

# Hons. Degree in Computing

## H4016 Text Analysis

Unit 2 – Preparing text data for data mining

Part II: Polysemy and the Hidden Markov model

# Recap on last week:

## Text Mining Process

### 5. Analyzing Results

### 4. Text/Data Mining

- Classification- Supervised Learning
- Clustering- Unsupervised Learning

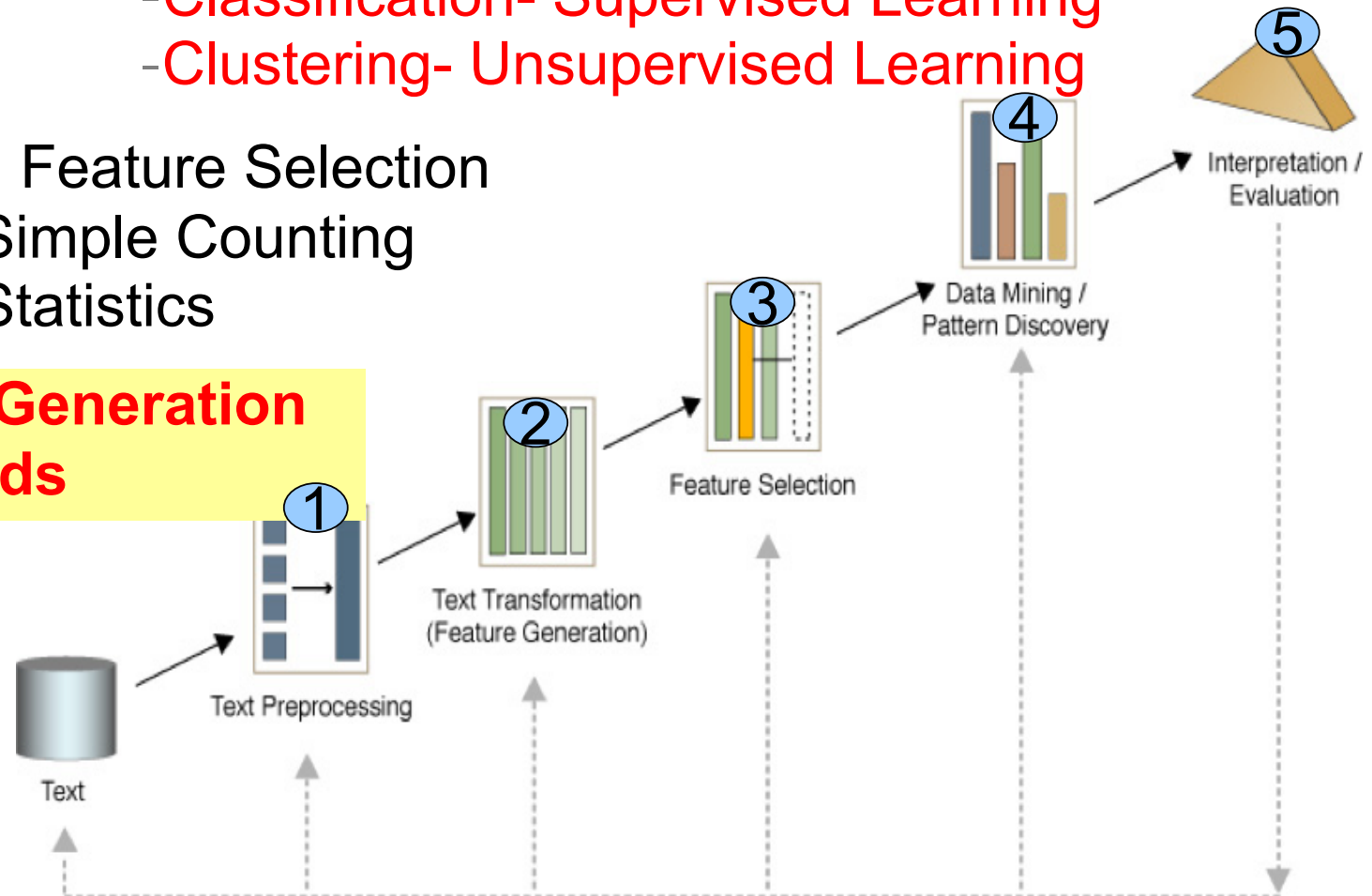
### 3. Feature Selection

- Simple Counting
- Statistics

### 2. Features Generation

- Bag of Words

### 1. Collect text and identify business objective



The last lecture identified the following techniques for improving on bag of words, and covered points 1 to 5. We will now focus on point 6 . . .

Improving on bag of words can include any of the following:

- 1) Recognise **Collocations (expressions) & phrases** – **with respect to, second hand, drop in**
- 2) Recognise **synonyms** – **phone, fone, telephone, mobile.**
- 3) **Concept typing**: recognising concepts as **person, place, business** etc.
- 4) **Intent** – recognising complaints, requests, praise etc.
- 5) Link words with the same **canonical form**: **'runner', 'ran', 'running'** are all from the stem 'run'

6) **recognise polysemy** – e.g. The word '**state**' can have many meanings.

# 5. Polysemy

Polysemy is a single word with different meanings

Words have 1.4 meanings on average

Verbs are worst, with 2.1 meanings on average

Some words have more than one part of speech (POS). For example, **Still** can be used as an **adjective**, a **noun**, a **verb**, an **adverb** or a **conjunction**:

- The still air. (adjective)
- A still is used to distill liquids. (noun)
- To still a craving. (verb)
- He is still running; Are you still here? (adverb)
- It was futile, still they fought. (conjunction)
- Collocations: still and all

# Word Sense Disambiguation (WSD)

*"You shall know a word by the company it keeps" (Firth, 1957)*

Common words tend to have more senses, for example **break**, **run**, **bank** or **state** each have at least 10 meanings.

Humans do not always agree on the meaning of a word when interpreting text. Language, by nature, is ambiguous.

The following section discusses approaches for finding the best meaning of a word, given its context.

The objective of using these approaches is to improve the accuracy of text analytics results.

# Word Sense Disambiguation (WSD)

For example: State has 25 meanings in dictionary.com

... some of these are:

Can you **state** your opinion

Say

To **state** a problem

Define

Toronto is in the **state** of Ontario

Province

The **state** has lowered its income tax level

Nation

The solid **state** of water is ice

Matter

He was in a bad **state**

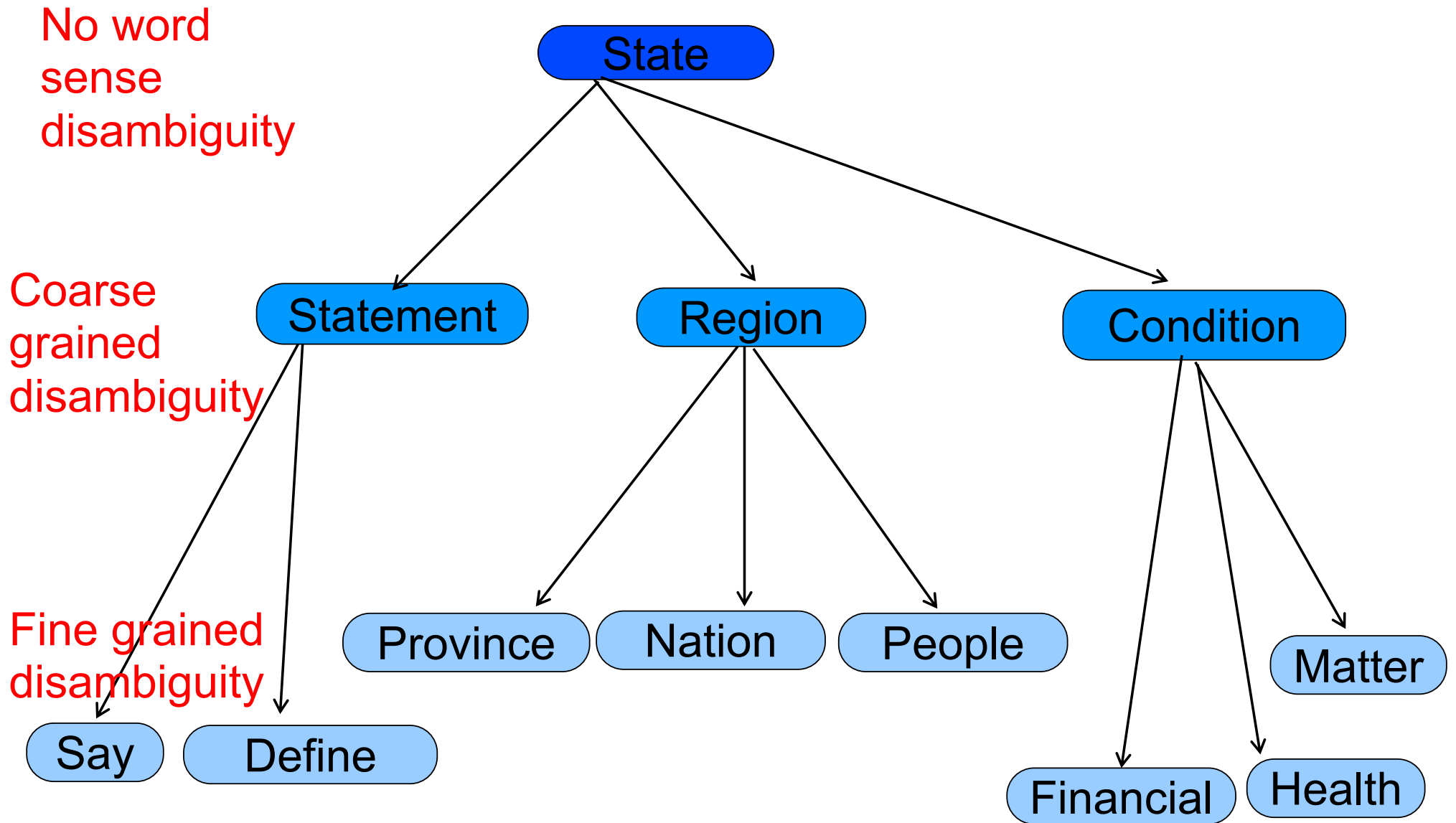
Health

The company is in a weak financial **state**

Financial

Some word senses are closer in meaning than others. So differences in senses can be coarse grained or fine grained.

# Word Sense Disambiguation (WSD)



# Word Sense Disambiguation (WSD) & Parts of Speech (POS) tagging

So how do you decide which is the correct part of speech for ambiguous words?

- Assigning the most common tag to each known word and the tag "proper noun" to all unknowns, will approach **90% accuracy** because many words are unambiguous.
- 97% of all words are either noun, verb, adverb or adjective
- 48% of ambiguous words have two possible tags, noun or verb
- 37% of ambiguous words can be tagged as either noun or adjective

To achieve higher rates of accuracy, some classification technique must be used to classify a word with the correct part of speech

- The most common approach is to use a Hidden Markov Model (a type of Bayesian Network)



# Markov Models and Hidden Markov Models

Hidden Markov models (**HMM**) are used to solve a number of NLP problems, including POS tagging.

**But before we look at HMM's, we will first look at Markov Models:**

Markov models explore the fact that letters, and words, do not occur at random in natural language.

- 'e' and 't' occur more often than other letters
- 'the' occurs more often than other words

We can view text as a stream of words, ordered by rules of grammar.

- Any word **x** in the stream will have **n** possible other words before it, and **m** possible other words after it.

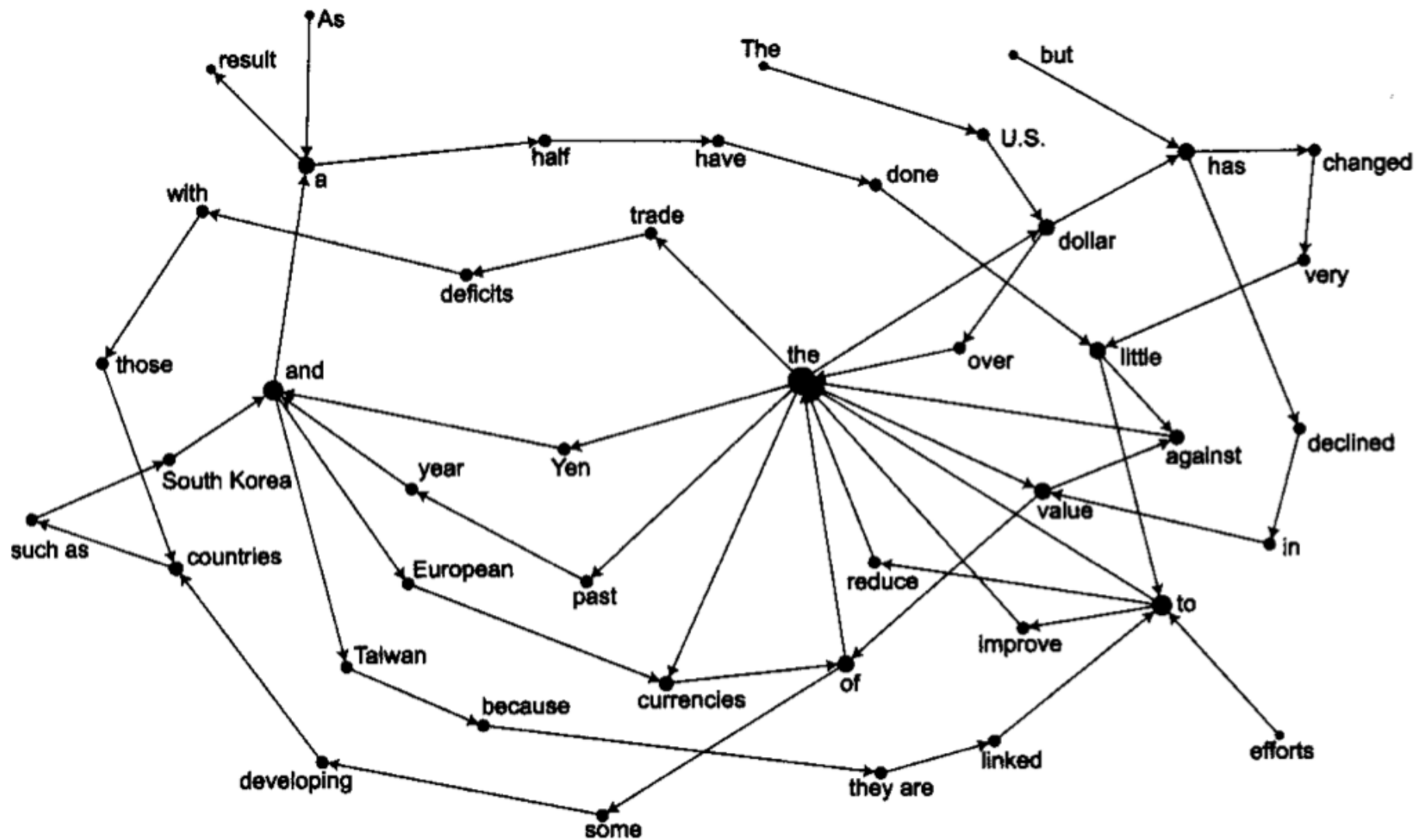
This can be modelled as a directed graph, showing the sequence in which words can occur. . .

# Markov Model

The following slide shows the Markov model generated from the following two sentences. Commas and full stops start a new path.

The U.S. Dollar has declined in value against the Yen and European currencies, but has changed very little against the currencies of some developing countries such as South Korea and Taiwan because they are linked to the value of the dollar.

As a result, efforts to reduce the value of the dollar over the past year and a half have done little to improve the trade deficits with those countries.



**FIGURE 4.2** Network of word sequences from two sentences.

# Markov model

Points to note:

- Nodes with a higher number of inputs and outputs are larger: **the**, **to**, **and** & **dollar** are some of the more popular words in this network
- Even for a short paragraph of text, the model is quite complex
- Words are case sensitive (e.g. **The** has a different node from **the**)
- The network can include compound tokens (e.g. **South Korea**; **such as**)

A Markov model can be used to generate artificial sentences by following arrows on the graph:

- e.g. “**The U.S. dollar has changed very little to improve the past year.**”
  - Syntactically is this a reasonable sentence?
  - Semantically does it make sense?
- What about “**The U.S. dollar over the trade deficits with those countries**”

# Hidden Markov model

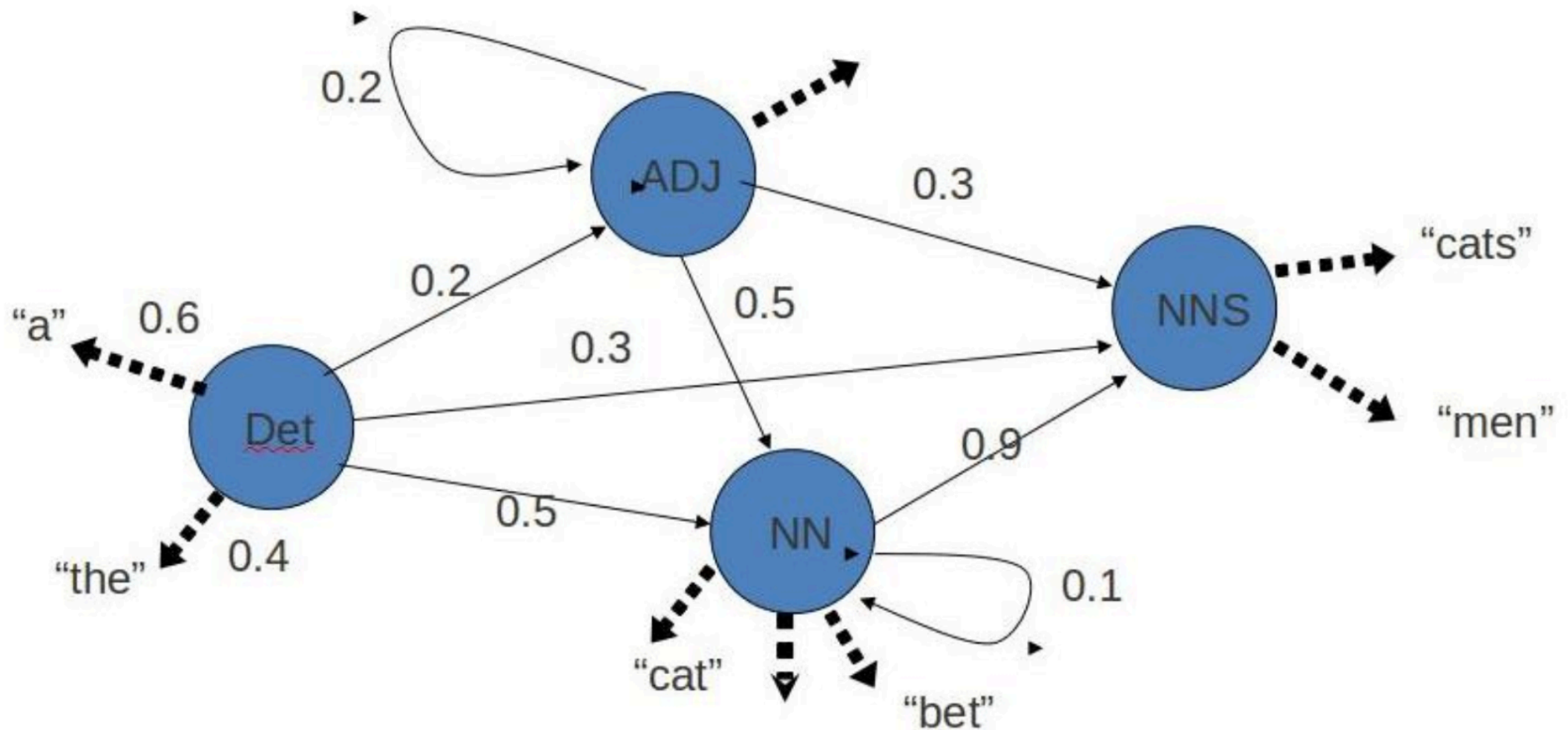
Slide 11 is a Markov model, where you know in advance what state (node / word) you are at.

In recognising parts of speech, a **hidden Markov model** is used.

It differs from a Markov model in that:

- It's a generic representation of Parts of Speech (POS) rather than actual words.
  - Each state is a POS (.e.g noun, verb etc) rather than a word.
- States are linked by a probability, determining the likelihood that those two states would appear beside each other in a sentence.

# Example of a transition network; one part of a Hidden Markov model for POS tagging



# Hidden Markov model

You can use a Hidden Markov model to determine what state (node / POS) you are likely to be at.

- Suppose you have a token “**coin**” from the sentence “...**to coin a phrase**..”. You want to determine if ‘**coin**’ refers to the **noun** meaning **money**, or a **verb** meaning **invent**.
- Each POS for the word “**coin**” has its own state (node) in the Markov diagram. You don't know in advance which state you are at (i.e. the **verb** or the **noun** node), so you want to determine the likelihood for each possible state.

# Hidden Markov model

A Hidden Markov Model (HMM) has three components:

1. Starting probabilities: In the English language, is this term more likely to be used as a noun, a verb, or another part of speech e.g. how often is this word used as a noun?

2. Transitions: A network of states, that has probabilities associated with the transitions between states. Probabilities are calculated based on observed sequences of states in training data.

e.g. How likely is it, that this term is a noun, giving the parts of speech of the words before and after it?

3. Observations: The sentence you are working on



# Hidden Markov model

## 1. Starting state:

The starting probabilities are how often a term appears in a particular state:

From a training corpus (text data), calculate how common each state of a term is, generating a table of probabilities as follows:

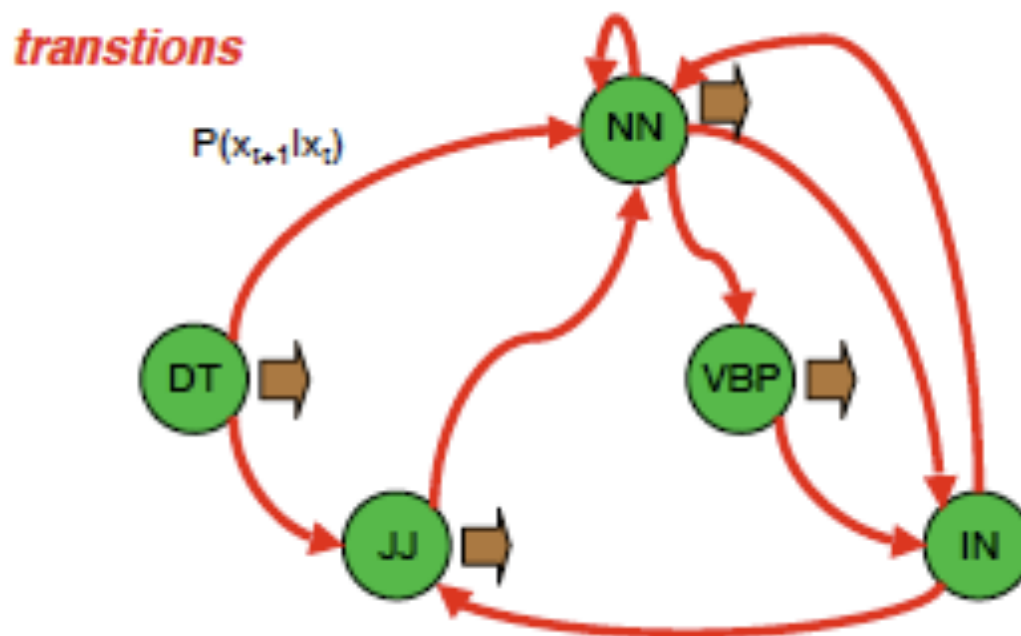
Term	State	Frequency
Coin	Noun	0.95 (95%)
Coin	Verb	0.05 (5%)

# Hidden Markov model

## 2. Transitions:

The **transitions** are the probability of one POS being adjacent to another in a sentence.

The follow slides will look at how to construct a transition diagram . . .



# Hidden Markov model

## 2. Transitions

Probabilities can be calculated from an existing tagged corpus/text.

A simple model would give a small number of states, i.e. just use:

- Nouns, Adjectives, Verbs, Adverbs, Conjunctions, Determiners(**both**), Pronouns (**he**), Prepositions (**by**), Interjections (**convey emotion - ouch**) and punctuation marks(!)

A more complex model could have many more states, e.g.

- Two types of nouns - **proper** and **common**, can be further subdivided into **person, place, thing, quality or idea**
- Many types of pronouns: **personal, relative, interrogative, reflexive, intensive, demonstrative** and **indefinite**.
- Three types of verbs: **regular, irregular** and **linking**. Can be further subdivided into **action** or a **state of being**.

# Example: Creating a HMM

Sample Text:

*\*stop electing life peers \*\**

*by Trevor Williams*

*A move to stop Mr. Gaitskell from nominating any more labour like peers is to be made at a meeting of labour MPs tomorrow. Mr Michael Foot has put down a resolution on the subject and he is to be backed by Mr Will Griffiths, MP for Manchester Exchange.*

Tagged text

*\* \_ stop\_VB electing\_VBG life\_NN peers\_NNS \*\* \_*

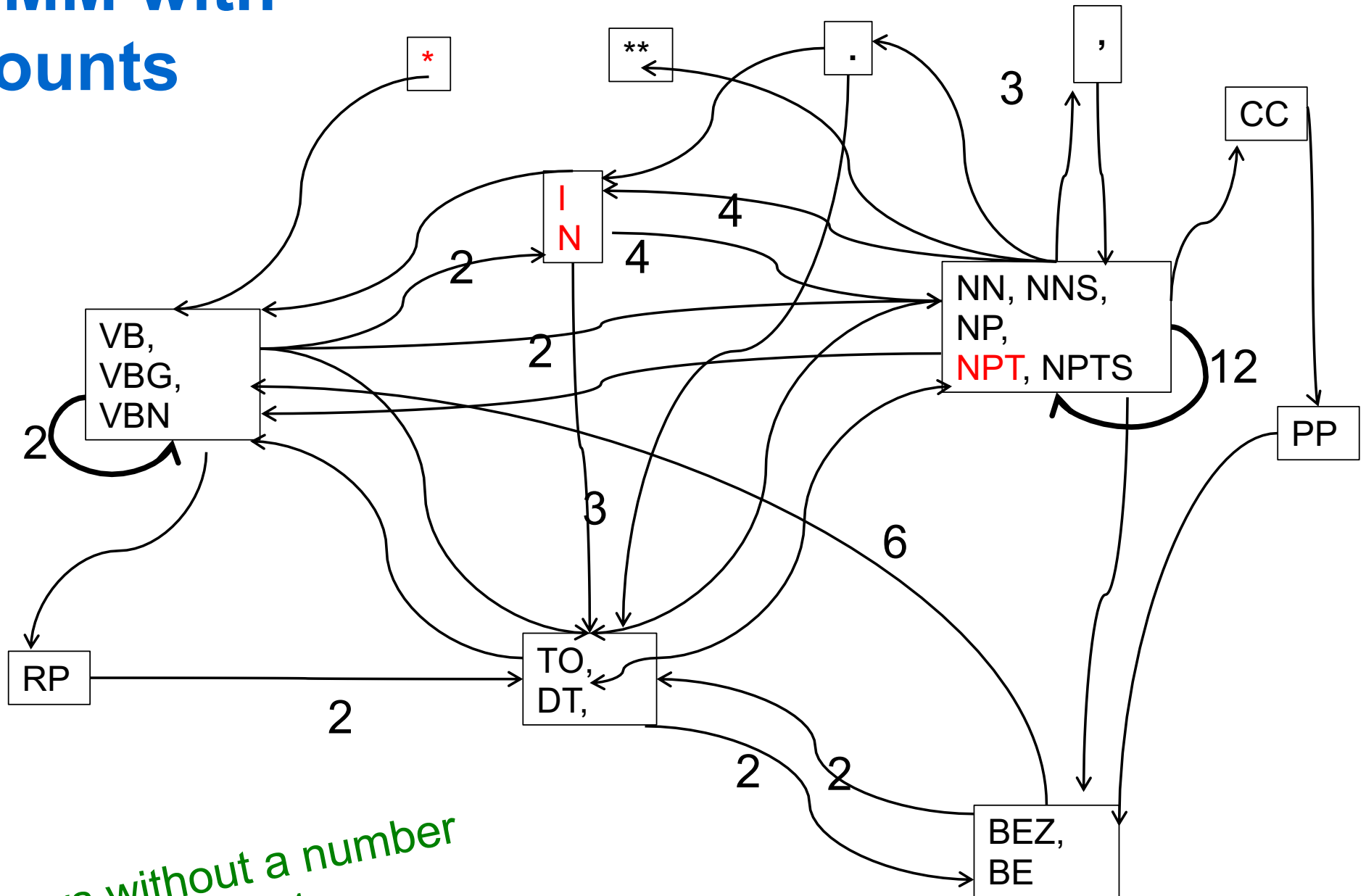
*by\_IN Trevor\_NP Williams\_NP*

*A\_DT move\_NN to\_TO stop\_VB Mr\_NPT Gaitskell\_NP from\_IN nominating\_VBG  
any\_DT more\_DT labour\_NN life\_NN peers\_NNS is\_BEZ to\_TO be\_BE made\_VBN  
at\_IN a\_DT meeting\_NN of\_IN labour\_NN MPs\_NPTS tomorrow\_NR .\_.*

*Mr\_NPT Michael\_NP Foot\_NP has\_VBN put\_VBN down\_RP a\_DT resolution\_NN  
on\_IN the\_DT subject\_NN and\_CC he\_PP is\_BEZ to\_TO be\_BE backed\_VBN by\_IN  
Mr\_NPT Will\_NP Griffiths\_NP ,\_, MP\_NPT for\_IN Manchester\_NP  
Exchange\_NP .\_.*

IN: preposition	R.*:adverb
DT: determiner	N.*:noun
CC: conjunction	V.*:verb

# HMM with counts



Arrows without a number  
have a count of 1

Note: The following slides use notation typically used for probabilities

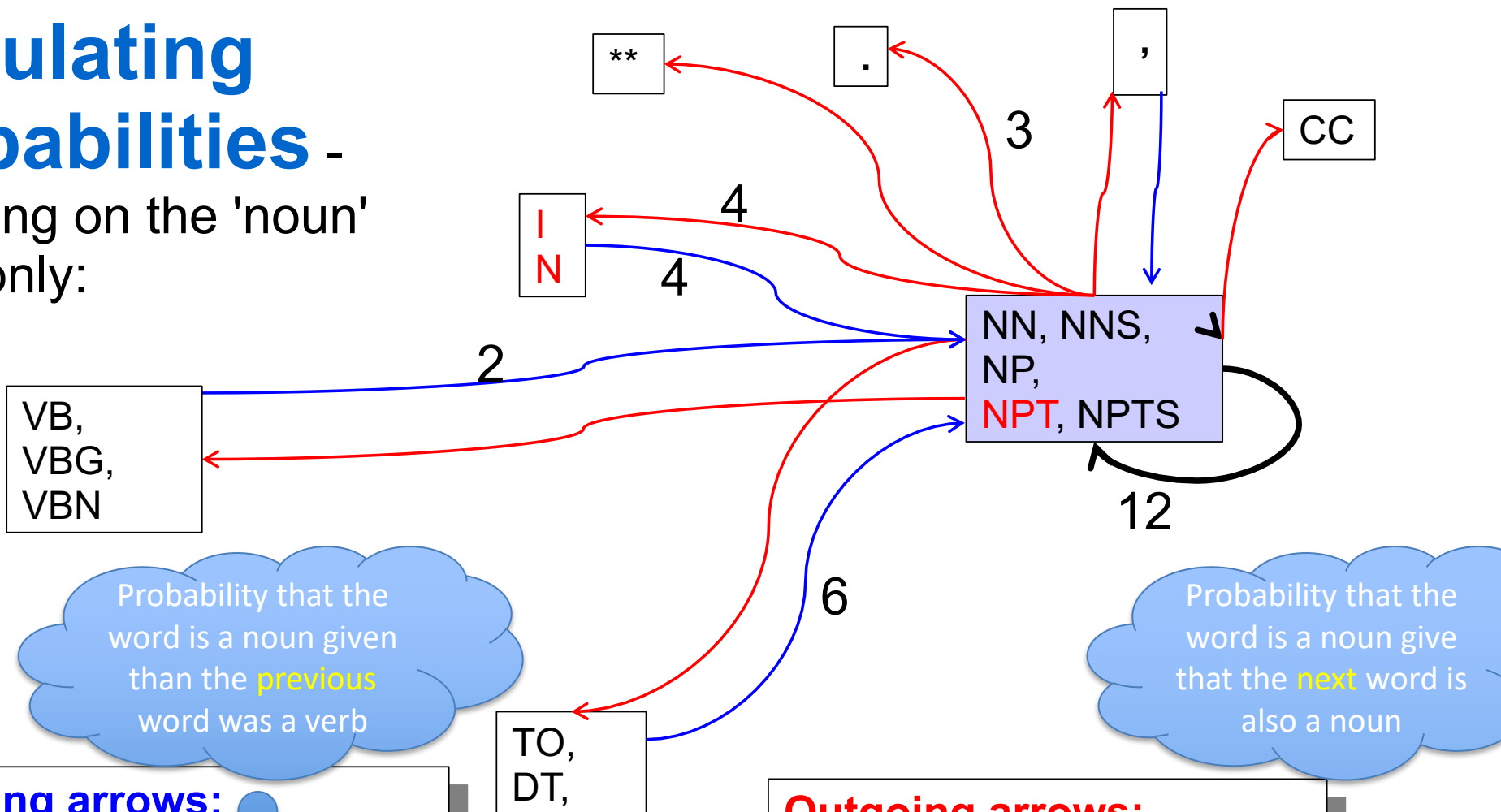
$P(a | b)$  is read as the probability of 'a' occurring given that 'b' has occurred, for example:

*$P(\text{it was raining} | \text{footpath is wet}) = 0.98$*  means there is a 98% chance that it was raining given that the footpath is wet

# HMM

## calculating probabilities -

Focusing on the 'noun' node only:



### Incoming arrows:

$$1+4+2+6+12=25$$

$$P(\text{Noun} \mid \text{Verb}) = 2/25$$

$$P(\text{Noun} \mid \text{IN}) = 4/25$$

$$P(\text{Noun} \mid \text{DT}) = 6/25$$

etc.

### Outgoing arrows:

$$1+1+3+1+4+1+1+12=24$$

$$P(\text{IN} \mid \text{Noun}) = 4/24$$

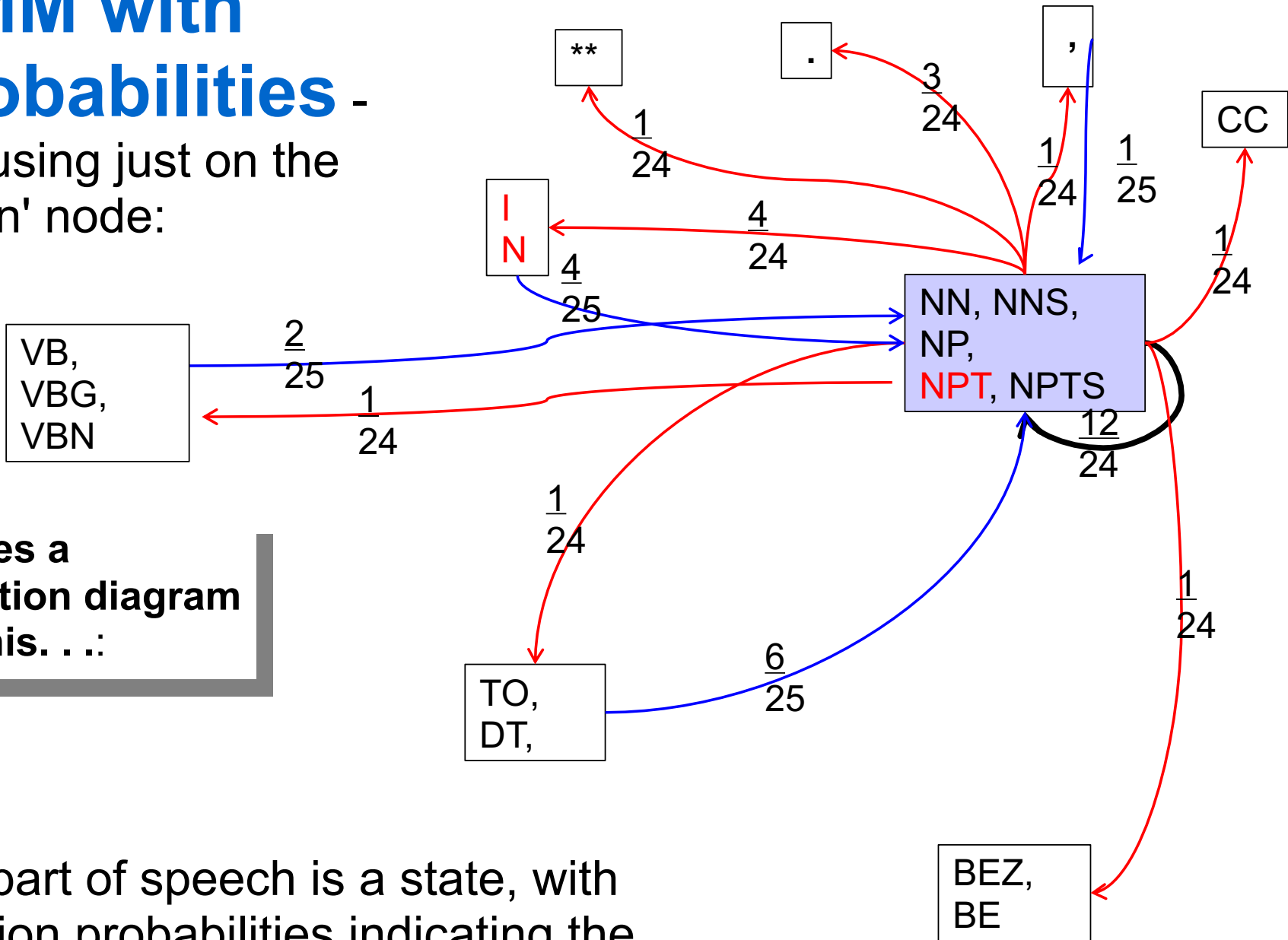
$$P(** \mid \text{noun}) = 1/24$$

$$P(\text{Noun} \mid \text{Noun}) = 12/24$$

etc.

# HMM with probabilities -

Focusing just on the 'noun' node:

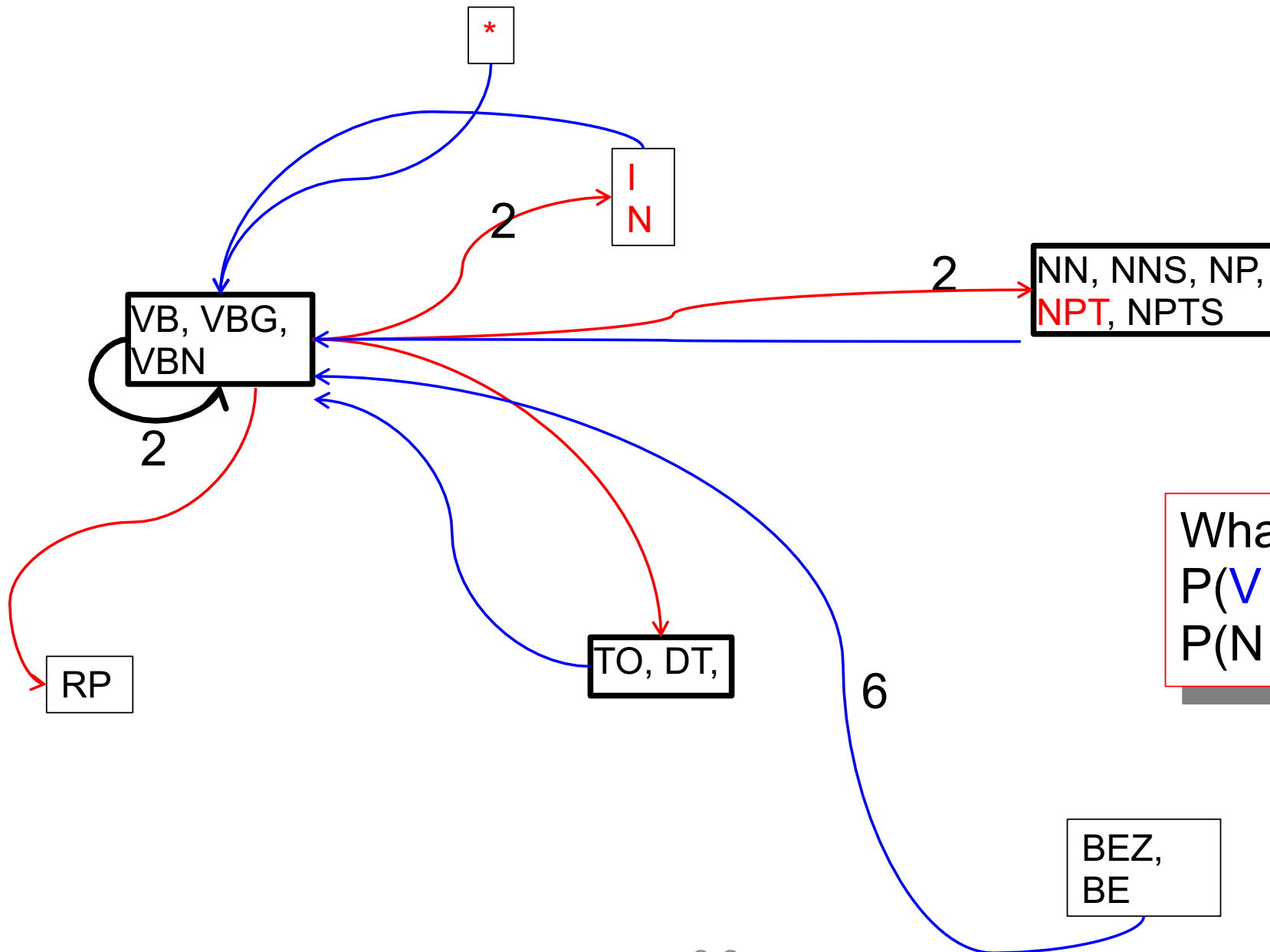


.. gives a transition diagram like this. . .:

Each part of speech is a state, with transition probabilities indicating the likelihood of one state following another.



**Exercise:** Calculate the probabilities for a verb if the next word is a noun; and the probability of a verb if the previous term was a determiner.



What is:  
 $P(V | DT)$   
 $P(N | V)$

# Hidden Markov model

So we've looked at the start state, and the transition diagram.  
The third part of a HMM is:

**3. The observation** which is the sentence you are attempting to tag:

e.g. Sonia is expected to race tomorrow

# Using a Hidden Markov Model

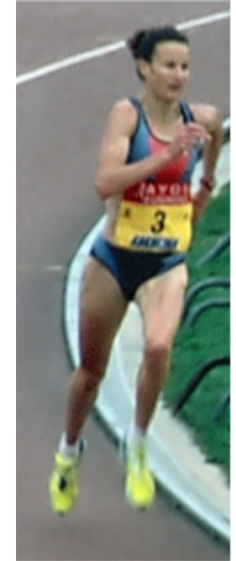
Multiply three figures:

1. Start state probability
2. Transition probability based on previous term(s)
3. Transition probability based on following term(s)

# Using a Hidden Markov model

Taking the following sentence (observation):

Sonia is expected to race tomorrow.



Words that have only one part of speech are tagged

Sonia(noun) is(verb) expected(verb) to(determiner) race(?)  
tomorrow(noun).

Race can be a noun (*when is the race?*) or a verb (*I will race to the shop*).

To determine which it is in this observation, we use a HMM to decide which is the most likely part of speech.

# Using a Hidden Markov model

## 1. We need to know the Start State:

The start state is the probability of a term having a particular part of speech.

- For most terms this is 1. (**Sonia** is a **noun** with probability of 1).
- For ambiguous words, such as **race**, the start state probabilities can be calculated from an existing tagged corpus/text.

Start state: Assume you are given a table of starting probabilities such as this, which includes the term 'race':

Terms:	Race	Light	...
Noun	0.05	0.30	
Verb	0.95	0.15	
Adjective	0	0.55	
Etc.	...	...	

Starting probabilities  
for 'Race'

Starting probabilities  
for 'light'

# Using a Hidden Markov model

Observation again: Sonia(noun) is(verb) expected(verb) to(determiner) race(?) tomorrow(noun).

2. We need to know the Transition probabilities: Transition probabilities for **race** are determined by looking at:

- one word before **race**, which is 'to', a determiner (DT)
- and one word after **race** which is 'tomorrow', a noun.

# Using a Hidden Markov model

First calculation: Probability that **race** is a **verb**:

## Start State:

- $P(\text{verb} | \text{race}) = 0.95$  The term **race** is a **verb** 95% of the time.

## Transitions (using the probabilities from our earlier slides):

- Looking back a term:  $P(\text{verb} | \text{DT}) = 1/12 = 0.083$ . There is an 12.5% chance that this term is a verb if the last term was a determiner.
- Looking forward a term:  $P(\text{noun} | \text{verb}) = 2/8 = 0.25$  There is a 8.3% chance that the next term is a noun if this term is a verb.

## Combine transition probabilities with start state probability:

- $*P(\text{verb} | \text{race}) * P(\text{verb} | \text{DT}) * P(\text{noun} | \text{verb}) = 0.95 * 0.083 * 0.25 = 0.02$ .

Combining the start state probability, and the two transition probabilities gives a probability of **0.02** that this is a verb.

# Using a Hidden Markov model

Second calculation: Probability that **race** is a **noun**:

## Start State:

- $P(\text{noun} | \text{race}) = 0.05$  The term **race** is a **noun** 5% of the time.

## Transitions (using the probabilities from our earlier slides)

- Looking back:  $P(\text{noun} | \text{DT}) = 6/25 = 0.24$  There is an 24% chance that if the last term was a preposition, the next term will be a noun.
- Looking forward:  $P(\text{noun} | \text{noun}) = 12/24 = 0.5$ . There is an 50% chance that if the next term is a noun, this term is a noun.

## Combine transition probabilities with start state probability:

- $P(\text{noun} | \text{race}) * P(\text{noun} | \text{DT}) * P(\text{noun} | \text{noun}) = 0.05 * 0.24 * 0.5 = 0.006$ .

Combining the start state probability, and the two transition probabilities give a probability of **0.006** that this is a noun.



# Using Hidden Markov model

Points to Note:

- As 0.02 is higher than 0.006, the HMM (based on the probabilities used in the last slide) correctly predicted that, in this sentence, race was a **verb**.
- *Note the importance of the start state probabilities, and not just the transition probabilities.*

This is a first order Markov model, because it looked at just one term on either side.

- A second order Markov model would look at 2 terms at each side.
- An  $m$ -order Markov model looks at  $m$  terms on each side.
- The higher the order, the more accurate the model, but at the cost of additional complexity & computation cost.
- First order Markov models are the most common.

# Sample exam question

Given the following hidden Markov model, and sample text, explain how to use the hidden Markov model to determine the correct part of speech:

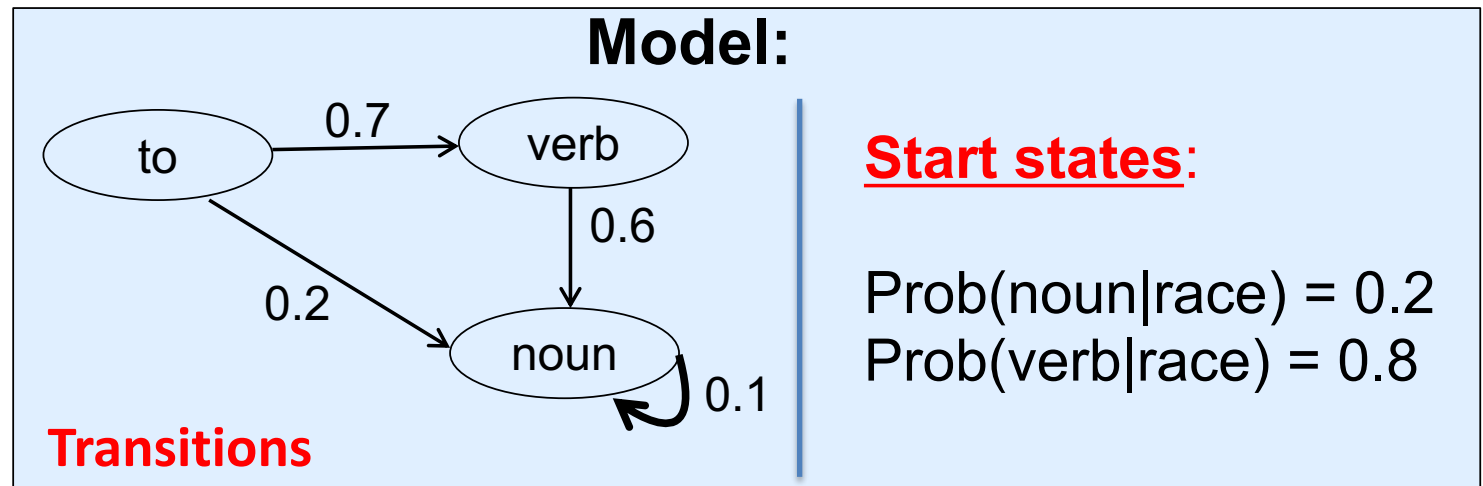
**“I had to race home”**

Relevant Parts of Speech:

to = to

race = noun or verb

home = noun



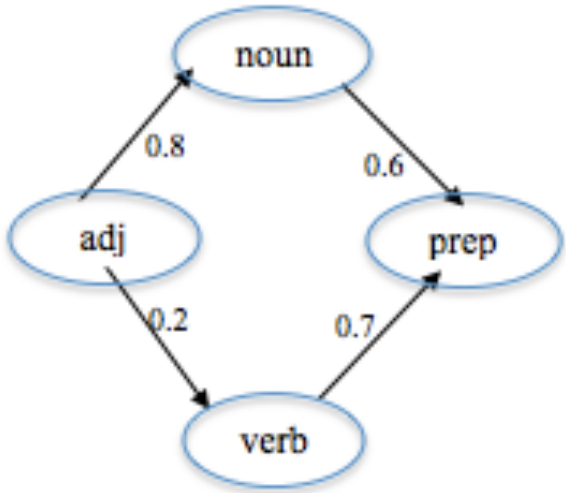
## Exercise: question from 2012 exam paper

A) Explain the components of a Hidden Markov Model and their relevance to text mining.

**(8 marks)**

B) Using the sample sentence and Hidden Markov Model given below, illustrate how to determine if 'walk' is a noun or a verb.

**(6 marks)**

Sample sentence	Model	
<p><b>He went for a long <u>walk</u> on the beech.</b></p> <p>Relevant Parts of Speech: long = <u>adj</u> walk = noun or verb on = preposition (prep)</p>	 <pre>graph TD; noun((noun)) -- 0.8 --&gt; adj((adj)); noun -- 0.6 --&gt; prep((prep)); adj -- 0.2 --&gt; verb((verb)); verb -- 0.7 --&gt; prep</pre>	<p><u>Prob(noun walk) = 0.4</u> <u>Prob(verb walk) = 0.6</u></p>

## Marking scheme for exercise:

A)

Role of Markov model: used to distinguish between different meanings of the same term - homonyms. **(1 mark)**

Three components:

Start state – probability of each meaning of the term occurring, without looking at context, calculated from a tagged corpus: **(3 marks)**

Transitions – probability of each meaning of the term based on terms before and after it, calculated from a tagged corpus: **(3 marks)**

Observation – sentence under review **(1 mark)**

B)

Probability it is a noun:  $0.4 * 0.8 * 0.6$  **(2 marks)**

Probability it is a verb:  $0.6 * 0.2 * 0.7$  **(2 marks)**

As 1<sup>st</sup> calculation is higher, in this sentence walk is a noun. **(2 marks)**

# POS taggers

The Hidden Markov Model is one of a number of approaches to POS tagging. Other approaches have been developed based on a range of classification algorithms including Decision Trees, Neural Networks, Support Vector Machines (SVM's), and rules based systems.

*Implementations:*

POS taggers: <http://www-nlp.stanford.edu/links/statnlp.html#Taggers>

# Summary

. . . . see mindmap on Moodle summarising lecture 2.1 and 2.2

