COMP30810
Intro to Text Analytics

Dr. Binh Thanh Le
thanhbinh.le@ucd.ie
Insight Centre for Data Analytics
School of Computer Science
University College Dublin

Previous lecture

CLEAN-UP

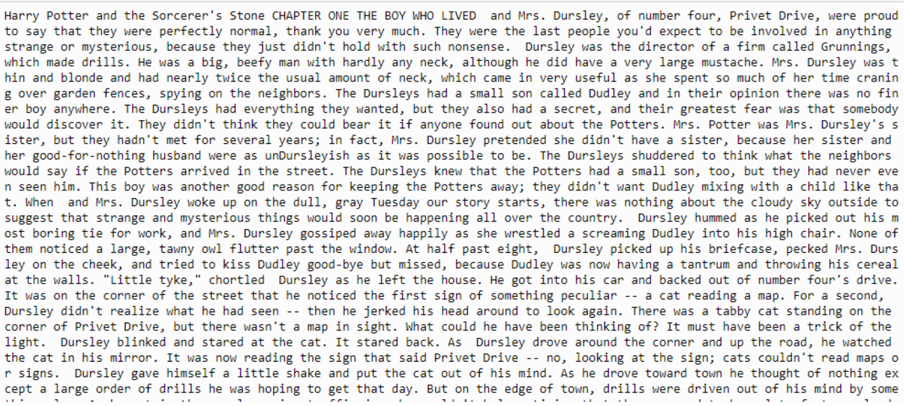
Remove punctuation

Remove stop words

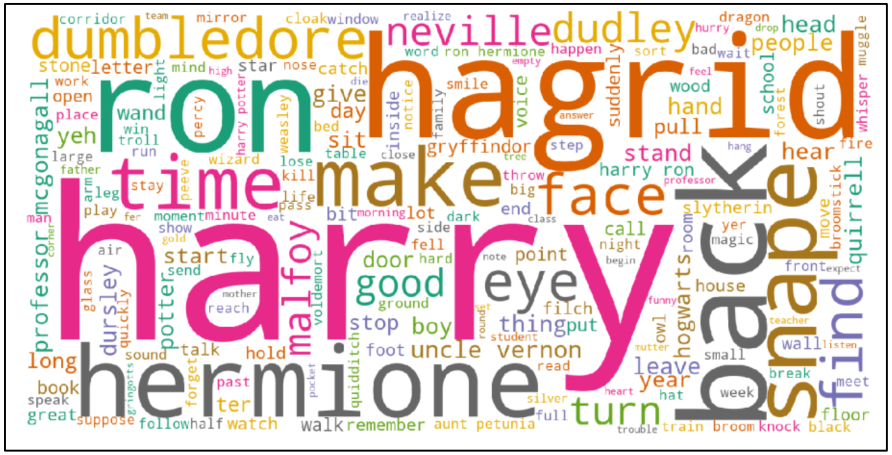
Normalize the case

Stemming | Lemmatization

Key wordlist



```
Harry Potter and the Sorcerer's Stone CHAPTER ONE THE BOY WHO LIVED  and Mrs. Dursley, of number four, Privet Drive, were proud
to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything
strange or mysterious, because they just didn't hold with such nonsense.  Dursley was the director of a firm called Grunnings,
which made drills. He was a big, beefy man with hardly any neck, although he did have a very large mustache. Mrs. Dursley was t
hin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent so much of her time cranin
g over garden fences, spying on the neighbors. The Dursleys had a small son called Dudley and in their opinion there was no fin
er boy anywhere. The Dursleys had everything they wanted, but they also had a secret, and their greatest fear was that somebody
would discover it. They didn't think they could bear it if anyone found out about the Potters. Mrs. Potter was Mrs. Dursley's s
ister, but they hadn't met for several years; in fact, Mrs. Dursley pretended she didn't have a sister, because her sister and
her good-for-nothing husband were as unDursleyish as it was possible to be. The Dursleys shuddered to think what the neighbors
would say if the Potters arrived in the street. The Dursleys knew that the Potters had a small son, too, but they had never eve
n seen him. This boy was another good reason for keeping the Potters away; they didn't want Dudley mixing with a child like tha
t. When  and Mrs. Dursley woke up on the dull, gray Tuesday our story starts, there was nothing about the cloudy sky outside to
suggest that strange and mysterious things would soon be happening all over the country.  Dursley hummed as he picked out his m
ost boring tie for work, and Mrs. Dursley gossiped away happily as she wrestled a screaming Dudley into his high chair. None of
them noticed a large, tawny owl flutter past the window. At half past eight,  Dursley picked up his briefcase, pecked Mrs. Durs
ley on the cheek, and tried to kiss Dudley good-bye but missed, because Dudley was now having a tantrum and throwing his cereal
at the walls. "Little tyke," chortled  Dursley as he left the house. He got into his car and backed out of number four's drive.
It was on the corner of the street that he noticed the first sign of something peculiar -- a cat reading a map. For a second,
Dursley didn't realize what he had seen -- then he jerked his head around to look again. There was a tabby cat standing on the
corner of Privet Drive, but there wasn't a map in sight. What could he have been thinking of? It must have been a trick of the
light.  Dursley blinked and stared at the cat. It stared back. As  Dursley drove around the corner and up the road, he watched
the cat in his mirror. It was now reading the sign that said Privet Drive -- no, looking at the sign; cats couldn't read maps o
r signs.  Dursley gave himself a little shake and put the cat out of his mind. As he drove toward town he thought of nothing ex
cept a large order of drills he was hoping to get that day. But on the edge of town, drills were driven out of his mind by some
```



Today tasks

Understand the text:

- Find the most common words in text
- N-gram
- Frequency
- Document summarization

Observation from a big scale

News Corpus

filename	content	category	keywords
001	Ad sales boost Time Warner profit Quarterly pr...	business	[ad, sale, boost, time, warner, profit, quarte...
002	Dollar gains on Greenspan speech The dollar ha...	business	[dollar, gain, greenspan, speech, dollar, hit,...
003	Yukos unit buyer faces loan claim The owners o...	business	[yukos, unit, buyer, face, loan, claim, owner,...
004	High fuel prices hit BA's profits British Airw...	business	[high, fuel, price, hit, ba, profit, british, ...
005	Pernod takeover talk lifts Domecq Shares in UK...	business	[pernod, takeover, talk, lift, domecq, share, ...
006	Japan narrowly escapes recession Japan's econo...	business	[japan, narrowly, escape, recession, japan, ec...
007	Jobs growth still slow in the US The US create...	business	[job, growth, slow, create, job, expect, janua...
008	India calls for fair trade rules India, which ...	business	[india, call, fair, trade, rule, india, attend...
009	Ethiopia's crop production up 24% Ethiopia pro...	business	[ethiopia, crop, production, 24, ethiopia, pro...
010	Court rejects $280bn tobacco case A US governm...	business	[court, reject, 280bn, tobacco, case, governme...
011	Ask Jeeves tips online ad revival Ask Jeeves h...	business	[jeeves, tip, online, ad, revival, jeeves, lea...
012	Indonesians face fuel price rise Indonesia's g...	business	[indonesian, face, fuel, price, rise, indonesi...

Last time, we learned how to extract the keywords (tokens)

Observation from a big scale

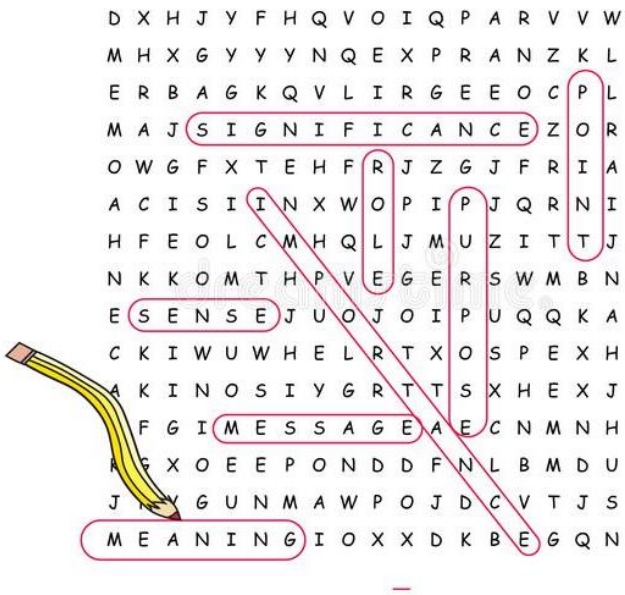
News Corpus

filename	content	category	keywords
001	Ad sales boost Time Warner profit Quarterly pr...	business	[ad, sale, boost, time, warner, profit, quarte...
002	Dollar gains on Greenspan speech The dollar ha...	business	[dollar, gain, greenspan, speech, dollar, hit,...
003	Yukos unit buyer faces loan claim The owners o...	business	[yukos, unit, buyer, face, loan, claim, owner,...
004	High fuel prices hit BA's profits British Airw...	business	[high, fuel, price, hit, ba, profit, british, ...
005	Pernod takeover talk lifts Domecq Shares in UK...	business	[pernod, takeover, talk, lift, domecq, share, ...
006	Japan narrowly escapes recession Japan's econo...	business	[japan, narrowly, escape, recession, japan, ec...
007	Jobs growth still slow in the US The US create...	business	[job, growth, slow, create, job, expect, janua...
008	India calls for fair trade rules India, which ...	business	[india, call, fair, trade, rule, india, attend...
009	Ethiopia's crop production up 24% Ethiopia pro...	business	[ethiopia, crop, production, 24, ethiopia, pro...
010	Court rejects $280bn tobacco case A US governm...	business	[court, reject, 280bn, tobacco, case, governme...
011	Ask Jeeves tips online ad revival Ask Jeeves h...	business	[jeeves, tip, online, ad, revival, jeeves, lea...
012	Indonesians face fuel price rise Indonesia's g...	business	[indonesian, face, fuel, price, rise, indonesi...

How to get the meaning refer to this?

1) Find a word, Concordance

- Helping us reduce the time of analysis.
- Extracting information from context
- Building the first understanding of document
- Making the first statistic for further analysis



1) Find a word, Concordance

```
raw
```

executed in 12ms, finished 15:40:24 2018-08-13

```
'Harry Potter and the Sorcerer\'s Stone \n\nCHAPTER ONE \n\nTHE BOY WHO LIVED \n\nMr. and Mrs. Dursley, of number four, Privet
Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you\'d expect to be in
volved in anything strange or mysterious, because they just didn\'t hold with such nonsense. \n\nMr. Dursley was the director o
f a firm called Grunnings, which made drills. He was a big, beefy man with hardly any neck, although he did have a very large m
ustache. Mrs. Dursley was thin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent
so much of her time craning over garden fences, spying on the neighbors. The Dursleys had a small son called Dudley and in thei
r opinion there was no finer boy anywhere. \n\nThe Dursleys had everything they wanted, but they also had a secret, and their g
reatest fear was that somebody would discover it. They didn\'t think they could bear it if anyone found out about the Potters.
Mrs. Potter was Mrs. Dursley\'s sister, but they hadn\'t met for several years; in fact, Mrs. Dursley pretended she didn\'t hav
e a sister, because her sister and her good-for-nothing husband were as unDursleyish as it was possible to be. The Dursleys shu
ddered to think what the neighbors would say if the Potters arrived in the street. The Dursleys knew that the Potters had a sma
ll son, too, but they had never even seen him. This boy was another good reason for keeping the Potters away; they didn\'t want
Dudley mixing with a child like that. \n\nWhen Mr. and Mrs. Dursley woke up on the dull, gray Tuesday our story starts, there w
as nothing about the cloudy sky outside to suggest that strange and mysterious things would soon be happening all over the coun
```

```
raw.index("Ron")
```

executed in 5ms, finished 15:41:20 2018-08-13

```
132602
```

Return the first index of that word

```
tokens = nltk.word_tokenize(raw)
text= nltk.Text(tokens)
text
```

executed in 580ms, finished 16:53:40 2018-08-13

```
<Text: Harry Potter and the Sorcerer 's
```

Get tokens and Text type

```
text.concordance('Ron')
```

executed in 47ms, finished 16:49:33 2018-08-13

Return concordance in text

```
Displaying 25 of 25 matches:
he said . `` First time at Hogwarts ? Ron 's new , too . '' She pointed at the
u 're nervous . Go on , go now before Ron . '' `` Er -- okay , '' said Harry .
 just taken out her handkerchief . `` Ron , you 've got something on your nose
e of the twins . `` Shut up , '' said Ron . `` Where 's Percy ? '' said their m
' `` It 's not funny . And look after Ron . '' `` Do n't worry , ickle Ronnieki
afe with us . '' `` Shut up , '' said Ron again . He was almost as tall as the
```

1) Find a word, Concordance

'**findall**' : find the multi-word, using Regular Expression

```
1 text.findall(r'<Harry> <.*> <Ron>')
```

executed in 39ms, finished 17:07:53 2018-08-13

```
Harry and Ron; Harry asked Ron; Harry and Ron; Harry and Ron; Harry
and Ron; Harry and Ron; Harry asked Ron; Harry borrowed Ron; Harry and
Ron; Harry and Ron; Harry and Ron; Harry and Ron; Harry remembered
Ron; Harry and Ron; Harry and Ron; Harry , Ron; Harry and Ron; Harry
and Ron; Harry and Ron; Harry filled Ron; Harry and Ron; Harry and
Ron; Harry and Ron; Harry and Ron; Harry and Ron; Harry and Ron; Harry
, Ron; Harry , Ron; Harry and Ron; Harry , Ron; Harry . Ron; Harry and
Ron; Harry and Ron; Harry and Ron; Harry and Ron; Harry and Ron; Harry
and Ron; Harry and Ron; Harry , Ron; Harry and Ron; Harry and Ron;
Harry and Ron; Harry knew Ron; Harry and Ron; Harry , Ron; Harry ,
Ron; Harry and Ron; Harry and Ron; Harry told Ron; Harry and Ron;
Harry and Ron; Harry and Ron; Harry , Ron; Harry , Ron; Harry , Ron
```

Or:

```
1 text.findall(r'<Harry> <\w+> <Ron>')
```

executed in 6ms, finished 17:08:36 2018-08-13

```
Harry and Ron; Harry asked Ron; Harry and Ron; Harry and Ron; Harry
and Ron; Harry and Ron; Harry asked Ron; Harry borrowed Ron; Harry and
Ron; Harry and Ron; Harry and Ron; Harry and Ron; Harry remembered
Ron; Harry and Ron; Harry and Ron; Harry and Ron; Harry and Ron; Harry
and Ron; Harry filled Ron; Harry and Ron; Harry and Ron; Harry and
Ron; Harry and Ron; Harry and Ron; Harry and Ron; Harry and Ron; Harry
and Ron; Harry and Ron; Harry and Ron; Harry and Ron; Harry and Ron;
Harry and Ron; Harry and Ron; Harry and Ron; Harry and Ron; Harry and
Ron; Harry knew Ron; Harry and Ron; Harry and Ron; Harry and Ron;
Harry told Ron; Harry and Ron; Harry and Ron; Harry and Ron
```

Different?

Example for extracting information

Who is the Harry's best friend?

Regular Expression

```
import re
print(len(re.findall(r'Harry and [R]\w+', raw)))
(re.findall(r'Harry and [R]\w+', raw))
```

executed in 6ms, finished 16:38:21 2018-08-14

```
37

['Harry and Ron',
 'Harry and Ron',
 'Harry and Ron',
 'Harry and Ron',
 'Harry and Ron',
```

```
import re
print(len(re.findall(r'Harry and [H][e]\w+', raw)))
(re.findall(r'Harry and [H][e]\w+', raw))
```

executed in 6ms, finished 16:38:57 2018-08-14

```
14

['Harry and Hermione',
 'Harry and Hermione',
 'Harry and Hermione',
 'Harry and Hermione',
 'Harry and Hermione',
 'Harry and Hermione',
```

How many professors are in Harry Potter 1?

```
import re
print(len(set(re.findall(r'[Pp]*rofessor [A-Z]\w+', raw)) ))
set(re.findall(r'[Pp]*rofessor [A-Z]\w+', raw))
```

executed in 24ms, finished 16:45:33 2018-08-14

```
9

{'Professor Binns',
 'Professor Dumbledore',
 'Professor Flitwick',
 'Professor McGonagall',
 'Professor Quirrell',
 'Professor Quirtell',
 'Professor Snape',
 'Professor Sprout',
 'Professor Vindictus'}
```

Regular Expression source

https://docs.python.org/2/howto/regex.html

Regular Expression Test script

https://regex101.com/

2) Top most common words in Text

- We can count the occurrence of a word in text

nltk.Text

```
text.count('Ron') + text.count('ron')
```

```
410
```

ron? → ronnie?

```
raw = 'Ron ronnie Ron Roneo ron'
tokens = nltk.word_tokenize(raw)
text= nltk.Text(tokens)
text
```

```
<Text: Ron ronnie Ron Roneo ron...>
```

```
text.count('Ron') + text.count('ron')
```

```
3
```

```
import re
print(len(re.findall(r' [Rr]on ', ' '.join(text))))
```

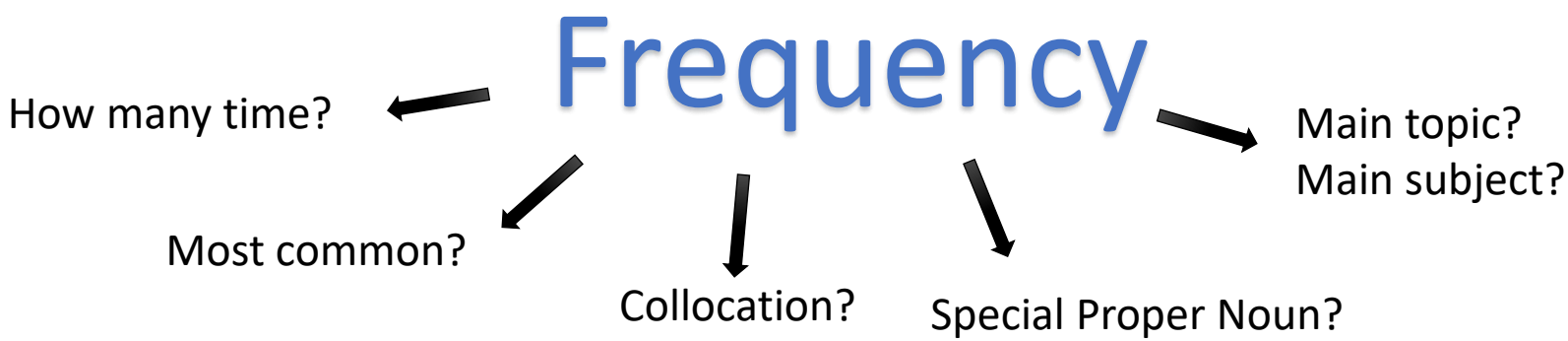
```
410
```

- However, we don't want to use this for **all words** in text file. [Right?]

and now we want to make some statistic analysis for further task.

How? → Frequency

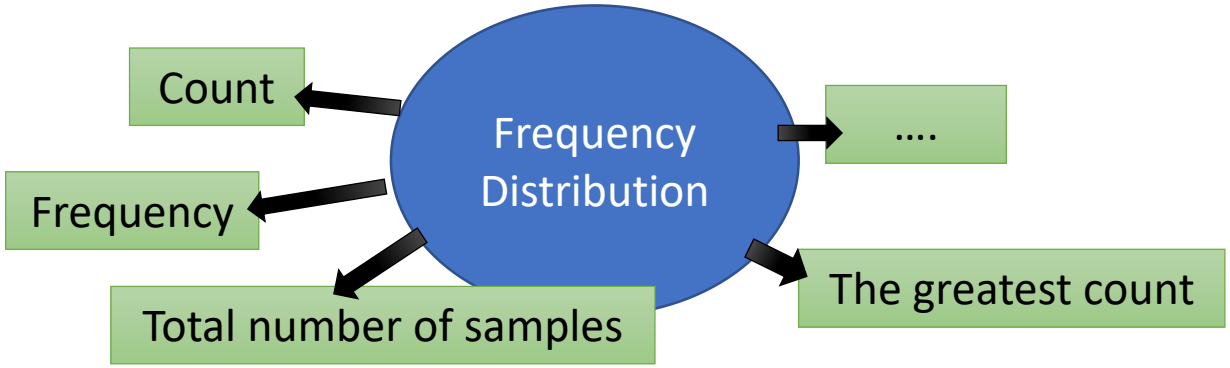
2) Top most common words in Text



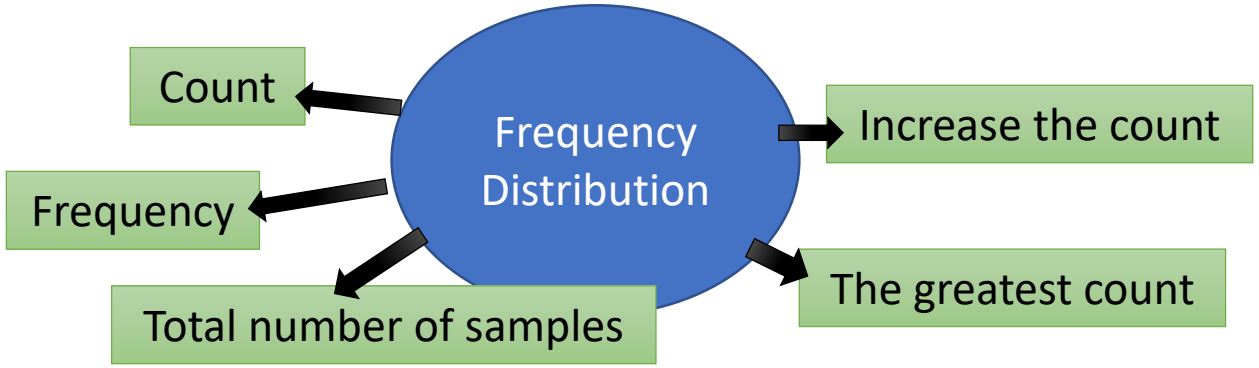
How? → <NLTK> FREQDIST

The `FreqDist` class is used to encode "frequency distributions", which count the number of times that each outcome of an experiment occurs.

How it works...



2) Top most common words in Text



FreqDist functions:

fdist = FreqDist(samples)	→ Create a frequency distribution containing the given samples
fdist['word']	→ Count of the number of times a given sample occurred
fdist.freq('word')	→ Frequency of a given sample
fdist.max()	→ Sample with the greatest count
fdist.tabulate()	→ Tabulate the frequency distribution
fdist.plot()	→ Graphical plot of the frequency distribution
fdist1 < fdist2	→ Test if samples in fdist1 occur less frequently than in fdist2
fdist.inc(sample)	→ Increment the count for this sample
…	

Example:

```
df_handle_h1
```

	content	keywords
0	Harry Potter and the Sorcerer's Stone CHAPTER ...	[harry, potter, sorcerer, stone, chapter, boy,...

```
fdist = nltk.FreqDist(df_handle_h1['keywords'].iloc[0])
```

Create a frequency distribution

```
fdist.max()
```

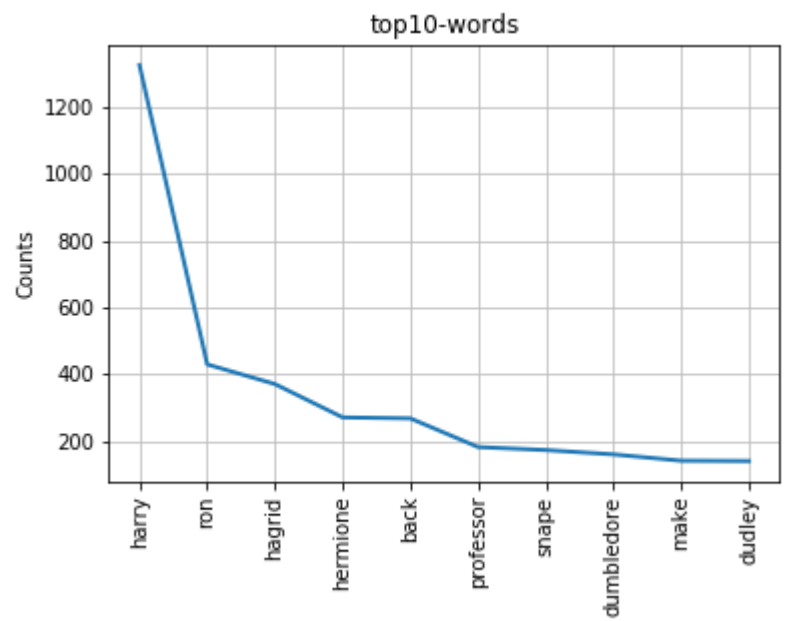
Find the greatest count word

```
'harry'
```

```
#What are the top-10 most frequent words
print("Most frequent top-10 words: ", fdist.most_common(10))
fdist.plot(10, title='top10-words')
```

main subject

```
[('harry', 1327),
 ('ron', 429),
 ('hagrid', 370),
 ('hermione', 270),
 ('back', 267),
 ('professor', 181),
 ('snape', 172),
 ('dumbledore', 159),
 ('make', 140),
 ('dudley', 139)]
```



3) Dispersion plot

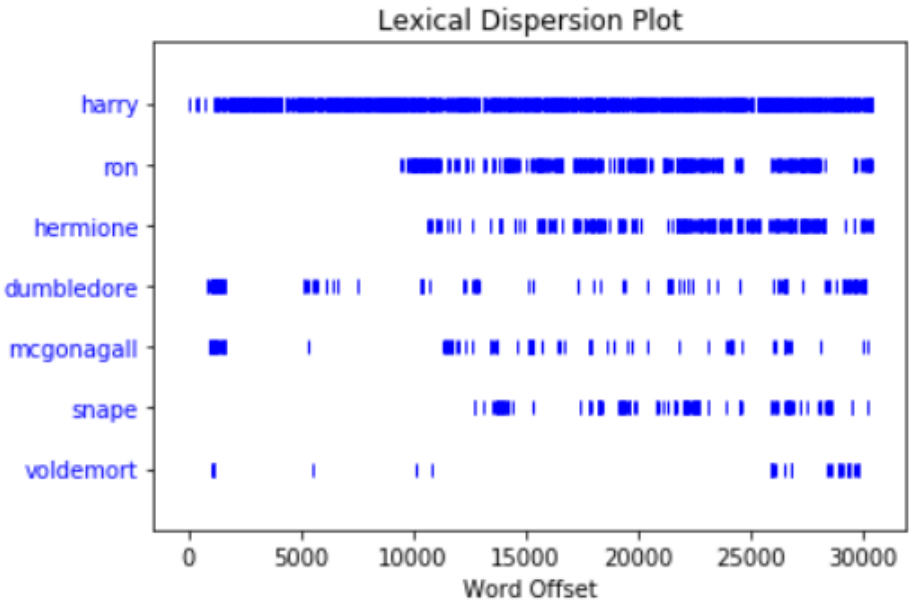
- From *nltk*: this is an utility for displaying lexical dispersion.

```
from nltk.draw.dispersion import dispersion_plot

fdist =  nltk.FreqDist(df_handle_h1['keywords'].iloc[0])
topw = [w[0]+':' + str(w[1]) for w in word_frequency.most_common(5)]

allwords = [w for w in df_handle_h1['keywords'].iloc[0]]

issue = ['harry','ron','hermione','dumbledore','mcgonagall','snape','voldemort']
dispersion_plot(allwords, issue, title='Lexical Dispersion Plot')
```



3) N-gram

- Sometimes collocations are a problem for splitting words

e.g. Fast food, pay attention, catch a cold, …

- Conjunctions, connecting words are also an another problem

e.g. In case, in spite of, even though…

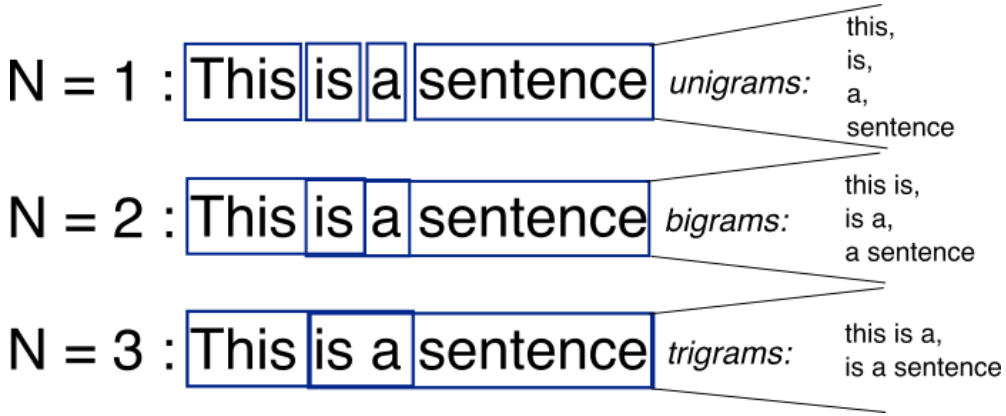
- The proper names are also a problem

e.g. "Harry Potter", "White House", "Burger King", …

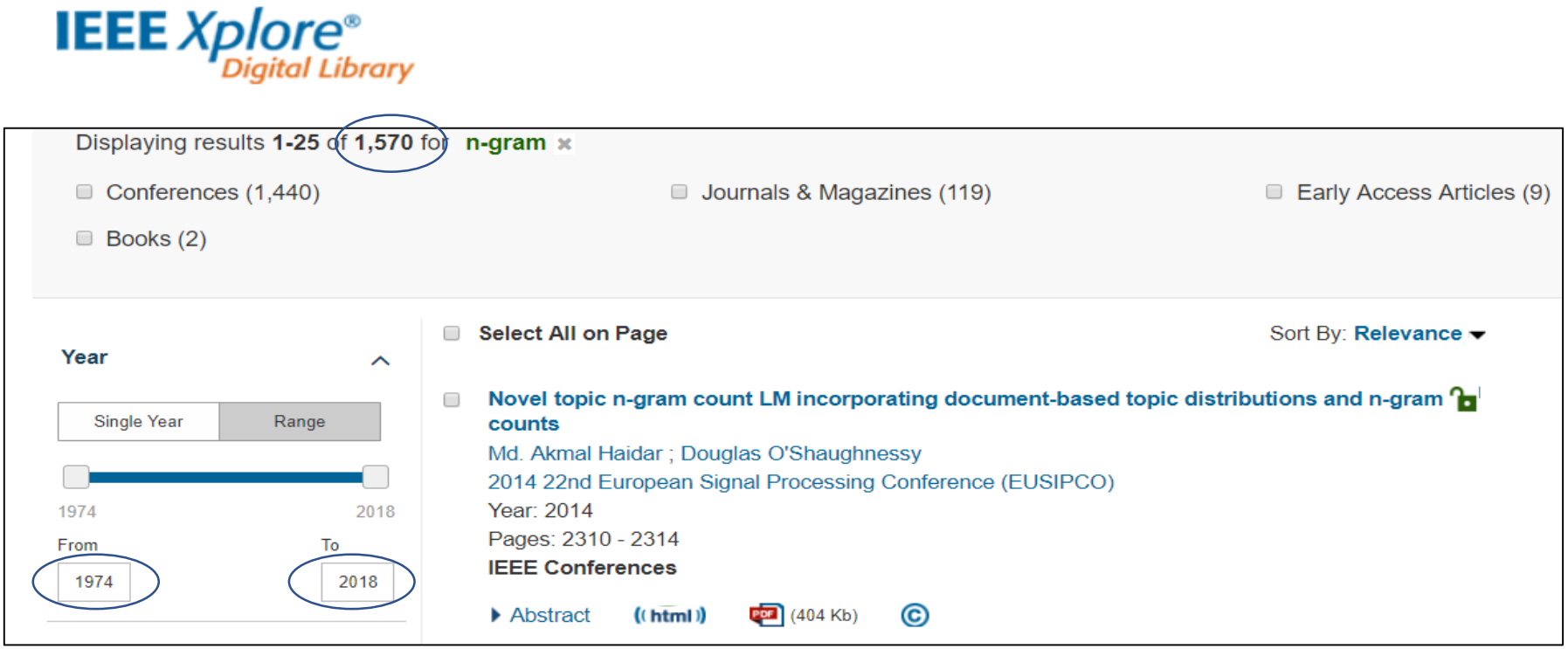
➔ To manage multi-words together: n-gram

*An **n-gram** is a contiguous sequence of n items from a given sample of text or speech*

<Wikipedia>



3) N-gram



Why use n-gram?

- Can capture all possibility of word's combination
→ Better coverage (e.g. collocations, proper noun, noun phrases,…)
→ Higher precision (more data → more information)

Why n-gram can work?

Because words are in order.

3) N-gram

Python code:

```
from nltk import ngrams
sentence = 'This is a sentences and I want to ngramize it'

tokens = sentence.split()

# from nltk.tokenize import RegexpTokenizer
# pattern = r'\w+'
# tokenizer = RegexpTokenizer(pattern)
# tokens = tokenizer.tokenize(sentence)

n = 2
bigram = ngrams(tokens, n)
for grams in bigram:
    print(grams)
```

< In case having punctuations>

```
('This', 'is')
('is', 'a')
('a', 'sentences')
('sentences', 'and')
('and', 'I')
('I', 'want')
('want', 'to')
('to', 'ngramize')
('ngramize', 'it')
```

4) Frequency for n-gram

Python code:

```
# raw text
raw = """The quick brown fox jumps over the lazy dog.
\n He lands head first on a rotting maple log.
\n Knocked unconscious, fox sleeps with shallow breath
\n until the lazy dog awakes and worries him to death."""

# tokenizer
from nltk.tokenize import RegexpTokenizer
pattern = r'\w+'
tokenizer = RegexpTokenizer(pattern)
tokens = tokenizer.tokenize(raw)

# decapitalize
dep_words = [word.lower() for word in tokens]

#Create your bigrams
n =2
bigram = ngrams(dep_words, n)

import collections
from collections import Counter
frequencies = Counter(bigram)

print("sorted by highest frequency first:")
frequencies.most_common(5)
```

Out:

```
sorted by highest frequency first:
[(('the', 'lazy'), 2),
 (('lazy', 'dog'), 2),
 (('the', 'quick'), 1),
 (('quick', 'brown'), 1),
 (('brown', 'fox'), 1)]
```

5) N-gram + POS structure

- To improve the accuracy of n-gram words, we can add POS structure to the n-gram

Example: **Bigrams:** (Noun, Noun), (Adjective, Noun)

```
import collections
from collections import Counter
frequencies = Counter(bigram)

print("sorted by highest frequency first:")
frequencies.most_common(5)
```

<previous slide>

```
sorted by highest frequency first:
[(('the', 'lazy'), 2),
 (('lazy', 'dog'), 2),
 (('the', 'quick'), 1),
 (('quick', 'brown'), 1),
 (('brown', 'fox'), 1)]
```

```
#bigrams
dict_items = list(dict(frequencies).items())
bigramFreqTable = pd.DataFrame(dict_items, columns=['bigram','freq']).sort_values(by='freq', ascending=False)
bigramFreqTable.head(5)
```

	bigram	freq
6	(the, lazy)	2
7	(lazy, dog)	2
0	(the, quick)	1
24	(breath, until)	1
19	(unconscious, fox)	1

<create dataframe>

5) N-gram + POS structure

- To improve the accuracy of n-gram words, we can add POS structure to the n-gram

Example: **Bigrams:** (Adjective, Noun), (Noun, Noun)

```
#function to filter for ADJ/NN bigrams
def rightTypes(ngram):
    if '-pron-' in ngram or 't' in ngram:
        return False

    acceptable_types = ('JJ', 'JJR', 'JJS', 'NN', 'NNS', 'NNP', 'NNPS')
    second_type = ('NN', 'NNS', 'NNP', 'NNPS')
    tags = nltk.pos_tag(ngram)
```

< Adj | Noun > *< Noun >*

```
    if tags[0][1] in acceptable_types and tags[1][1] in second_type:
        return True
    else:
        return False
```

5) N-gram + POS structure

- To improve the accuracy of n-gram words, we can add POS structure to the n-gram

Example: **Bigrams:** (Adjective, Noun), (Noun, Noun)

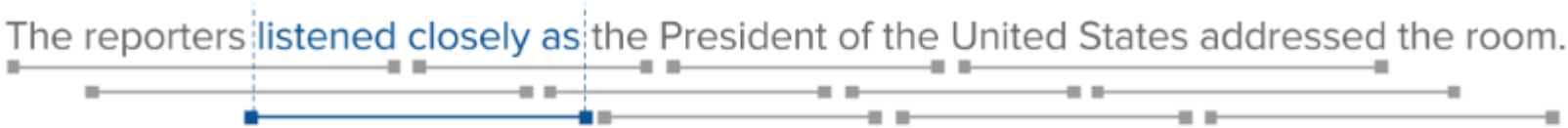
```
filtered_bi = bigramFreqTable[bigramFreqTable.bigram.map(lambda x: rightTypes(x))]
filtered_bi
```

	bigram	freq
7	(lazy, dog)	2
19	(unconscious, fox)	1
23	(shallow, breath)	1
16	(maple, log)	1
1	(quick, brown)	1
2	(brown, fox)	1

<It's better now!>

6) Note about N-gram

- **Choosing the Right n-Gram Window**



- To identify all of the n-grams from our text, we simply slide a fixed-length window over a list of words until the window reaches the end of the list.

Unfortunately, if we build a model based on an n-gram order that is too high, it will be very unlikely that we'll see any repeated entities.

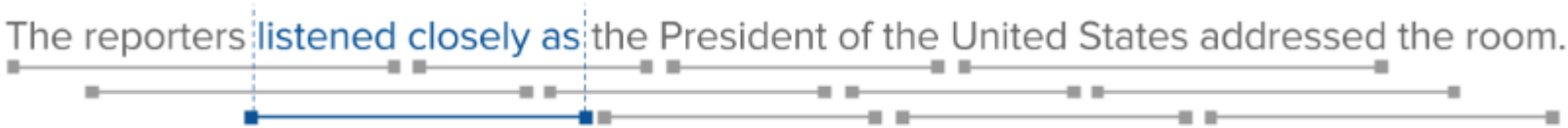
In order to capture the entirety of the phrase "**the President of the United States**," we would have to set

n=6:

```
('The', 'reporters', 'listened', 'closely', 'as', 'the'),
('reporters', 'listened', 'closely', 'as', 'the', 'President'),
('listened', 'closely', 'as', 'the', 'President', 'of'),
('closely', 'as', 'the', 'President', 'of', 'the'),
('as', 'the', 'President', 'of', 'the', 'United'),
('the', 'President', 'of', 'the', 'United', 'States'),
('President', 'of', 'the', 'United', 'States', 'addressed'),
('of', 'the', 'United', 'States', 'addressed', 'the'),
('the', 'United', 'States', 'addressed', 'the', 'room'),
('United', 'States', 'addressed', 'the', 'room', '.')
```

6) Note about N-gram

- **Choosing the Right n-Gram Window**



- To identify all of the n-grams from our text, we simply slide a fixed-length window over a list of words until the window reaches the end of the list.

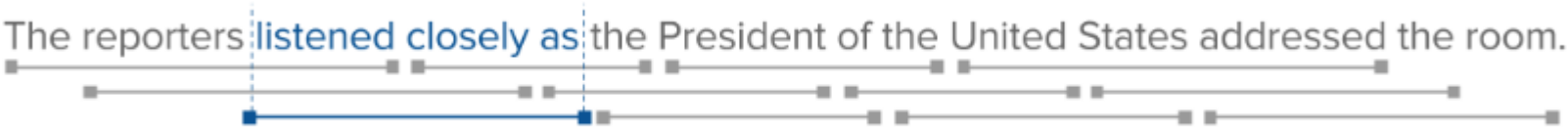
Unfortunately, if we build a model based on an n-gram order that is too high, it will be very unlikely that we'll see any repeated entities.

Too large of an n may **add too much noise** by overlapping independent contexts.

If the window is larger than the sentence, it might not even produce any n-grams at all.

6) Note about N-gram

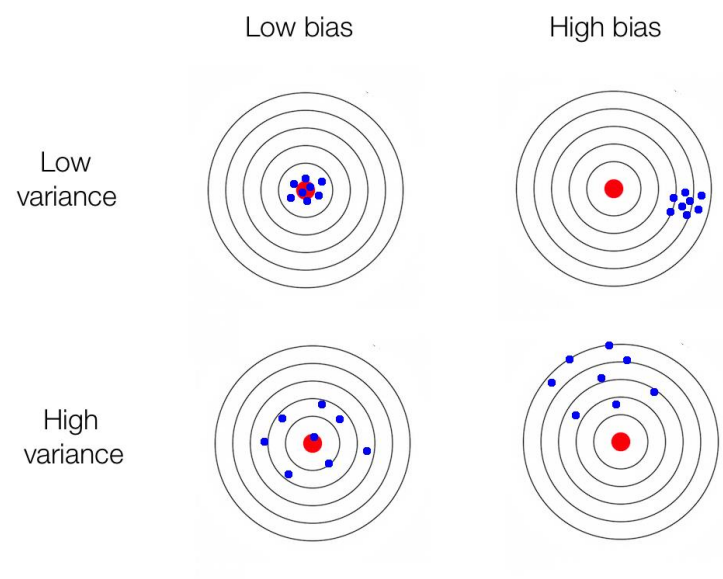
- **Choosing the Right n-Gram Window**



- To identify all of the n-grams from our text, we simply slide a fixed-length window over a list of words until the window reaches the end of the list.

> Unfortunately, if we build a model based on an n-gram order that is too high, it will be very unlikely that we'll see any repeated entities.

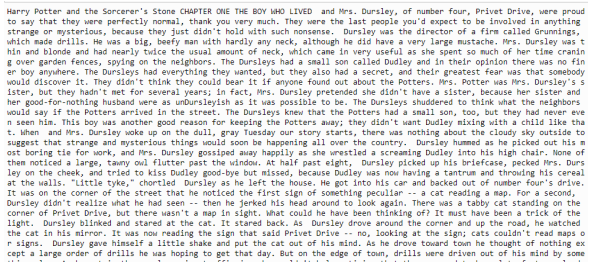
- Choosing **n** can also be considered as balancing the **trade-off between bias and variance**.
  - A small **n** leads to a simpler (weaker) model, therefore causing more error due to bias.
  - A larger **n** leads to a more complex model (a higher-order model), thus causing more error due to variance.



Summary

Corpus

Document



```
Harry Potter and the Sorcerer's Stone CHAPTER ONE THE BOY WHO LIVED  and Mrs. Dursley, of number four, Privet Drive, were proud
to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything
strange or mysterious, because they just didn't hold with such nonsense.  Dursley was the director of a firm called Grunnings,
which made drills. He was a big, beefy man with hardly any neck, although he did have a very large mustache. Mrs. Dursley was t
hin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent so much of her time cranin
g over garden fences, spying on the neighbors. The Dursleys had a small son called Dudley and in their opinion there was no fin
er boy anywhere. The Dursleys had everything they wanted, but they also had a secret, and their greatest fear was that somebody
would discover it. They didn't think they could bear it if anyone found out about the Potters. Mrs. Potter was Mrs. Dursley's s
ister, but they hadn't met for several years; in fact, Mrs. Dursley pretended she didn't have a sister, because her sister and
her good-for-nothing husband were as unDursleyish as it was possible to be. The Dursleys shuddered to think what the neighbors
would say if the Potters arrived in the street. The Dursleys knew that the Potters had a small son, too, but they had never eve
n seen him. This boy was another good reason for keeping the Potters away; they didn't want Dudley mixing with a child like tha
t. When  and Mrs. Dursley woke up on the dull, gray Tuesday our story starts, there was nothing about the cloudy sky outside to
suggest that strange and mysterious things would soon be happening all over the country.  Dursley hummed as he picked out his m
ost boring tie for work, and Mrs. Dursley gossiped away happily as she wrestled a screaming Dudley into his high chair. None of
them noticed a large, tawny owl flutter past the window. At half past eight,  Dursley picked up his briefcase, pecked Mrs. Durs
ley on the cheek, and tried to kiss Dudley good-bye but missed, because Dudley was now having a tantrum and throwing his cereal
at the walls. "Little tyke," chortled  Dursley as he left the house. He got into his car and backed out of number four's drive.
It was on the corner of the street that he noticed the first sign of something peculiar -- a cat reading a map. For a second,
Dursley didn't realize what he had seen -- then he jerked his head around to look again. There was a tabby cat standing on the
corner of Privet Drive, but there wasn't a map in sight. What could he have been thinking of? It must have been a trick of the
light.  Dursley blinked and stared at the cat. It stared back. As  Dursley drove around the corner and up the road, he watched
the cat in his mirror. It was now reading the sign that said Privet Drive -- no, looking at the sign; cats couldn't read maps o
r signs.  Dursley gave himself a little shake and put the cat out of his mind. As he drove toward town he thought of nothing ex
cept a large order of drills he was hoping to get that day. But on the edge of town, drills were driven out of his mind by some
```

Raw text

wordlist

Most common words

Frequency

N-gram

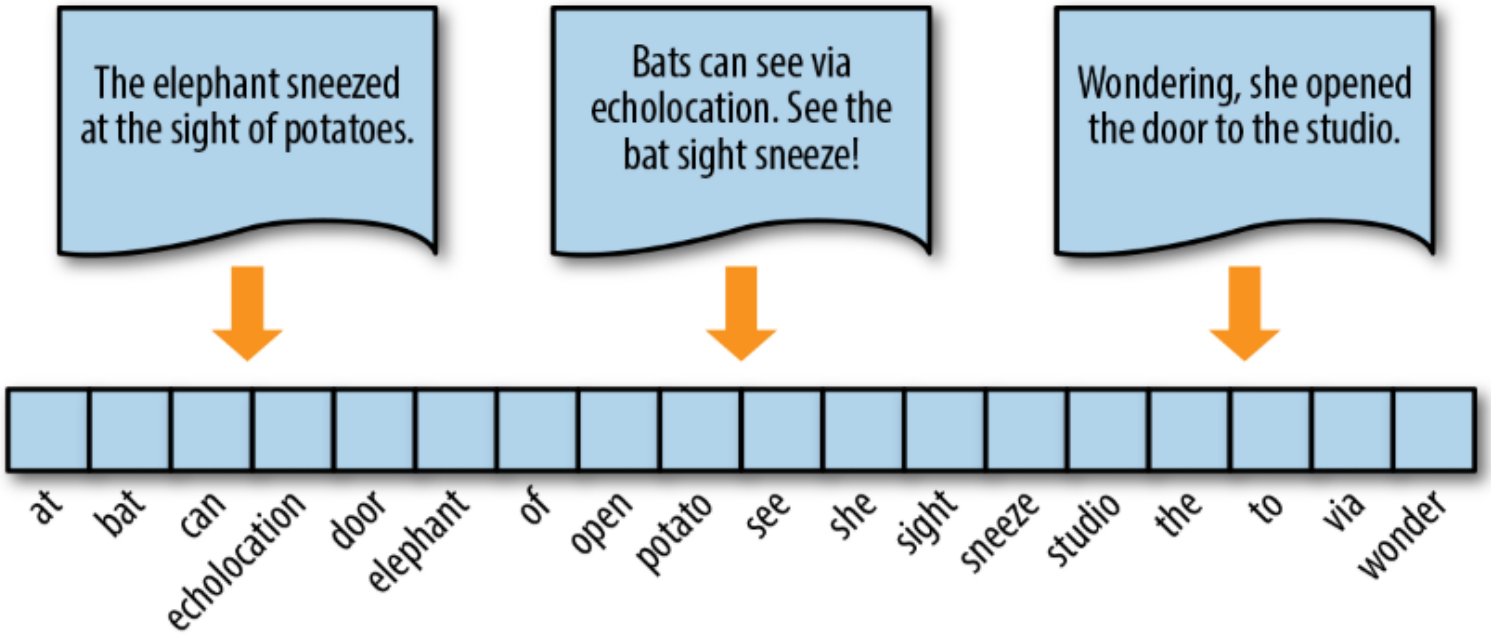
POS

Document Summarization

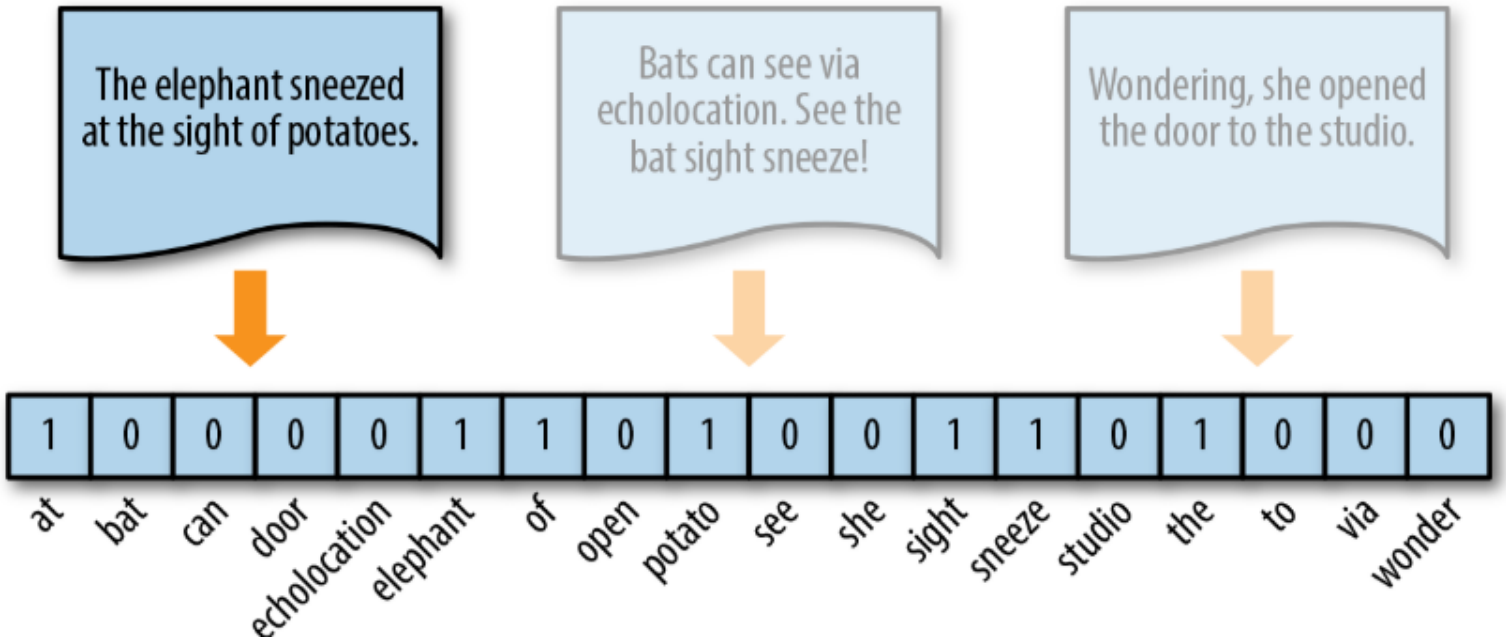
<Meaning>

→Next lecture←

Next lecture



Next lecture



Next lecture