405**

# Natural Language Processing

**Dr. S. Supraja**  
**NTU Electrical and Electronic Engineering**

# Linguistic Analysis and Information Extraction

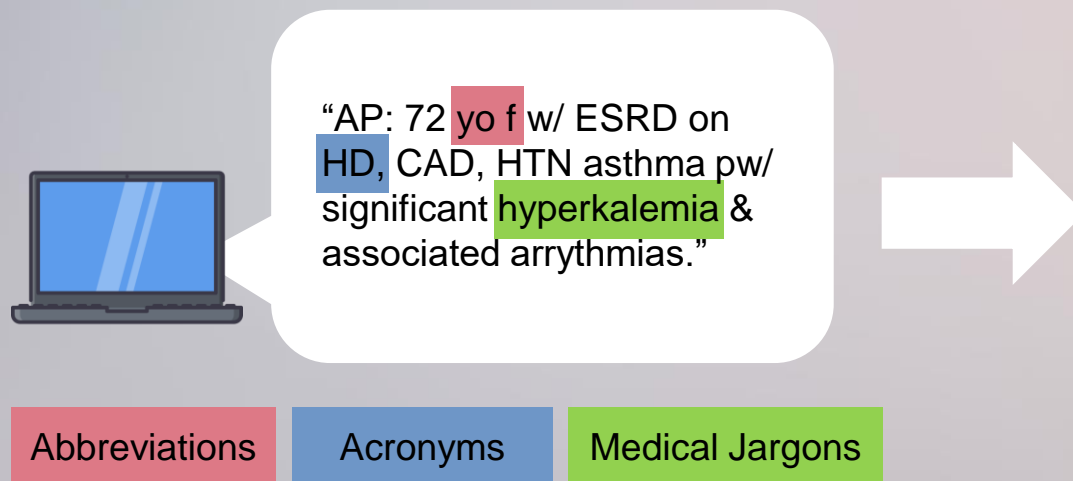


# Information Extraction

## Information Extraction Systems:

- Locate and comprehend pertinent sections of text.
- Summarise useful information across documents.
- Produce a structured representation of the information.

### Unstructured Data



### Meaningful Information

Attributes	Medical Text
Age	72 years old (yo)
Gender	Female (f)
Condition	Hypertensive disease (HD)
Symptom	hyperkalemia



## Goals:

### **Information Organisation:**

To organise information in a way that is useful for human understanding

### **Entity and Relationship Identification:**

To identify the relationship between entities (ie. names/organisations) through their relationships within text

### **Generation of Structured Data:**

To put information contained within text into a semantically precise form that allows for further inferences to be made by algorithms

# Named Entity Recognition (NER)

- Most important text information lies within named entities.
- These include names, locations, companies, dates, etc.
- Works by extracting the most important pieces of information from unstructured text.
- Delivers critical insights by picking up mentions of certain organisations/people.

The screenshot displays a text snippet with various named entities highlighted in colored boxes and labeled with a letter in a small box next to them. The labels are defined in a legend at the top: Person (p), Loc (l), Org (o), Event (e), Date (d), and Other (z).

Legend:

- Person: p
- Loc: l
- Org: o
- Event: e
- Date: d
- Other: z

Text snippet:

Barack Hussein Obama II (p) (born August 4, 1961 (d)) is an American (z) attorney and politician who served as the 44th President of the United States (l) from January 20, 2009 (d) to January 20, 2017 (d). A member of the Democratic Party (o), he was the first African American (z) to serve as president. He was previously a United States Senator (z) from Illinois (l) and a member of the Illinois State Senate (o).



# Named Entity Recognition (NER)

- Eg: Google uses NER to retrieve low-level information for certain searches.

NER being used to locate information on the birthplace of J.R.R. Tolkien within a body of text.

A screenshot of a Google search interface. The search bar contains the text "tolkien birthplace". Below the search bar are tabs for "Images", "News", "Maps", "Videos", "Books", "Flights", and "Finance". The search results show "About 232,000 results (0.61 seconds)". The first result is "John Ronald Reuel Tolkien / Place of birth" with the title "Bloemfontein, South Africa". The description states: "John Ronald Reuel Tolkien, author of The Hobbit and The Lord of the Rings, was born in Bloemfontein, an Afrikaans - speaking area of South Africa, on 3 January 1892. His father had become a bank manager there." Below the description is a link from "Birmingham City Council" with the URL "https://www.birmingham.gov.uk>info>about\_jrr\_tolkien". At the bottom, there is a link "Tolkien's early years | About J.R.R ... - Birmingham City Council".

tolkien birthplace


Images News Maps Videos Books Flights Finance

About 232,000 results (0.61 seconds)

John Ronald Reuel Tolkien / Place of birth

## Bloemfontein, South Africa

John Ronald Reuel Tolkien, author of *The Hobbit* and *The Lord of the Rings*, was born in Bloemfontein, an Afrikaans - speaking area of South Africa, on 3 January 1892. His father had become a bank manager there.

 Birmingham City Council  
[https://www.birmingham.gov.uk>info>about\\_jrr\\_tolkien](https://www.birmingham.gov.uk>info>about_jrr_tolkien)

[Tolkien's early years | About J.R.R ... - Birmingham City Council](#)

# Named Entity Recognition (NER)

## NER primarily focuses on:

- Organisations
- Locations
- Dates
- Persons
- Events
- These entities can be adjusted depending on the nature of the task.
  - E.g.: A materials-science related model could feature sector-specific entities such as material names, manufacturing processes, etc.

# Named Entity Recognition (NER)

**Any NER task needs to accomplish two basic goals:**

Detecting a named entity.  
Detecting a word or a string of words that form an entity.  
With each word representing a token, “United Overseas Bank” is a string of three tokens representing one entity.

Categorising the entity.  
Entity categories need to be created based on the task at hand. Common categories include people, organisation and time. Granular rules need to be created in order to classify these entities into their respective subcategories.



# Named Entity Recognition – Using spaCy

```
import spacy
from spacy import displacy

NER = spacy.load("en_core_web_sm")
```

Displacy is a built-in visualiser in SpaCy, it'll help us better visualise the data later.

en\_core\_web\_sm is a pretrained English pipeline that includes an NER tagger.

```
text1= NER(raw_text)

for word in text1.ents:
    print(word.text,word.label_)
```

These tokens can be accessed via the ents property.

The NER() function returns tagged NEs as a span (ordered list of tokens)

# Named Entity Recognition (NER)

## Raw Text:

From 1925 to 1945, Tolkien was the Rawlinson and Bosworth Professor of Anglo-Saxon and a Fellow of Pembroke College, both at the University of Oxford. He then moved within the same university to become the Merton Professor of English Language and Literature and Fellow of Merton College, and held these positions from 1945 until his retirement in 1959. Tolkien was a close friend of C. S. Lewis, a co-member of the informal literary discussion group The Inklings. He was appointed a Commander of the Order of the British Empire by Queen Elizabeth II on 28 March 1972.

## NER Tags

1925 to 1945 DATE  
Tolkien PERSON  
Rawlinson PERSON  
Anglo NORP  
Fellow of Pembroke College ORG  
the University of Oxford ORG  
the Merton Professor of English Language and Literature ORG  
Fellow of Merton College ORG  
1945 DATE  
1959 DATE  
Tolkien PERSON  
C. S. Lewis PERSON  
Inklings ORG  
the British Empire GPE  
Elizabeth II PERSON  
28 March 1972 DATE

# Named Entity Recognition (NER)

To render the text:

```
displacy.render(text1, style="ent", jupyter=True)
```

From 1925 to 1945 DATE , Tolkien PERSON was the Rawlinson PERSON and Bosworth Professor of Anglo NORP -Saxon and a Fellow of Pembroke College ORG , both at the University of Oxford ORG . He then moved within the same university to become the Merton Professor of English Language and Literature ORG and Fellow of Merton College ORG , and held these positions from 1945 DATE until his retirement in 1959 DATE . Tolkien PERSON was a close friend of C. S. Lewis PERSON , a co-member of the informal literary discussion group The Inklings ORG . He was appointed a Commander of the Order of the British Empire GPE by Queen Elizabeth II PERSON on 28 March 1972 DATE .



# Named Entity Recognition (NER)

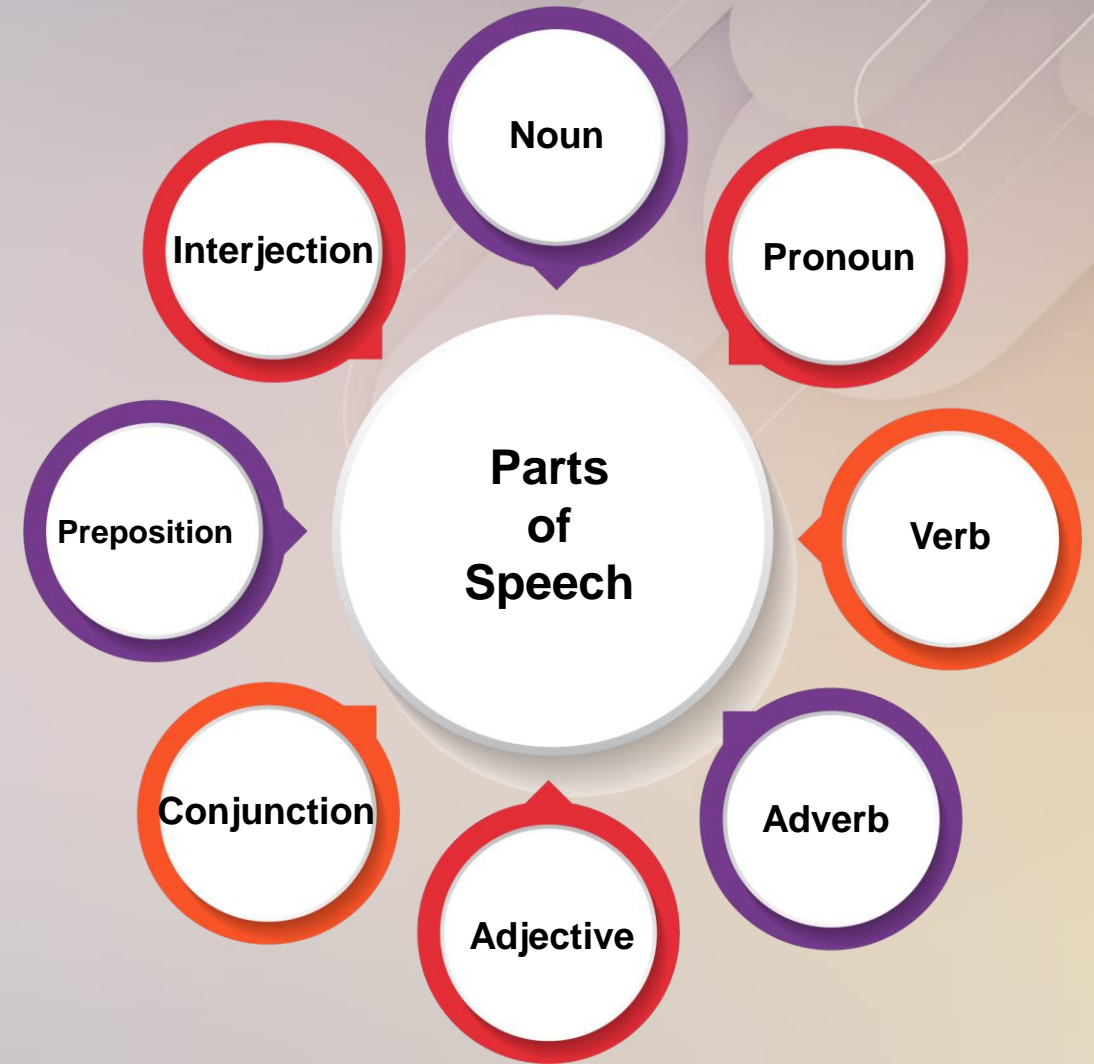


## Applications

- Performing sentiment analysis toward a company or product.
- Relations between entities represent a large number of IE relations.
- Answer detection – Answers often come in the form of named entities.
- Low-level information retrieval – Assists in queries on the nature/history of an entity.

# Part-Of-Speech Tagging

- Words found in natural languages can be categorised based on their roles and functions within a sentence. Common categories include:
  - Nouns
  - Verbs
  - Adjectives
  - Adverbs
  - Conjunctions



# Part-Of-Speech Tagging

- By analysing the structural and semantic context of words in speech, text can be processed more accurately.
- E.g.: Depending on context, the word 'run' can either be a noun or a verb. POS tagging allows computers to contextualise these words for more accurate processing and comprehension.

Noun:

The children went for a **run** in the park.

Verb:

She likes to **run** every morning to stay fit.



# Part-Of-Speech Tagging

- POS tagging aims to automate the task of tagging parts-of-speech to determine the syntactic and grammatical role of each word in the context of a sentence.
- Tagging can be done using linguistic patterns, context and predefined dictionaries.
- POS tagging can also be achieved using probability models, such as Hidden Markov Models.

# Part-Of-Speech Tagging



- Hidden Markov Models:
  - HMM (Hidden Markov Model) is a stochastic technique for POS tagging.
  - Works by leveraging state transitions and observations to determine the most likely sequence for POS tags in a sentence.



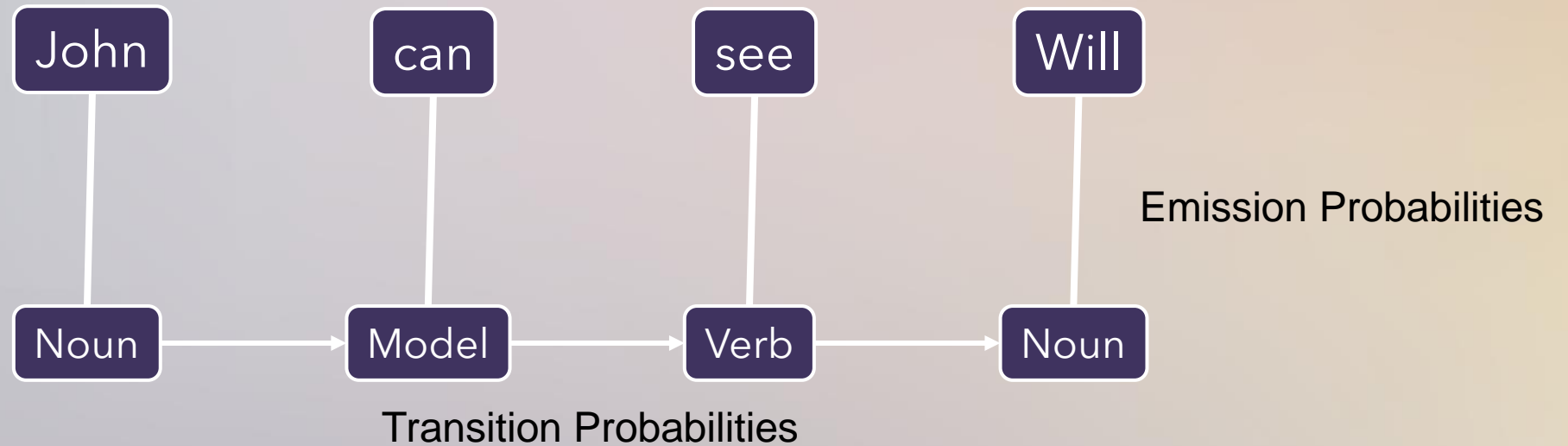
- Components of a HMM include:
  - States (Specific POS tags like 'noun', 'verb', 'adjective')
  - Observations (Each observation corresponds to a word in a sentence)



- Two sets of probabilities are used:
  - Transition Probability:  
The probability of **transitioning** from one POS tag to another.
  - Emission Probability:  
The probability of a certain word being **emitted** from a POS tag.

# Part-Of-Speech Tagging

- HMMs (Worked Example):
  - Consider a simple tagger with 3 tags:
    - Noun
    - Model
    - Verb



# Part-Of-Speech Tagging

- Calculating Emission Probabilities:
- Sentence Bank:
  - Mary Jane can see Will.
  - Spot will see Mary.
  - Will Jane spot Mary?
  - Mary will pat Spot.

# Part-Of-Speech Tagging

- Calculating Emission Probabilities:
  - Frequency of occurrence:

Word	Noun	Model	Verb
Mary	4	0	0
Jane	2	0	0
Will	1	3	0
Spot	2	0	1
can	0	1	0
see	0	0	2
pat	0	0	1

# Part-Of-Speech Tagging

- Calculating Emission Probabilities:

$$\frac{\text{Occurrence}}{\text{Total}} = \text{Emission Probability:}$$

Word	$P(\text{Noun})$	$P(\text{Model})$	$P(\text{Verb})$
Mary	4/9	0	0
Jane	2/9	0	0
Will	1/9	3/4	0
Spot	2/9	0	1/4
can	0	1/4	0
see	0	0	2/4
pat	0	0	1/4



# Part-Of-Speech Tagging

- Calculating Transition Probabilities:
  - Define start and end tags as <s> and <e> respectively;

Sentence Bank:

- <s> Mary Jane can see Will. <e>
- <s> Spot will see Mary. <e>
- <s> Will Jane spot Mary? <e>
- <s> Mary will pat Spot. <e>

Calculate co-occurrence probability: E.g.: <s> is followed by a noun  $\frac{3}{4}$  times as above.

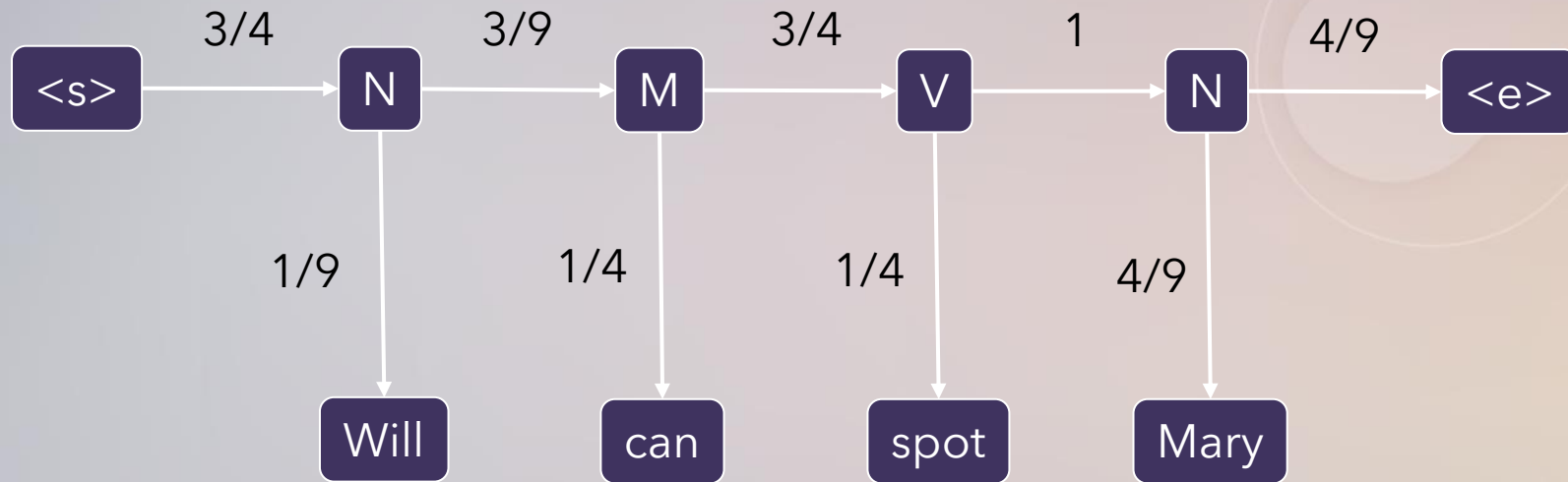
# Part-Of-Speech Tagging

- Calculating Emission Probabilities:
  - Calculate co-occurrence counts:

	Noun	Model	Verb	<e>
<s>	3/4	1/4	0	0
Noun	1/9	3/9	1/9	4/9
Model	1/4	0	3/4	0
Verb	4/4	0	0	0

# Part-Of-Speech Tagging

- Calculating Transition Probabilities:
  - Calculate co-occurrence counts:

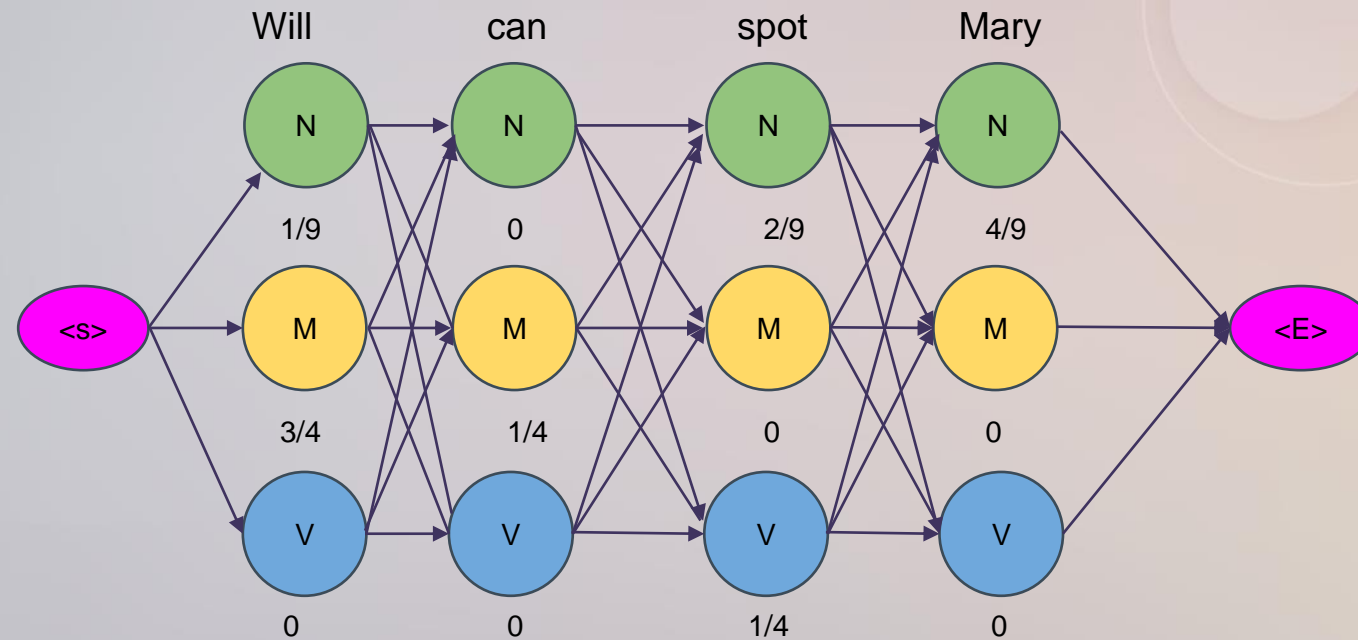


$$\frac{3}{4} * \frac{1}{9} * \frac{3}{9} * \frac{1}{4} * \frac{3}{4} * \frac{1}{4} * 1 * \frac{4}{9} * \frac{4}{9} = 0.00025720164$$

By calculating these probabilities, we can determine the most likely tags for words in the sentence.

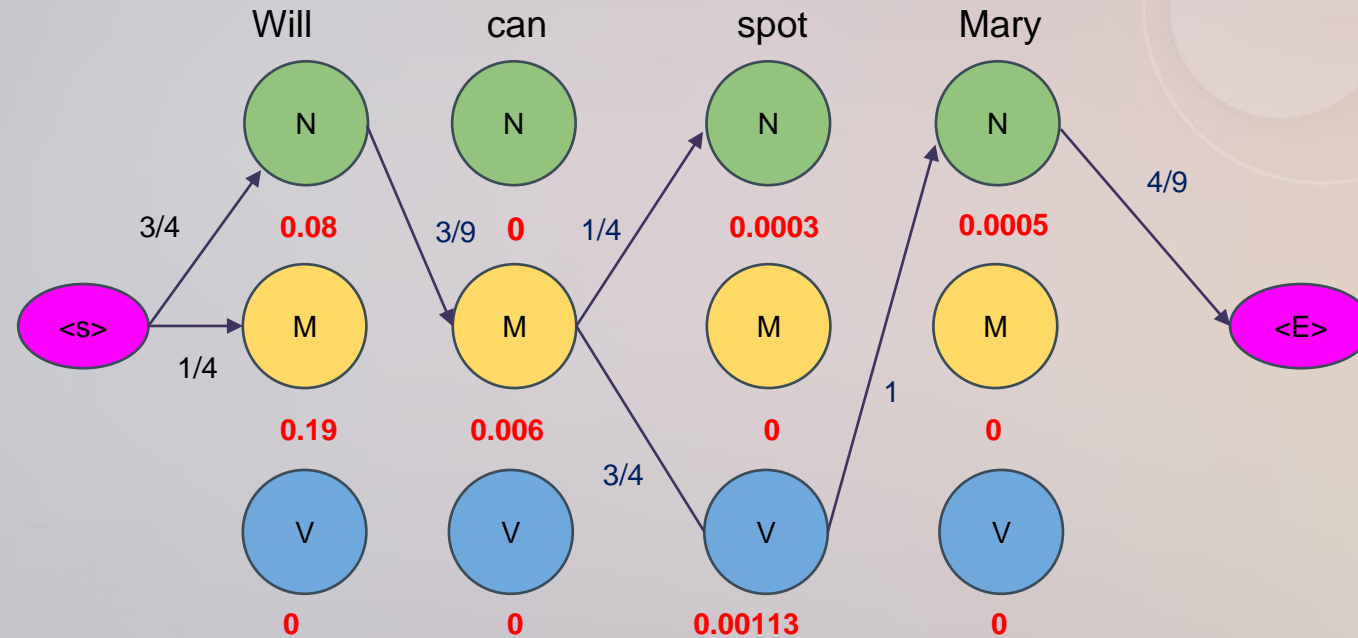
# Part-Of-Speech Tagging

- Calculating Transition Probabilities
  - Probability tree for all possible tags



# Part-Of-Speech Tagging

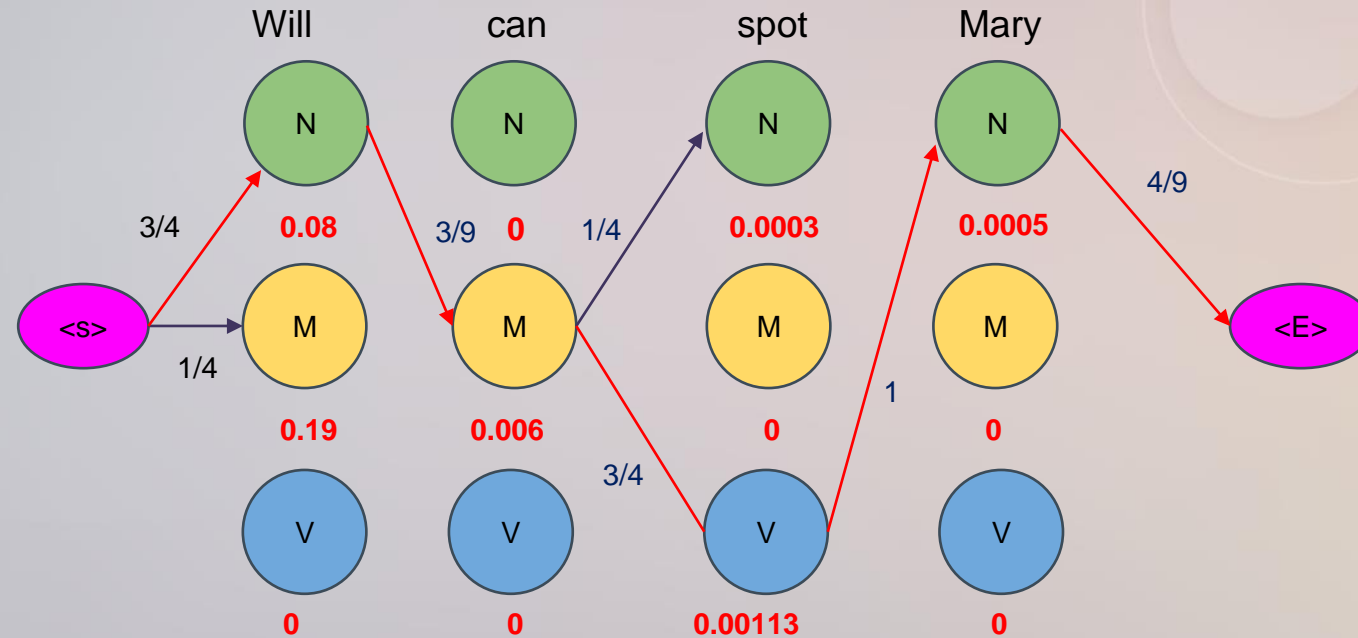
- Optimisation with the Viterbi algorithm:
  1. Store all intermediate probabilities for each state transition.



# Part-Of-Speech Tagging

- Optimisation with the Viterbi algorithm:

2. From the endpoint, backtrace the path with the highest probability to the start point.





# Part-Of-Speech Tagging (Pretrained tagger in spaCy)

```
import pandas as pd

text = ['You know the greatest lesson of history?',
        'It's that history is whatever the victors say it is.',
        'That's the lesson. Whoever wins, that's who decides the history.']

df = pd.DataFrame(text, columns=['Sentence'])

df
```

```
import spacy

#load the small English model
nlp = spacy.load("en_core_web_sm")

#list to store the tokens and pos tags
token = []
pos = []

for sent in nlp.pipe(df['Sentence']):
    if sent.has_annotation('DEP'):
        #add the tokens present in the sentence to the token list
        token.append([word.text for word in sent])
        #add the pos tage for each token to the pos list
        pos.append([word.pos_ for word in sent])
```

nlp.pipe allows text to be processed in batches rather than one-by-one

# Part-Of-Speech Tagging

## Original Sentences:

Sentence
You know the greatest lesson of history?
It's that history is whatever the victors say it is.
That's the lesson. Whoever wins, that's who decides the history.

## Tokenized Sentences:

token
You, know, the, greatest, lesson, of, history, ?
It, 's, that, history, is, whatever, the, victors, say, it, is, .
That, 's, the, lesson, ., Whoever, wins, ,, that, 's, who, decides, the, history, .

## POS Tags:

pos
PRON, VERB, DET, ADJ, NOUN, ADP, NOUN, PUNCT
PRON, VERB, SCONJ, NOUN, AUX, PRON, DET, NOUN, VERB, PRON, AUX, PUNCT
PRON, VERB, DET, NOUN, PUNCT, PRON, VERB, PUNCT, PRON, VERB, PRON, VERB, DET, NOUN, PUNCT

Visualising **VERB**

Part-of-Speech  
**NOUN**

Tags **NOUN**

With **ADP**

NLTK **NOUN**

and **CONJ**

Spacy **NOUN**

# Part-Of-Speech Tagging

## Applications:



- Text classification: POS tagging can aid in categorizing texts into various groups, by conducting sentiment analysis. Through the examination of the part-of-speech tags assigned to words within a text, algorithms can better understand the text's subject matter.



- Machine translation: By identifying the grammatical structure and relationships between words in the source language, and mapping them to the target language, POS tagging can be used to help translate texts.



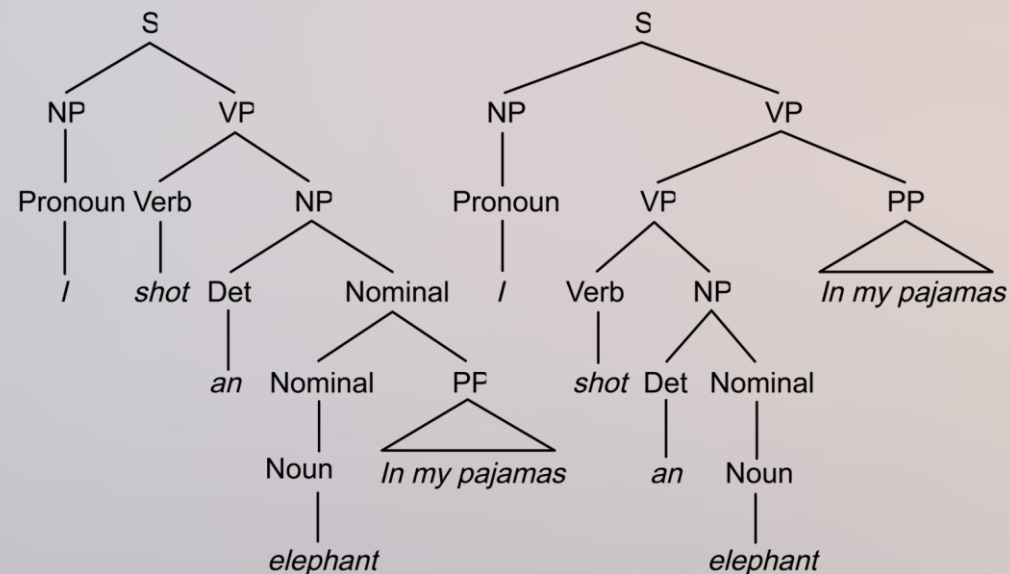
- Natural language generation: POS tagging can be used to generate natural-sounding text by selecting appropriate words and constructing grammatically correct sentences. This is useful for tasks such as chatbots and virtual assistants.

# Dependency Parsing

- Dependency parsing is a natural language processing technique that analyses the grammatical structure of a sentence by identifying the relationships between words.
- Dependency Parsing offers a more detailed analysis of sentence structure compared to other parsing methods.
- Enables a deeper understanding of the relationships between words, enhancing context comprehension.

# Dependency Parsing

- Dependency parsing identifies which words are the main components (heads) and which words depend on them (modifiers).
- Each relationship is assigned a label (e.g., subject, object, modifier) to indicate the grammatical role of the dependent word in relation to the head word.
- These relationships form a tree-like structure, revealing the hierarchical organisation of the sentence.



# Dependency Parsing

- Worked Example (spaCy):

```
# Load the language model
nlp = spacy.load("en_core_web_sm")

sentence = 'I saw a kitten eating chicken in the kitchen.'

# nlp function returns an object with individual token information,
# linguistic features and relationships
doc = nlp(sentence)

print("{:<15} | {:<8} | {:<15} | {:<20}".format('Token', 'Relation', 'Head', 'Children'))
print("-" * 70)

for token in doc:
    # Print the token, dependency nature, head and all dependents of the token
    print("{:<15} | {:<8} | {:<15} | {:<20}"
          .format(str(token.text), str(token.dep_), str(token.head.text), str([child for child in token.children])))

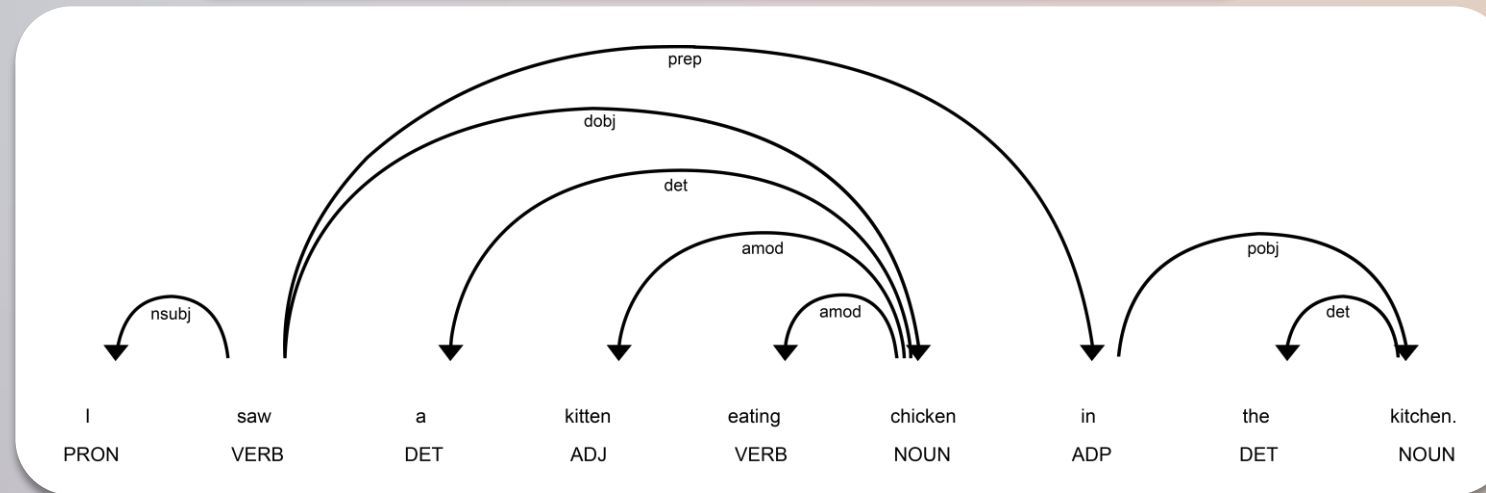
# Use displayCy to visualize the dependency
displacy.render(doc, style='dep', jupyter=True, options={'distance': 120})
```



# Dependency Parsing

- Worked Example (spaCy):

Token	Relation	Head	Children
I	nsubj	saw	[]
saw	ROOT	saw	[I, chicken, in, .]
a	det	chicken	[]
kitten	amod	chicken	[]
eating	amod	chicken	[]
chicken	dobj	saw	[a, kitten, eating]
in	prep	saw	[kitchen]
the	det	kitchen	[]
kitchen	pobj	in	[the]
.	punct	saw	[]



# Summary

## Key points discussed this week:

- **Information Extraction Systems including:**
  - **Named Entity Recognition:**  
Process of identifying and categorising specific named entities, such as names, dates, locations, and more, within text.
- **Part Of Speech Tagging:**  
Assigning grammatical labels to words in a sentence to indicate their syntactic roles and categories, like nouns, verbs, adjectives, etc.
- **Dependency Parsing:**  
Parsing technique that analyses the syntactic structure of a sentence by identifying the relationships between words, showing how they depend on one another.
- **spaCy:**  
A Python library that provides tools and resources for information extraction.

# References

- <https://www.hitachi.com/rd/sc/aiblog/021/index.html>
- <https://www.analyticsvidhya.com/blog/2021/11/a-beginners-introduction-to-ner-named-entity-recognition/>

No part of this video shall be filmed, recorded, downloaded, reproduced, distributed, republished or transmitted in any form or by any means without written approval from the University.