Hons. Degree in Computing
H4016 Text Mining & Information Retrieval

Unit 2 – Preparing text data for data mining

Part II: polysemy and markov model

G.Gray

Recap on last week:
Text Mining Process

5. Analyzing Results

4. Text/Data Mining

-Classification- Supervised Learning

-Clustering- Unsupervised Learning

3. Feature Selection

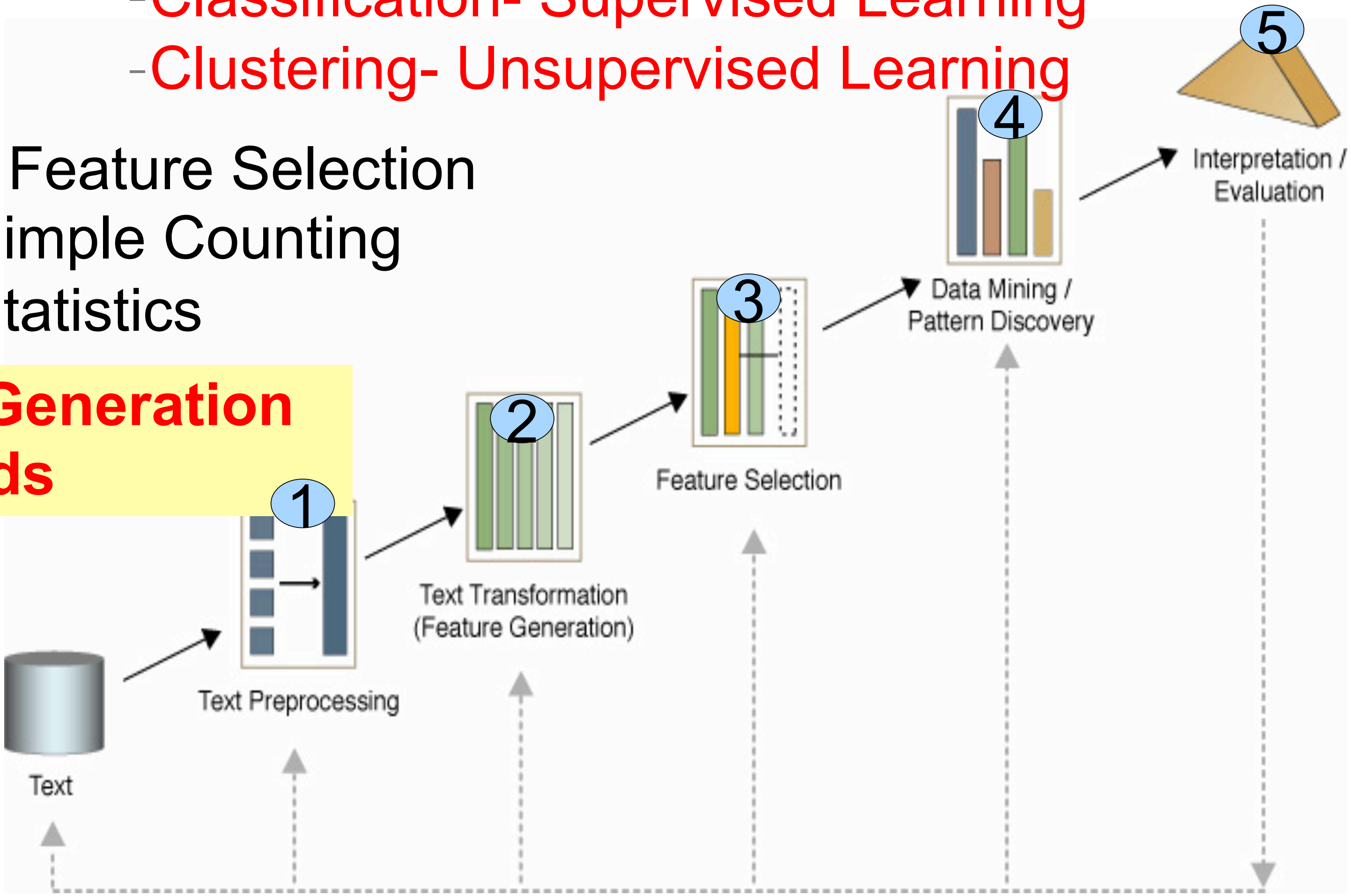
-Simple Counting

-Statistics

2. **Features Generation**

-**Bag of Words**

1. Collect text and identify business objective


5
Interpretation / Evaluation
4
Data Mining / Pattern Discovery
3
Feature Selection
2
Text Transformation (Feature Generation)
1
Text Preprocessing
Text

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 – is the comment a complaint, a request, . .

5) Link words with the same canonical form –' I am', 'you are', 'he is' are all from the verb 'be

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

5. Polysemy

Polysemy is a single word with different meanings

Words have, on average, 1.4 meanings

Verbs are worst, with on average, 2.1 meanings

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.

This next 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 mining results.

Word Sense Disambiguation (WSD)

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

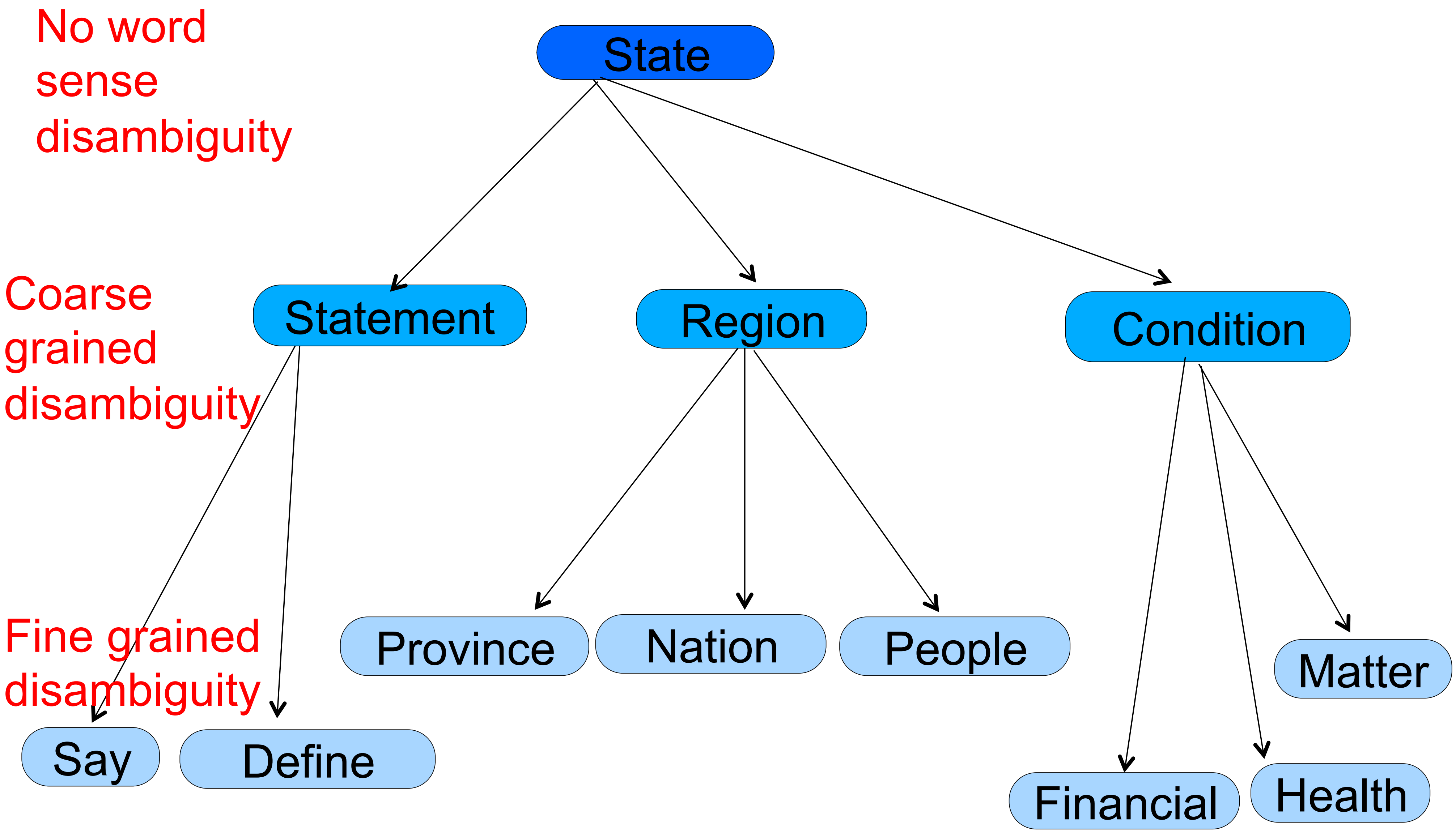
He was is a bad state Health

The company is in a weak financial state Financial

Some word **senses** are closer in meaning than others.

Differences in **senses** can be coarse grained or fine grained.

Word Sense Disambiguation (WSD)



No word sense disambiguity

State

Coarse grained disambiguity

Statement

Region

Condition

Fine grained disambiguity

Say

Define

Province

Nation

People

Matter

Financial

Health

Word Sense Disambiguation (WSD) / parts of speech 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 fist 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

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.

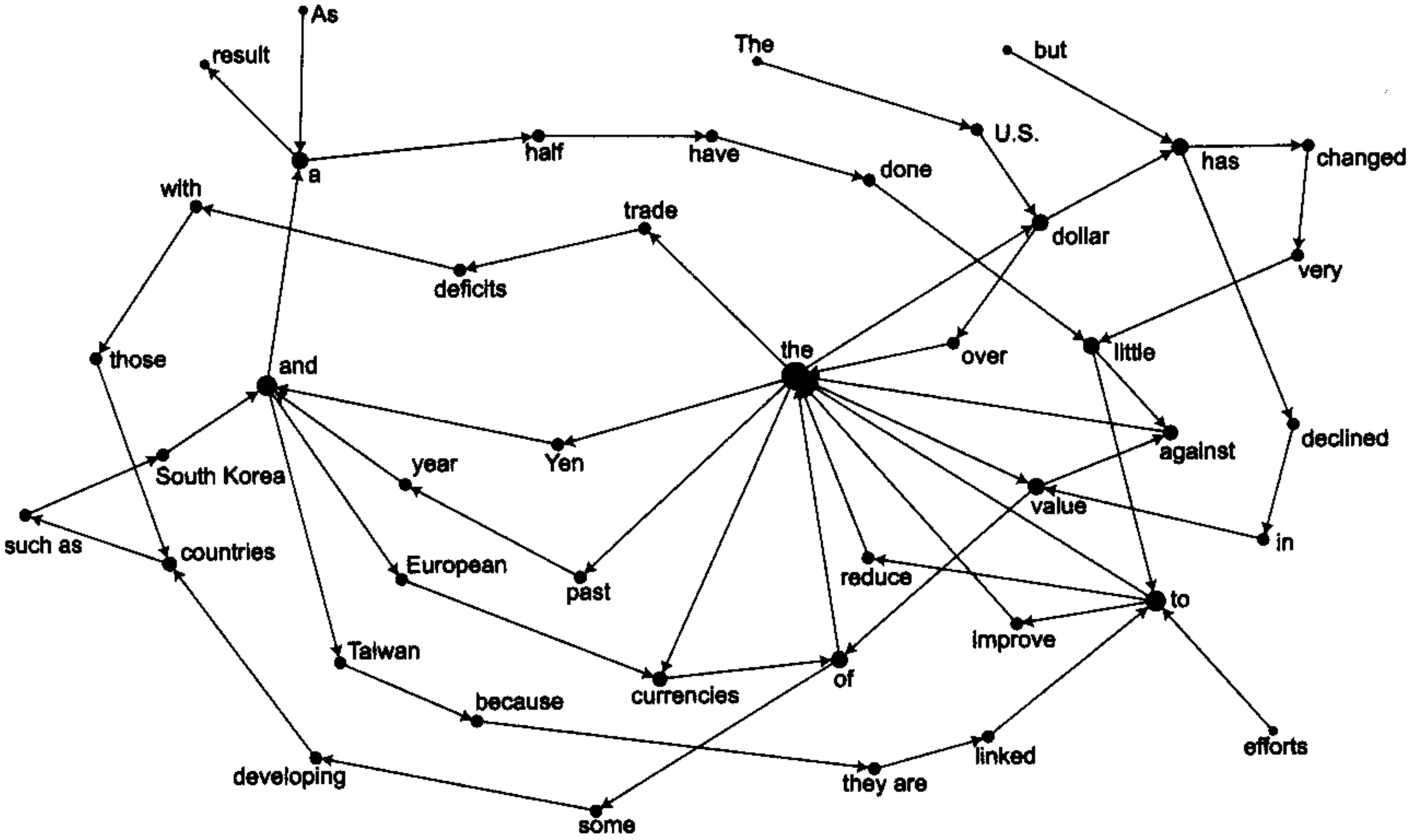


FIGURE 4.2 Network of word sequences from two sentences.

from Konchady, Text Mining Application Programming

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

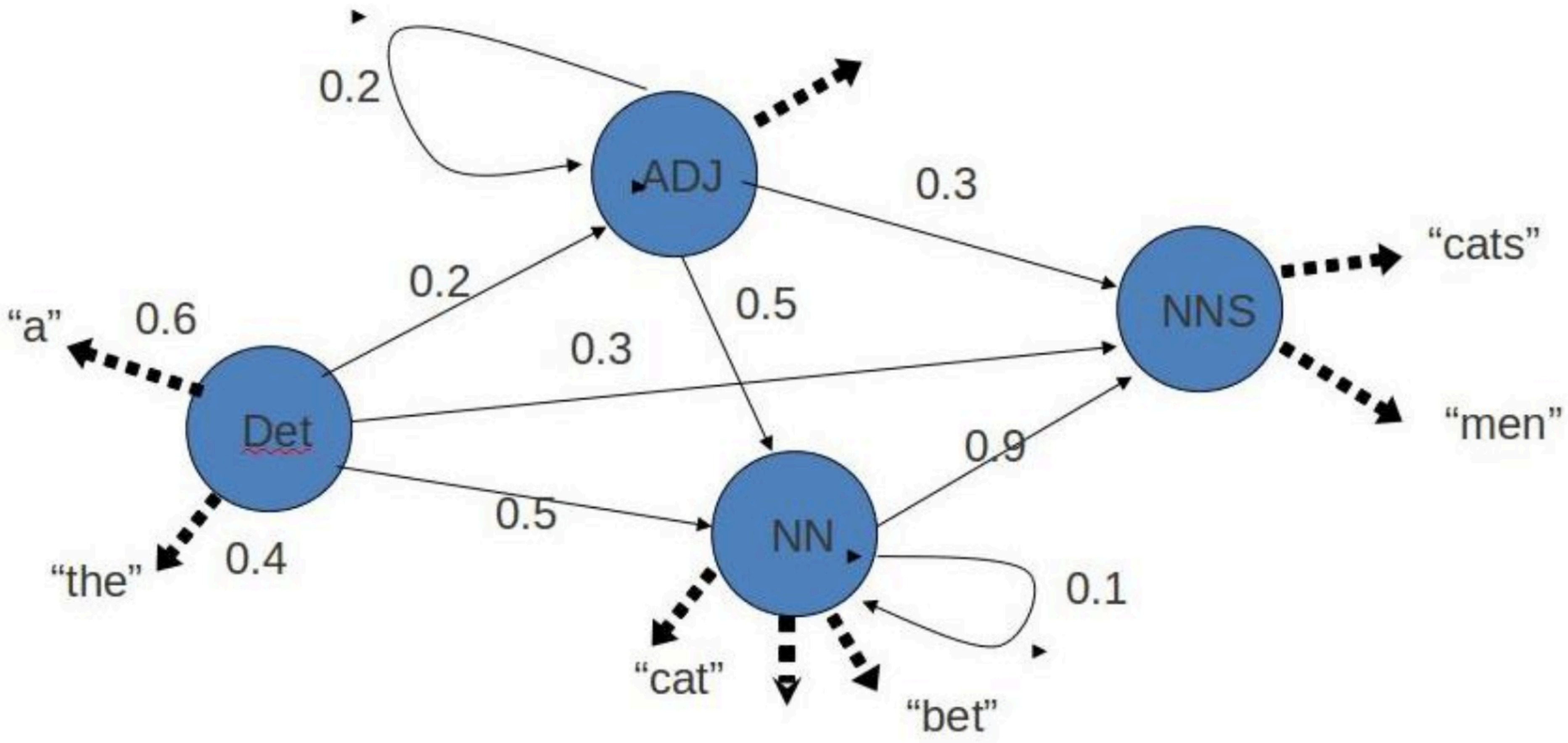
Slide 11 is a Markov model, where you know in advance what state (node / word) you are at.

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 Hidden Markov model for POS tagging



Hidden Markov model

You 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 is ’coin' refering 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 other parts of speech i.e. 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 dataset, calculate how common each state of a term is, given a table of probabilities as follows:

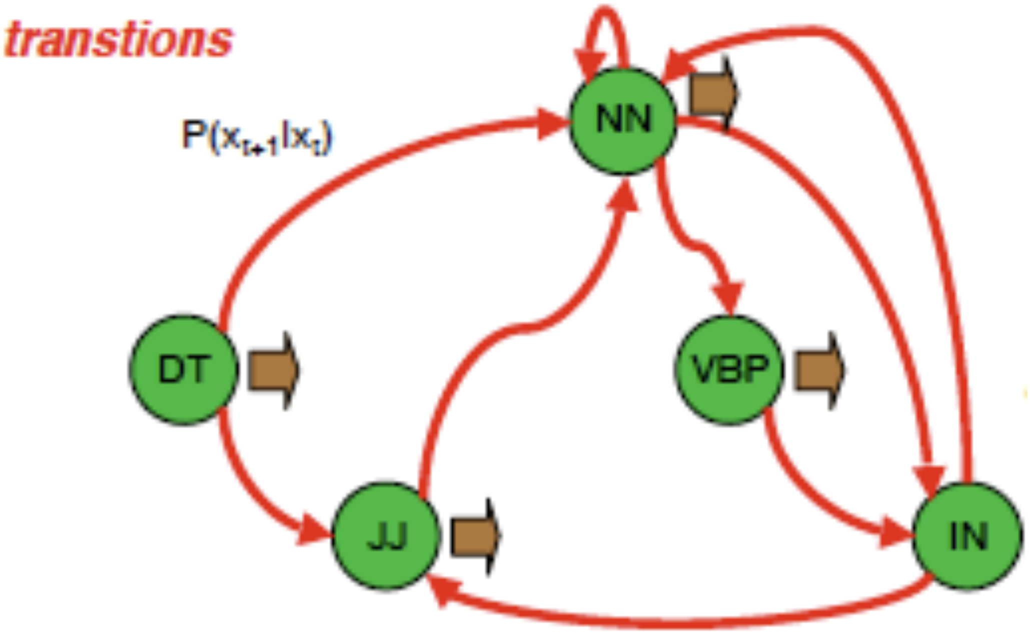
Term	State	Frequency
Coin	Noun	0.95 (95%)
Coin	Verb	0.05 (5%)

Hidden Markov model

2. Transitions:

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

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 gave a small number of states, e.g.

- 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.

See PartsOfSpeech.pdf for examples

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 ._.

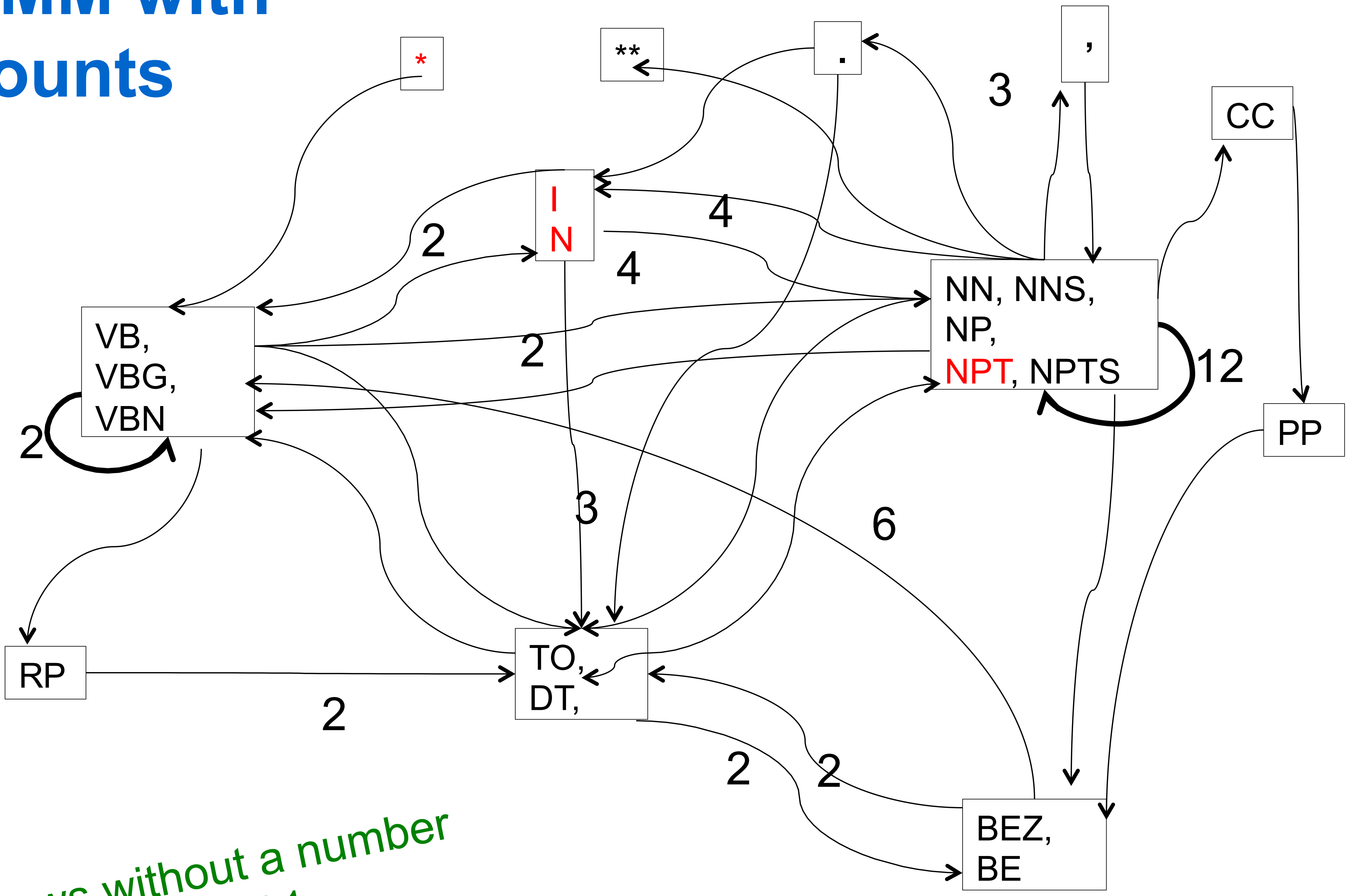
IN: preposition
DT: determiner
CC: conjunction

R.*:adverb
N.*:noun
V.*:verb

Adapted from http://icame.uib.no/lob-eks.html

G.Gray

HMM with counts



Arrows without a number have a count of 1

G.Gray

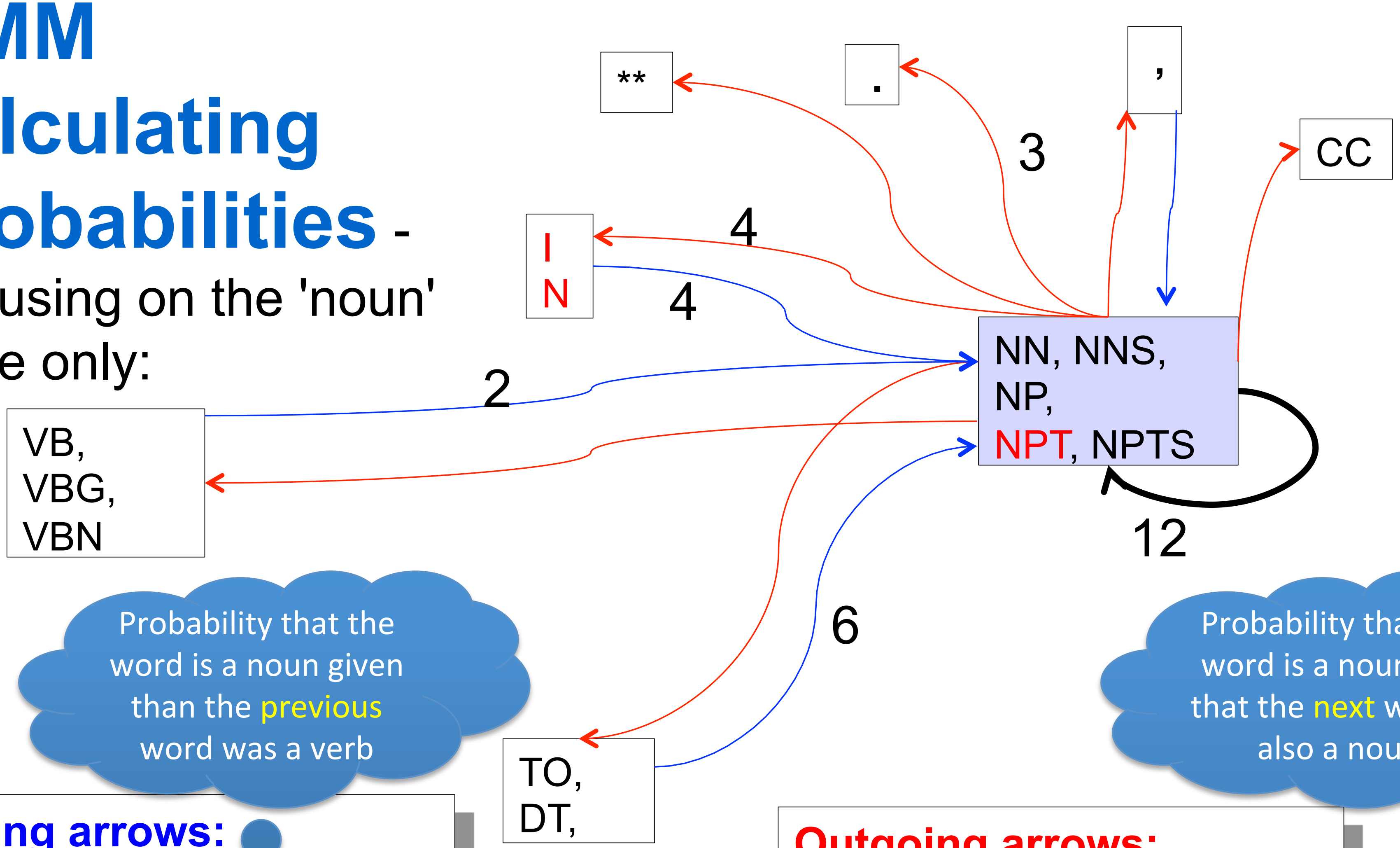
Note: The following slides use notation typically used for probabilities

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

P(it was raining | 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:



Probability that the word is a noun given than the previous word was a verb

Probability that the word is a noun give that the next word is also a noun

Incoming arrows:

1+4+2+6+12=25

P(Noun | Verb) = 2/25

P(Noun | IN) = 4/25

P(Noun | DT) = 6/25

etc.

Outgoing arrows:

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

P(IN | Noun) = 4/24

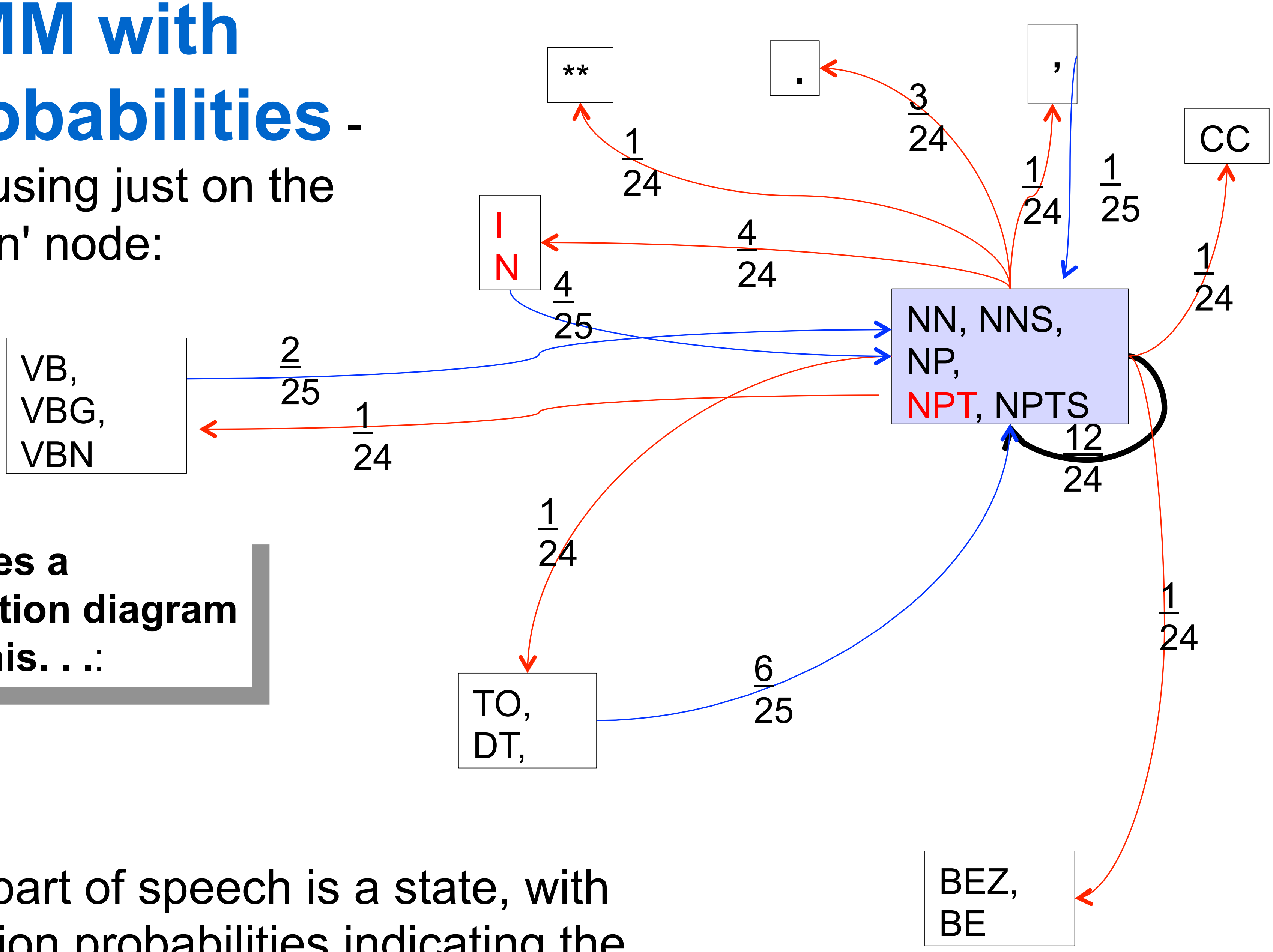
P(** | noun) = 1/24

P(Noun | Noun) = 12/24

etc.

HMM with probabilities -

Focusing just on the 'noun' node:

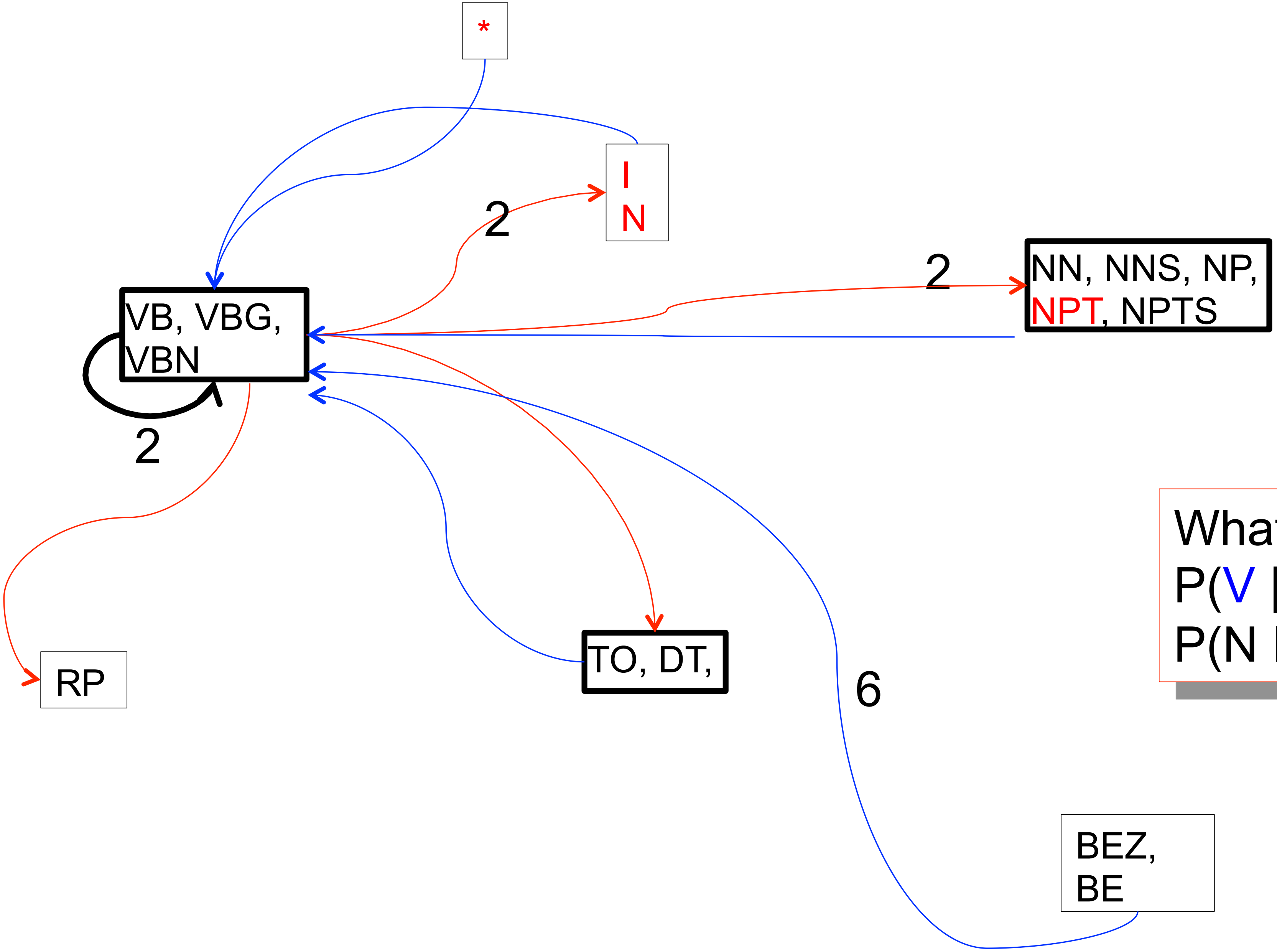


. . gives a transition diagram like this. . .:

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

G.Gray

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



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

G.Gray

Hidden Markov model

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

e.g. Sonia is expected to race tomorrow

Using a Hidden Markov Model

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. 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.

Example:

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. Transition: Transition probabilites 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

Probability that race is a **verb**:

Transitions (using the probabilities from our earlier slides):

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

State:

- P(verb|race) = 0.95 The term race is a verb 95% of the time.

Combine transition probabilities with state probability:

- P(verb|DT)*P(noun|verb)*P(verb|race) = 0.083*0.25*0.95=0.02.

Combining the state sequence probabilities, and the observation probability gives a probability of 0.02 that this is a verb.

Using a Hidden Markov model

Probability that race is a **noun**:

Transitions (using the probabilities from our earlier slides)

- Looking back: P(noun|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(noun|noun) = 12/24 = 0.5. There is an 50% chance that if the next term is a noun, this term is a noun.

State:

- P(noun|race) = 0.05 The term race is a noun 5% of the time.

Combine transition probabilities with state probability:

- P(noun|DT)*P(noun|noun)*P(noun|race)= 0.24*0.5*0.05=0.006. Combining the state sequence probabilities, and the observation probability 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 observation probabilities, and not just the transition probabilities.*

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

- A second order Markov model would look at 2 words at each side. The state sequence would then be longer requiring additional computations.
- An *m*-order Markov model looks at *m* words 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.

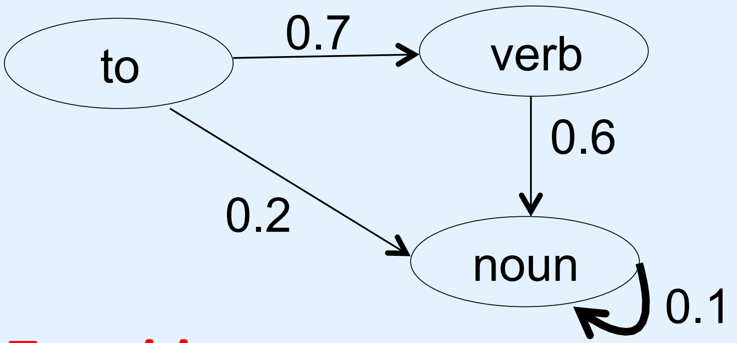
Sample exam question

Given the following hidden markov mode, and sample text, explain how to use the 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

Model:



Transitions

Start states:

Prob(noun|race) = 0.2
Prob(verb|race) = 0.8

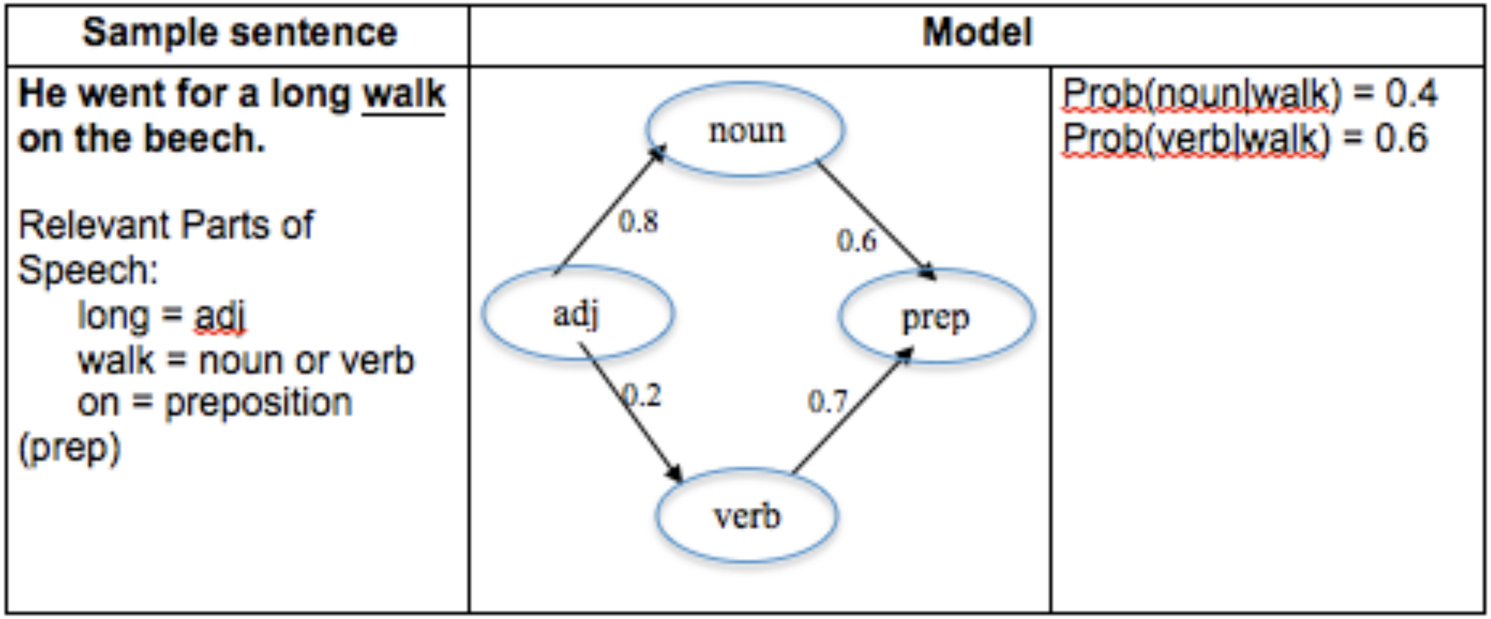
Exercise: question from 2012 exam paper

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	

He went for a long walk on the beech.
Relevant Parts of Speech:
long = adj
walk = noun or verb
on = preposition (prep)
		Prob(noun|walk) = 0.4
		Prob(verb|walk) = 0.6

noun
adj
prep
verb
0.8
0.6
0.2
0.7

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 1st calculation is higher, in this sentence walk is a noun. **(2 marks)**

POS taggers

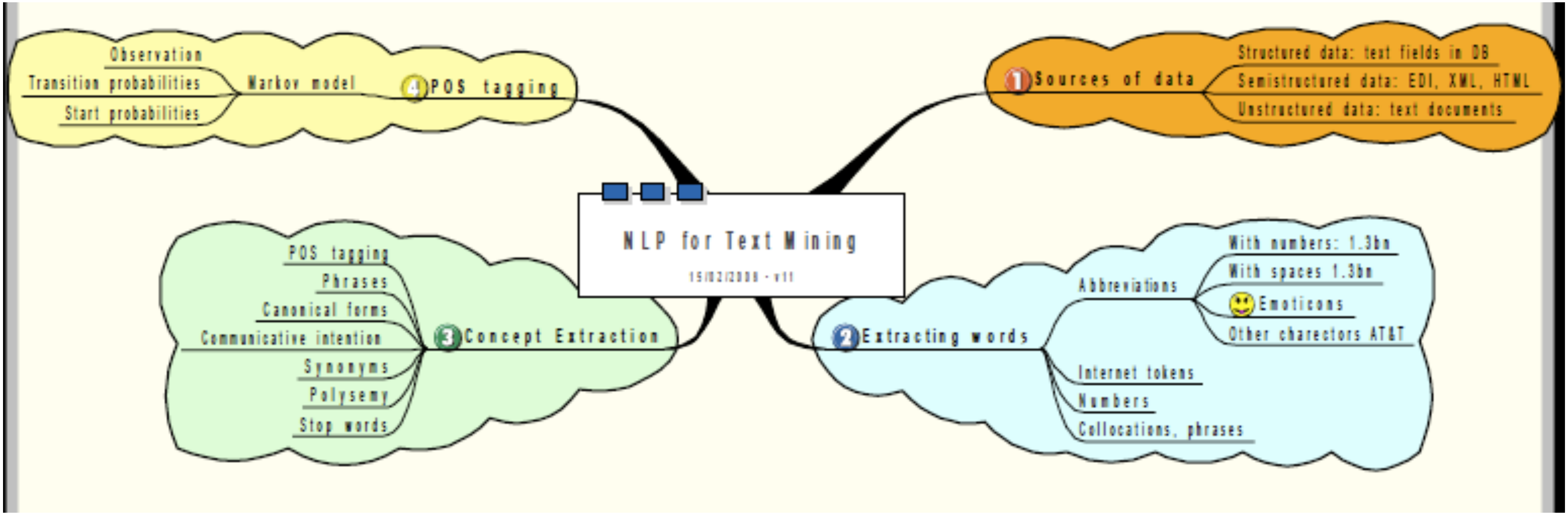
The Hidden Markov Model is one of a number of approaches to POS tagging. Other approaches have been developed based on a range of classification algorithms including Decision Trees, Neural Networks, Support Vector Machines (SVM's), and rules based systems.

- *Implementations:*

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

Summary

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



G.Gray