COMP90042
Web search and text analysis

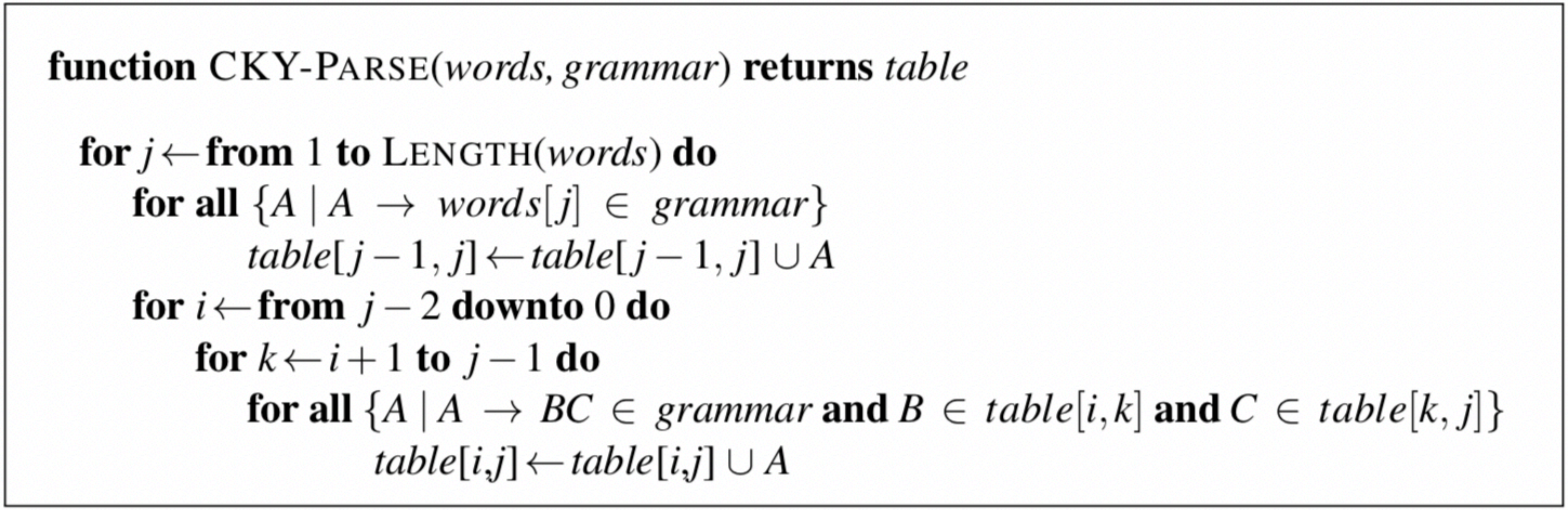
Workshop Week 11

xudong.han@unimelb.edu.au
https://github.com/HanXudong/COMP90042_Workshops

Review: CYK parsing

0 1 2 3 4

a	man	saw	John
[0,1]	[0,2]	[0,3]	[0,4]
	[1,2]	[1,3]	[1,4]
		[2,3]	[2,4]
			[3,4]



```
function CKY-PARSE(words, grammar) returns table

  for $j \leftarrow$ from 1 to LENGTH(words) do
      for all $\{A \mid A \rightarrow words[j] \in grammar\}$
          $table[j-1,j] \leftarrow table[j-1,j] \cup A$
      for $i \leftarrow$ from $j-2$ downto 0 do
          for $k \leftarrow i+1$ to $j-1$ do
              for all $\{A \mid A \rightarrow BC \in grammar$ and $B \in table[i,k]$ and $C \in table[k,j]\}$
                  $table[i,j] \leftarrow table[i,j] \cup A$
```

Figure 12.5 The CKY algorithm.

Q1
Construct a dependency parse for the following sentence:

Yesterday, I shot an elephant in my pyjamas.

- maintain two data structures
  - buffer: input words yet to be processed
  - stack: head words currently being processed

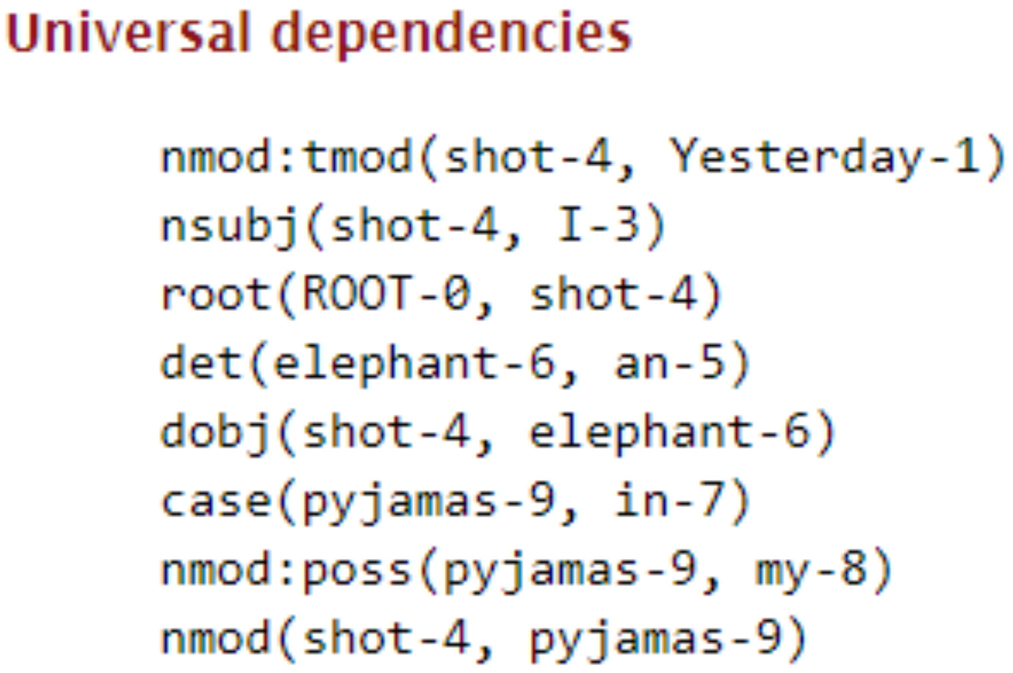
- two types of transitions
  - shift = move word from buffer on to top of stack
  - arc = add arc (left/ right) between top two items on stack and remove dependent from stack

Relations

Relations	Description	Examples
NSUBJ	Nominal subject	**United** canceled the flight.
DOBJ	Direct object	United diverted the **flight** to Reno.
IOBJ	Indirect object	We booked **her** the flight to Miami.
CCOMP	Clausal complement	
XCOMP	Open clausal complement	
NMOD	Nominal modifier	We took the **morning** flight.
AMOD	Adjectival modifier	Book the **cheapest** flight.
NUMMOD	Numeric modifier	Before the storm JetBlue canceled **1000** flights.
APPOS	Appositional modifier	United, a **unit** of UAL, matched the fares.
DET	Determiner	**The** flight was canceled.
CASE	Prepositions, postpositions and other case markers	Book the flight **through** Houston.
CONJ	Conjunct	We flew to Denver and **drove** to Steamboat.
CC	Coordinating conjunction	We flew to Denver **and** drove to Steamboat.

- two types of transitions
  - shift = move word from buffer on to top of stack
  - arc = add arc (left/ right) between top two items on stack and remove dependent from stack

Buffer	Stack	Action
Yesterday, I shot an elephant in my pyjamas.		Shift
I shot an elephant in my pyjamas	Yesterday	Shift
shot an elephant in my pyjamas	Yesterday, I	Shift
an elephant in my pyjamas	Yesterday, I, shot	Arc-Left (I <- shot)
an elephant in my pyjamas	I, shot	Arc-Left (Yesterday <- shot)
an elephant in my pyjamas	shot	



Universal dependencies

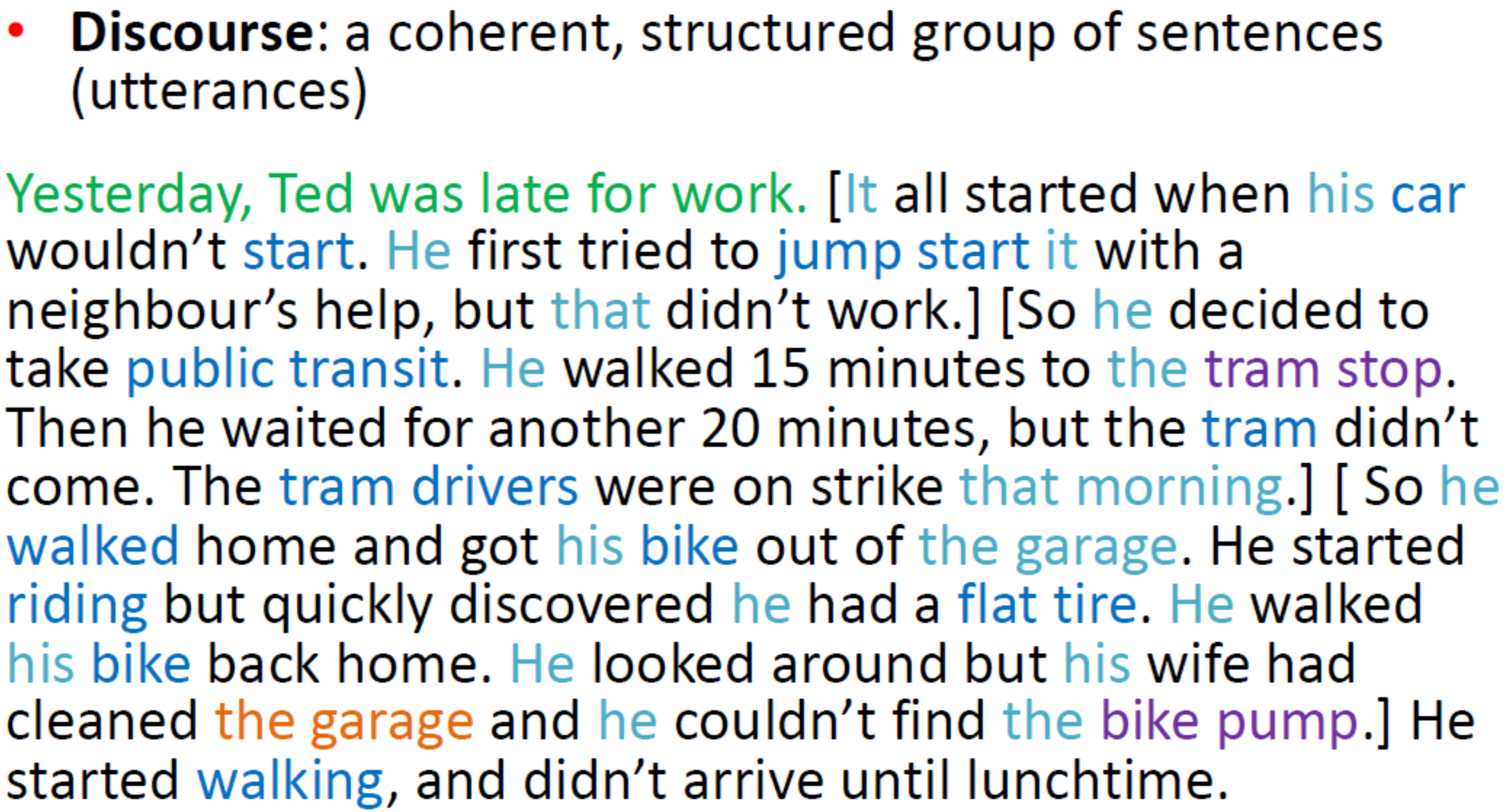
```
nmod:tmod(shot-4, Yesterday-1)
nsubj(shot-4, I-3)
root(ROOT-0, shot-4)
det(elephant-6, an-5)
dobj(shot-4, elephant-6)
case(pyjamas-9, in-7)
nmod:poss(pyjamas-9, my-8)
nmod(shot-4, pyjamas-9)
```

http://nlp.stanford.edu:8080/parser/index.jsp

Q2
In what ways is dependency parsing similar to CYK parsing? In what ways is it different?

- Connections:
  - Both methods are attempting to determine the structure of a sentence.
  - Both methods process the tokens in sentence one-by-one, left-to-right.
- Differences:
  - dependency parser doesn’t explicitly tag the sentence.
  - the transition-based Dependency parsing can take into account other non-local relations in the sentence, whereas CYK’s depend only on local sub-tree
  - CYK and numerous fragments to chart, which will not be used in final structures, whereas the transition-based dependency parser only adds edges that will be in the final structure.

Q3
What is Discourse Segmentation?



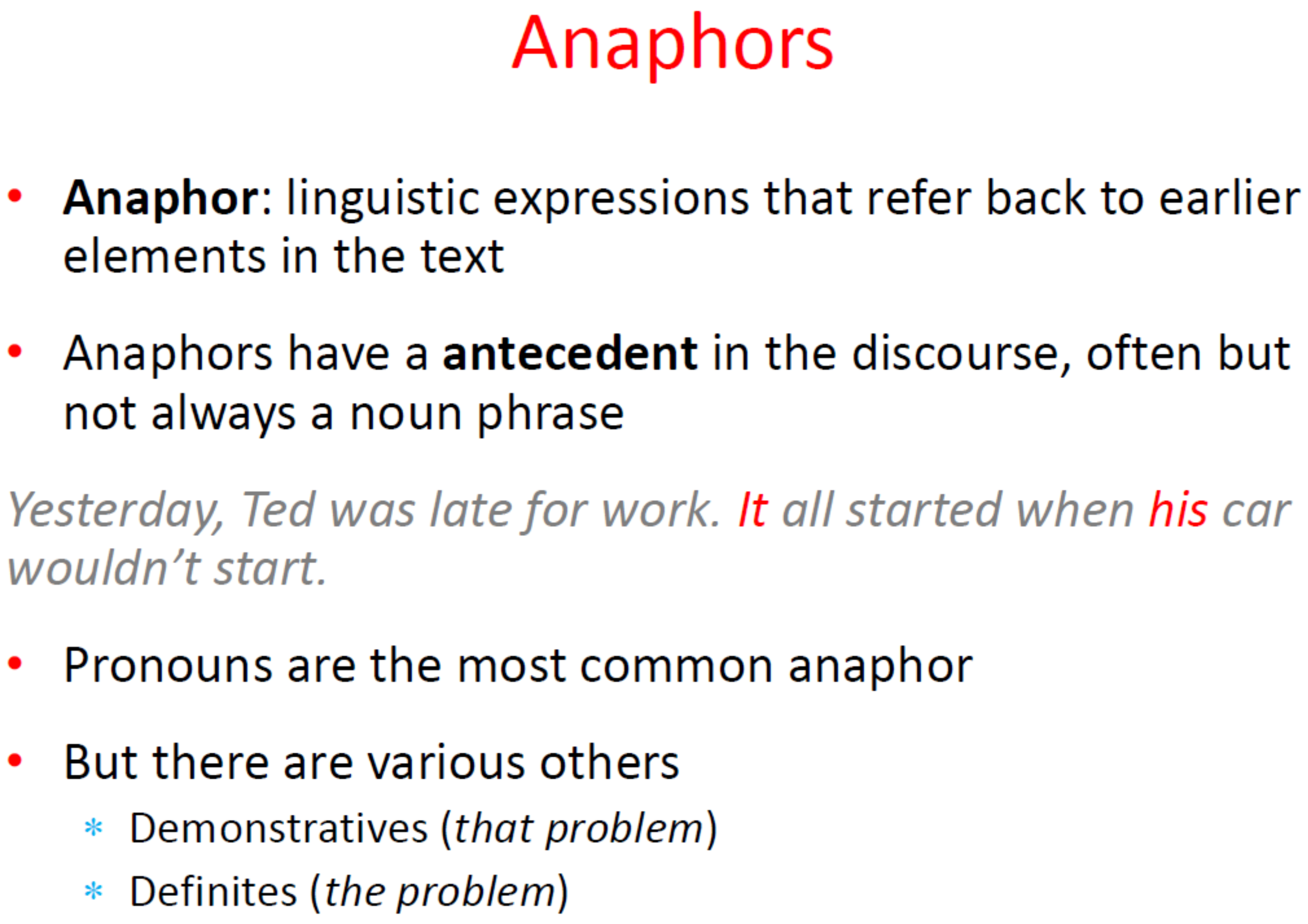
- **Discourse**: a coherent, structured group of sentences (utterances)

Yesterday, Ted was late for work. [It all started when his car wouldn’t start. He first tried to jump start it with a neighbour’s help, but that didn’t work.] [So he decided to take public transit. He walked 15 minutes to the tram stop. Then he waited for another 20 minutes, but the tram didn’t come. The tram drivers were on strike that morning.] [So he walked home and got his bike out of the garage. He started riding but quickly discovered he had a flat tire. He walked his bike back home. He looked around but his wife had cleaned the garage and he couldn’t find the bike pump.] He started walking, and didn’t arrive until lunchtime.

Q3
What do the segments consist of, and what are some methods we can use to find them?

- Discourse segmentation
    - Assumption: text can be divided into a number of discrete contiguous sections.
    - Task: classifying whether a boundary exists between any two sentences.
- Methods
    - Unsupervised approach: based on threshold / K
    - Supervised approach: features
    - Rule-based approach: using discourse markers

Q4 What is an anaphor



Anaphors

- **Anaphor**: linguistic expressions that refer back to earlier elements in the text
- Anaphors have a **antecedent** in the discourse, often but not always a noun phrase

Yesterday, Ted was late for work. It all started when his car wouldn't start.

- Pronouns are the most common anaphor
- But there are various others
  - Demonstratives (*that problem*)
  - Definites (*the problem*)

Q4-b What are some useful heuristics (or features) to help resolve anaphora?

- The most obvious but inherent unreliable heuristic is the recreancy heuristic: given multiple possible referents, the mostly intended one is the one most recently used in the text.

- A better heuristic is that the most likely referent(consistent in meaning with the anaphor) is the focus of the discourse.

- We can also build a supervised machine learning model, usually based around the semantic properties of the anaphor words and the sentence structure.