SENTIMENT ANALYSIS FUNDAMENTALS

FEATURE ENGINEERING FOR SENTIMENT ANALYSIS

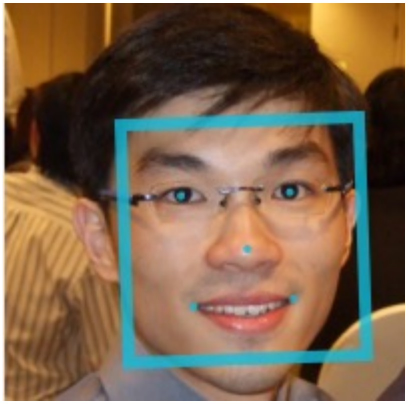
DR LEK HSIANG HUI

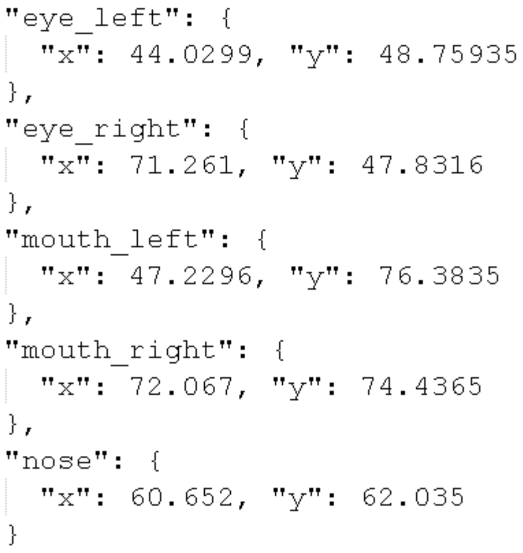
Features

- **Feature** (aka predictors, attributes, etc) = important piece of information that models the data
- **Feature Engineering** = Designing a feature set that is useful for a particular domain
  - Need to convert the data into a “simplified” format that represents the data
  - This requires understanding of the problem domain

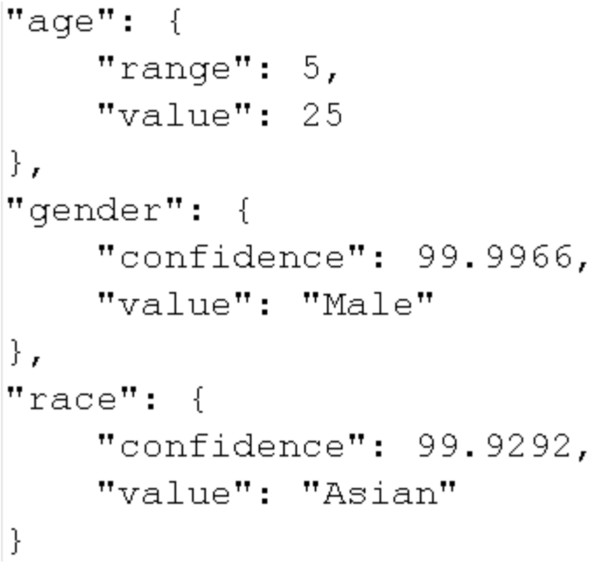
Features (Non-Textual)

- Facial Analytics
  - position of left eye, position of right eye, position of the left/right side of mouth, shape of the nose, etc



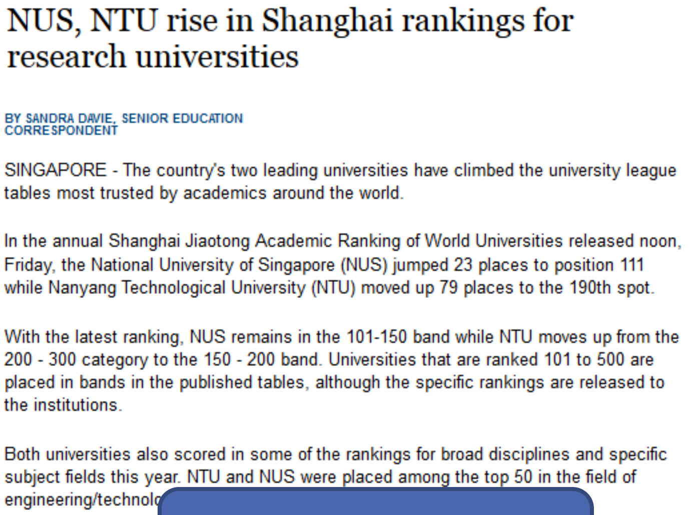


```
"eye_left": {
  "x": 44.0299, "y": 48.75935
},
"eye_right": {
  "x": 71.261, "y": 47.8316
},
"mouth_left": {
  "x": 47.2296, "y": 76.3835
},
"mouth_right": {
  "x": 72.067, "y": 74.4365
},
"nose": {
  "x": 60.652, "y": 62.035
}
```



```
"age": {
     "range": 5,
     "value": 25
},
"gender": {
     "confidence": 99.9966,
     "value": "Male"
},
"race": {
     "confidence": 99.9292,
     "value": "Asian"
}
```

Handling Text



NUS, NTU rise in Shanghai rankings for research universities

BY SANDRA DAVIE, SENIOR EDUCATION CORRESPONDENT

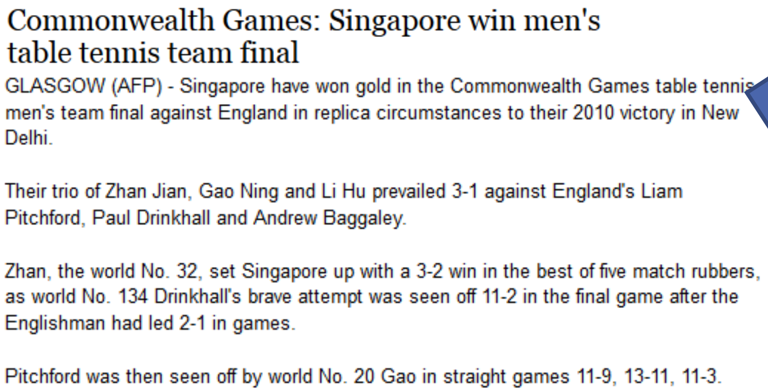
SINGAPORE - The country's two leading universities have climbed the university league tables most trusted by academics around the world.

In the annual Shanghai Jiaotong Academic Ranking of World Universities released noon, Friday, the National University of Singapore (NUS) jumped 23 places to position 111 while Nanyang Technological University (NTU) moved up 79 places to the 190th spot.

With the latest ranking, NUS remains in the 101-150 band while NTU moves up from the 200 - 300 category to the 150 - 200 band. Universities that are ranked 101 to 500 are placed in bands in the published tables, although the specific rankings are released to the institutions.

Both universities also scored in some of the rankings for broad disciplines and specific subject fields this year. NTU and NUS were placed among the top 50 in the field of engineering/technolo

Documents



Commonwealth Games: Singapore win men's table tennis team final

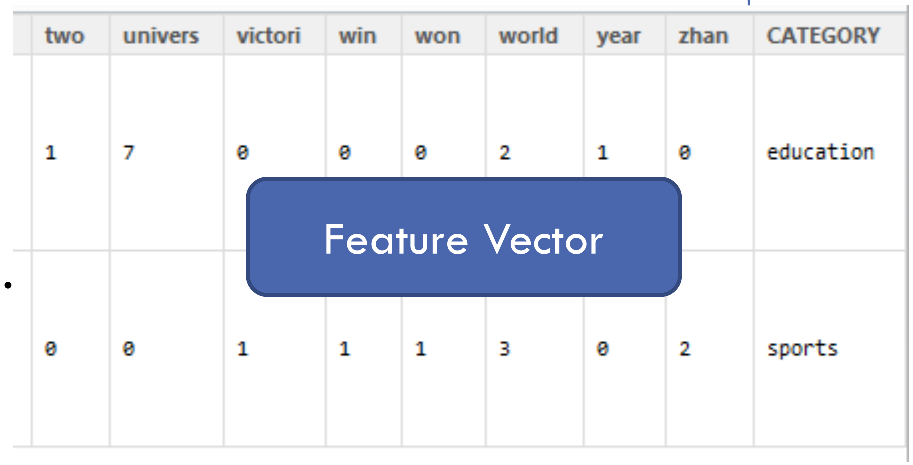
GLASGOW (AFP) - Singapore have won gold in the Commonwealth Games table tennis men's team final against England in replica circumstances to their 2010 victory in New Delhi.

Their trio of Zhan Jian, Gao Ning and Li Hu prevailed 3-1 against England's Liam Pitchford, Paul Drinkhall and Andrew Baggaley.

Zhan, the world No. 32, set Singapore up with a 3-2 win in the best of five match rubbers, as world No. 134 Drinkhall's brave attempt was seen off 11-2 in the final game after the Englishman had led 2-1 in games.

Pitchford was then seen off by world No. 20 Gao in straight games 11-9, 13-11, 11-3.

How can we improve the simple "bag-of-words" model?



two	univers	victori	win	won	world	year	zhan	CATEGORY
1	7	0	0	0	2	1	0	education
0	0	1	1	1	3	0	2	sports

Feature Vector

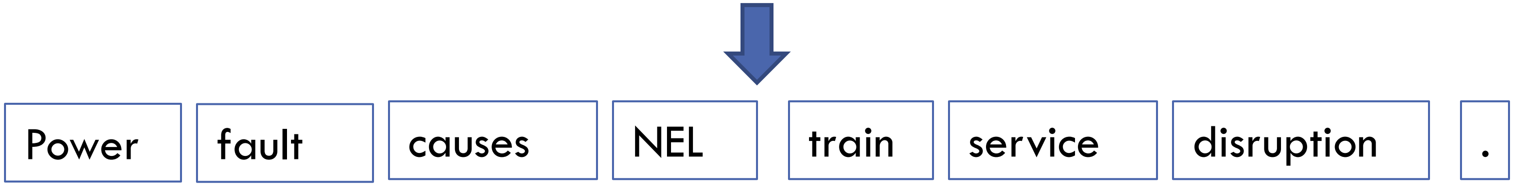
…

Recap of Text Pre-processing

Tokenization

- Process of breaking a sentence into different tokens

Power fault causes NEL train service disruption.



Power	fault	causes	NEL	train	service	disruption	.

- Generally an easy task for English
  - Split the string by space and punctuation
- Some problems for hypenation, apostrophe, periods
  - aren’t → [“are”, “n’t”], o’neill → [“o’neill”]
  - Bishan-Toa Payoh→ [“Bishan ”, “Toa Payoh”], co-education → [“co-education”]
  - Ph.D., Mr.

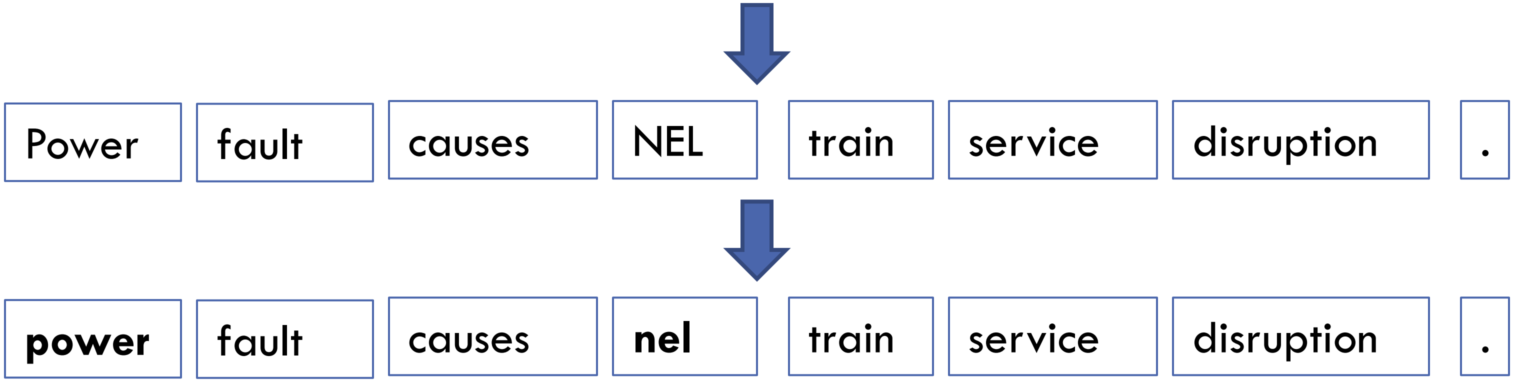
Text Normalization

- Text normalization trying to convert text into a general/standard form
  - E.g. “car” instead of “cars”
    - Want to generalize the text as much as possible
    - So that the machine learning algorithm can learn better
    - I.e. want to have **as few unique tokens as possible**

Convert to Lower Case

- One way to reduce the number of unique tokens is by converting the tokens to lower case

Power fault causes NEL train service disruption.



Power	fault	causes	NEL	train	service	disruption	.
power	fault	causes	**nel**	train	service	disruption	.

- The word ***power*** is then treated as the same token as the ***power*** in “*There is a **power** failure at my area.*”

Remove Punctuation

- Another common text normalization strategy is to remove all the punctuation:
  - Ph.D. → Phd, Mr. → Mr, etc, What?!!! → What
- This essentially reduce the number of features/tokens of the text

Token Replacement

- One effective way to generalize the text is to convert tokens into a more general form
  - E.g.
    14 Sep 2022, 08/08/2000 → DATE
    51, twenty → NUM
    John Smith → PERSON
    :) :] :)) → HAPPY_FACE

Stemming

- In additional to the obvious transformation (change to lower case, etc), we can also transform words to their **stem** (or root form)
  - E.g.
    books → book
    beautiful → beauty
    eats → eat
- This process is called **stemming**

Lemmatization

- Stemming might not result in proper words
  - E.g.

    Europe imports more diesel from MidEast, Asia to replace Russia

    → (apply porter stemming)

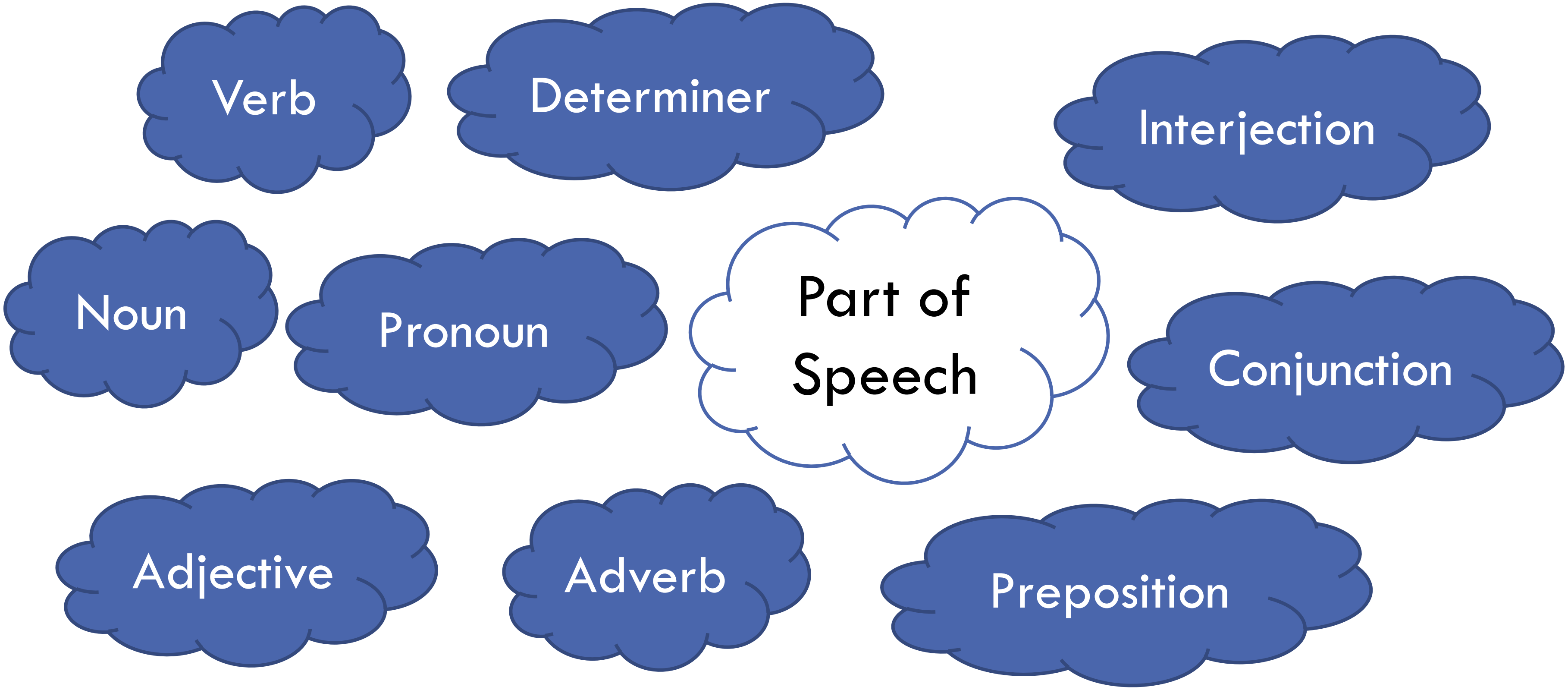
    Europ *import more diesel from MidEast, Asia to* ***replac*** *Russia*
- An alternative to stemming is **lemmatization**
  - Lemmatization also has the same general objective of converting a word to its base form
  - Except that this is done using the help of a **dictionary** (e.g. WordNet)

Recap: Lemmatization

- **Lemmatization** aims to convert words to its base form
- We often also need to provide the **Part of Speech** tags for it to work properly

Part of Speech (POS) Tagging

- □ Each English word can be assigned to one of the 9 **Part of Speech (POS)** tag



Order of performing Text Normalization

- Text preprocessing is often performed as a pipeline
- But be careful of the order by which we perform the different text processing tasks
  - Malaysian ringgit slips to a new 24-year low vs US dollar
    →
    malaysian ringgit slips to a new 24-year low vs **us** dollar

This might be treated as the pronoun "us" instead of the proper noun (country)

Other Text Pre-processing Techniques

Stop Words Removal

- Stop words are the common words that is used in the language
  - E.g.
    a, the, so, them, he, she, who, what, when, how, is, are, etc
- In text processing, stop words are usually ignored to improve performance (speed & accuracy)

Term Weighting

- We have discussed **term frequency**

All the possible features in the entire **corpus (collection of all documents)**

amazed	adapted	books	success	…	…	CLASS
1	0	2	0	…	…	CLASS1
…	…	…	…	…	…	…
0	0	0	3	…	…	CLASS2

Each cell denotes the number of times the token appears in this document

Term Weighting

□ In the lab, we have considered **term presence**

All the possible features in the entire **corpus (collection of all documents)**

amazed	adapted	books	success	…	…	CLASS
true	false	true	false	…	…	CLASS1
…	…	…	…	…	…	…
false	false	false	true	…	…	CLASS2

Each cell denotes whether the token exists in this document

Term Weighting

- Term frequency (TF)
  - Words that occur more ➔ document seems to focus more on that idea
  - Higher value ➔ feature contributes more weight
- Term Presence
  - Some words (after excluding the stop words) appears more/less frequent in the language, it does not they are more/less important compared to the rest

Term Weighting

- Term frequency (TF)
  - Words that occur more ➔ document seems to focus more on that idea
  - Higher value ➔ feature contributes more weight
- An improvement theoretically speaking is to also consider how a word is used in other documents in the corpus

Term Weighting

- tf.idf (term frequency*inverse document frequency) is another approach to reflect how important a word is
  - Made up of 2 components: tf & idf
  - tf = how many times the term appears in the document
  - $$idf = \log\left(\frac{N}{df}\right)$$

    where

    N = total number of doc

    df = document frequency (number of documents with this term)

Term Weighting

$$idf = \log\left(\frac{N}{df}\right)$$

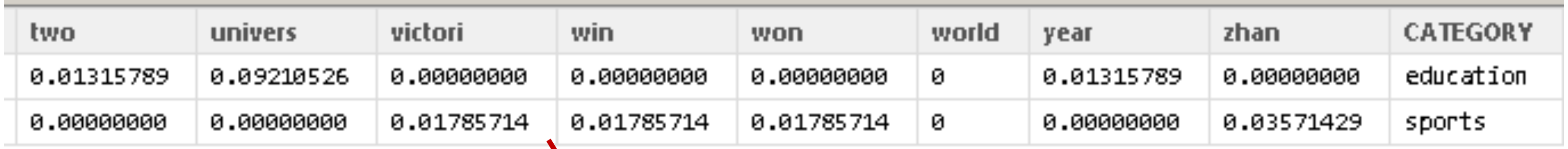
- Example, if we have 100 documents in our corpus, and the term NLP appears in just 1 document
  idf("nlp") = log (100)
- The term "i" (stop word) appears in all the documents
  idf("i") = log(1) = 0
- Justification for idf
  - If a term appears on many documents, each time it appears in a document ➔ probably not important
  - If a term is seldom seen, when it appears ➔ likely to be important (i.e. document is likely to be about it)

tf.idf

- After motivating why idf makes sense, we still need to cater for the fact that, if a term is mentioned many times in a document, it is probably important

- Final weight = tf * idf

Feature Vector (weighted by tf.idf)



two	univers	victori	win	won	world	year	zhan	CATEGORY
0.01315789	0.09210526	0.00000000	0.00000000	0.00000000	0	0.01315789	0.00000000	education
0.00000000	0.00000000	0.01785714	0.01785714	0.01785714	0	0.00000000	0.03571429	sports

The weights for each feature = tf * idf

Note that tf is normalized
(i.e. tf = term frequency / total number terms in doc)

N-Grams

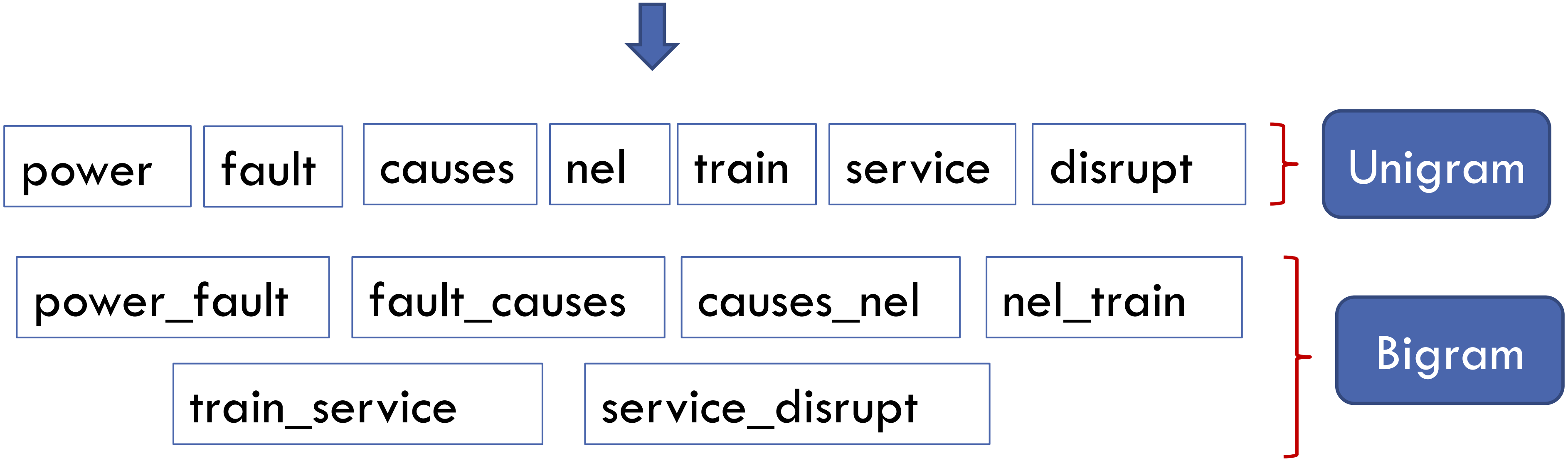
Types of Feature

- So far, the only type of token used is bag-of-words or unigram
- In NLP, the term **n-gram** refers to a contiguous sequence of n tokens
  - Bigram = 2 consecutive tokens
  - Trigram = 3 consecutive tokens
- Sometimes we might want to add in bigram features

Bigram Features

- Bigram example:

Power fault causes NEL train service disruption.



power fault causes nel train service disrupt

Unigram

power_fault fault_causes causes_nel nel_train

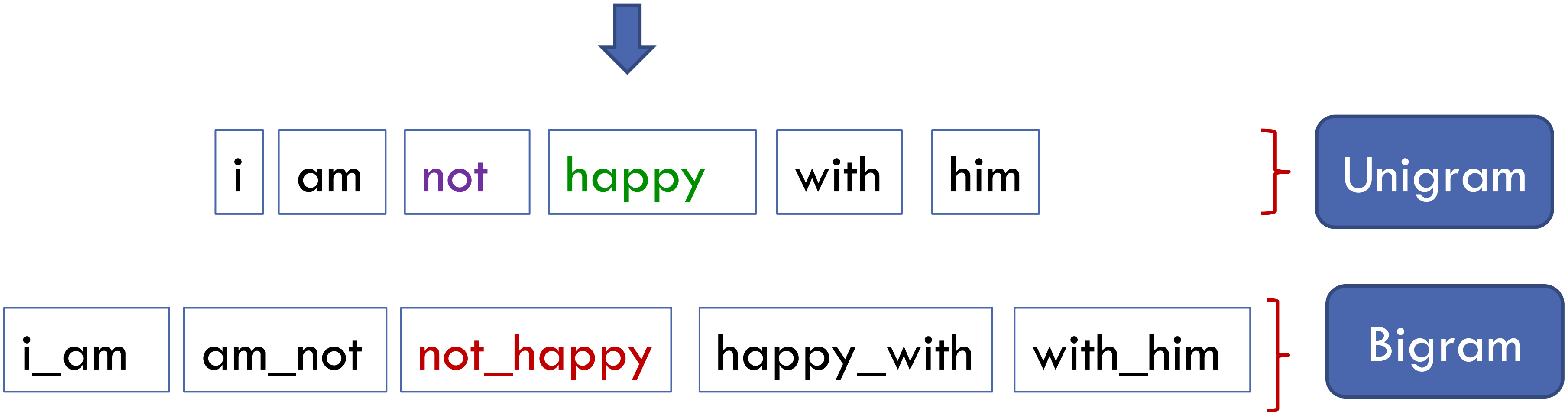
train_service service_disrupt

Bigram

Bigram Features

- Why is bigram a good idea?

I am not happy with him



i | am | not | happy | with | him

Unigram

i_am | am_not | not_happy | happy_with | with_him

Bigram

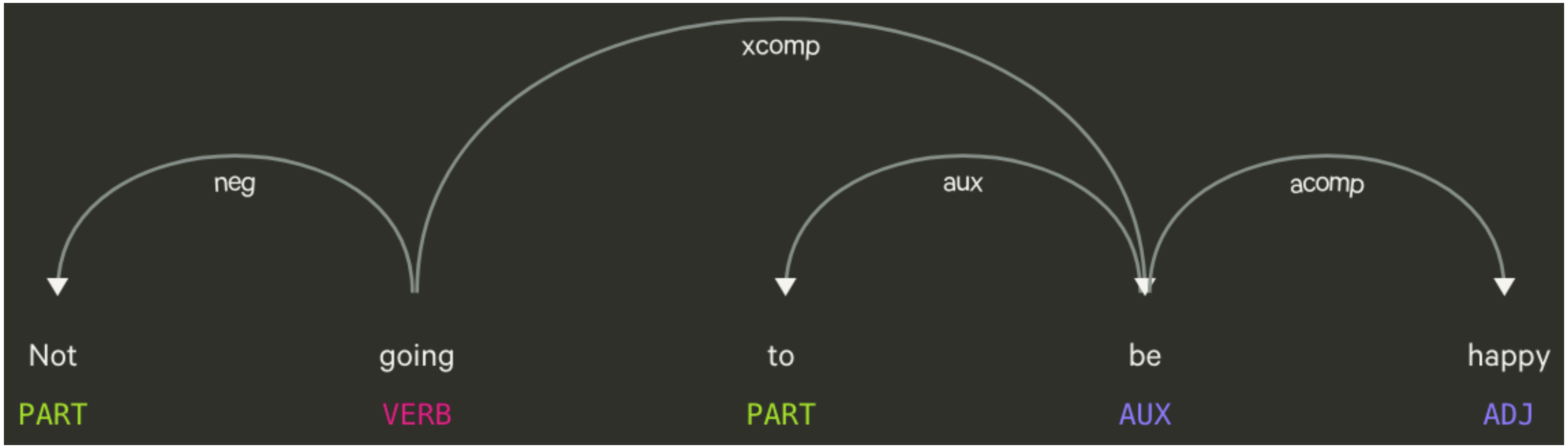
Negation

Negation

- To build an accurate sentiment analyzer, it is important to handle negation cases
  - It's not good :(
  - I don't like it
- By incorporating bigrams, we can handle some of these cases
  - But it is not able to handle every single case
    - ***Not*** *only happy but elated*
    - ***Not*** *going to be happy*

Negation

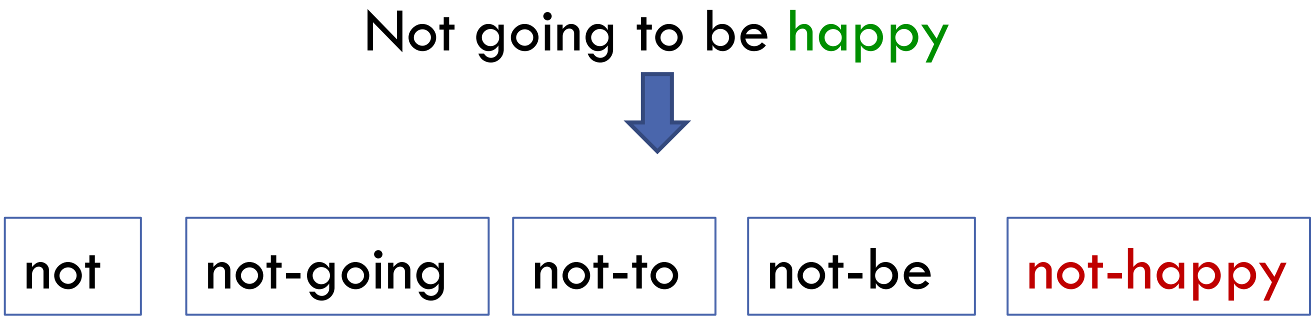
- In order to further improve on this, we could perform **dependency parsing**



- But note that this would require hand crafting various rules to handle the syntactic structures
  - Just a slight change in the sentence structure will lead to a different dependency parse

Negation

- A simpler way to handle this is to define a **negation window size** and artificially toggle the sentiment of the word that is within the negation window size
- For example:
  - Negation window size of 4



Not going to be happy

not | not-going | not-to | not-be | not-happy

SENTIMENT ANALYSIS FUNDAMENTALS

FEATURE ENGINEERING FOR SENTIMENT ANALYSIS

DR LEK HSIANG HUI