BITI 3413 NLP
TEXT PROCESSING

BY

TS DR HALIZAH BASIRON

What will you learn?

1. ~~Intro to NLP (Lecture 1)~~
2. Text Processing & Normalization (Lecture 2 & 3)
3. Feature Engineering (Lecture 4 & 5)
4. Language Models (Lecture 6)
5. Part Of Speech (POS) Tagging (Lecture 7)
6. Text Classification (Lecture 8)
7. Syntax Analysis (Lecture 9 & 10)
8. Semantic Analysis (Lecture 11 & 12)
9. Discourse Analysis (Lecture 13)
10. NLP Applications (Lecture 14)

TOPIC RECALL

- NLP
- History
- Components of NLP
- Levels of NLP
- Challenges faced when implement NLP
- NLP Datasets, Libraries and API

Learning Outcome for Lecture 2

1) Describe what is a regular expression

2) Know regular expression symbols

3) Distinguish between words tokens and word types

4) Explain the purpose of corpora in NLP

Outline

What is a Regular Expression?

Regular Expression Syntax

Words and Corpora

Outline

What is a Regular Expression?

Regular Expression Syntax

Words and Corpora

POLIS DIRAJA MALAYSIA
REPOT POLIS

Balai : PUTRA PERDANA
Daerah : SEPANG
Kontinjen : SELANGOR
No Repot : PUTRA PERDANA/002114/13
Tarikh : 21/05/2013
Waktu : 1258 PM
Bahasa Diterima : B. Malaysia

Butir-butir Penerima Repot
Nama : [redacted] No Personel : [redacted] Pangkat : [redacted]
Butir-butir Jurubahasa (Jika Ada)
Nama : — No K/P (Baru) : — No Polis/Tentera: —
No Paspot: — Bahasa Asal : —
Alamat: —

Butir-butir Pengadu
Nama : [redacted]
No K/P (Baru) : [redacted] No Polis/Tentera : — No Paspot : —
No Sijil Beranak : —
Jantina : Perempuan Tarikh Lahir : 24/04/1975 Umur : 38 tahun 0 bulan
Keturunan : Cina Warganegara : Malaysia
Pekerjaan :—
Alamat Tempat Tinggal : NO. 2 JALAN PP 2/8 TAMAN PUTRA PRIMA 47100 PUCHONG , SELANGOR.
Alamat Ibu/Bapa : —
Alamat Pejabat : —
No Tel (Rumah) : — No Tel (Pejabat) : — No Tel (HP) 012-345678

Pengadu Menyatakan:-
PADA 21/05/2013 @ JAM LIKURANG 0850 HRS , SEMASA SAMPAI DI HADAPAN TEMPAT KERJA DI ALAMAT NO. 15 JALAN INDUSTRI MAS 10 TAMAN MAS PUCHONG DAN TURUN DARI M/KAR NO. PENDAFTARAN WES 7175 JENIS MERCEDES , TIBA-TIBA DATANG DUA LELAKI BANGSA INDIA DAN MELAYU MENAIKI SEBUAH M/SIKAL JENIS DAN NO. PENDAFTARAN TIDAK PASTI TERUS MENARIK BEG TANGAN SAYA. KEMUDIAN SAYA SEMPAT BERGELUT DENGAN MEREKA DAN SALAH SEORANG DARI MEREKA MENENDANG DI BELAKANG BADAN MENYEBABKAN SAYA TERJATUH DAN MEREKA TERUS MELARIKAN DIRI MENAIKI M/SIKAL TERSEBUT. SAYA MENGALAMI KECEDERAAN KECIL DI KEPALA DAN SIKU SEBELAH KANAN DIDALAM BEG TANGAN SAYA MENGANDUNGI DOKUMEN PENTING ANTARANYA :-

1) KAD PENGENALAN
2) LESEN MEMANDU
3) KAD ATM MAYBANK , HONG LEONG DAN AM BANK

REGULAR EXPRESSION

How to extract date, time, phone numbers, etc. from a police report?

21/05/2013	\d*\/\d*\/\d
012-345678	\d*-\d*
1258 PM	\d*\s[aApP][Mm]

Regular Expression

- How to extract hashtags from a tweet, getting email id or phone numbers etc from a large unstructured text content?

Regular Expression

Hi,

I called Jon on Tuesday, March 25th at 7pm and expressed a concern about my slow times accessing www.cnn.com. He said he would fix it, but I never heard back. Can someone contact me at Kellie.Booth@if.com ASAP? What does Ctrl-F5 mean, by the way?

Thanks

Kellie

Hi, ..., thanks

I called ...

March 27th

ww.blabla.com

Patterns?

Human Brain VS Text Processing

https://www.slideshare.net/MesutGne/regular-expression-regex-fundamentals

Regular Expression

Human Brain VS Text Processing

https://www.slideshare.net/MesutGne/regular-expression-regex-fundamentals

RegEx

- A formal language for specifying text strings.
- It is used in every computer language, word processor, and text processing tools.
- Its purpose is useful for searching in texts, when we have a pattern to search for and a corpus of texts to search through.
- A regular expression search function will search through the text data, returning all texts that match the pattern.
- Also known as regex or regexp or RE.

Regular Expressions: Use case

- Search for specific text format such as email, date of birth, web link.
- Input validation
- Data collection from web scraping
- Text cleaning

https://www.regular-expressions.info/ https://developers.google.com/edu/python/regular-expressions

Outline

What is a Regular Expression?

Regular Expression Syntax

Words and Corpora

Regular expressions

- How can we search for any of these?
  - woodchuck
  - woodchucks
  - Woodchuck
  - Woodchucks

- Tools to explore:
  a. https://regex101.com/
  b. https://www.regexpal.com

Regular Expressions

- Think of regular expressions as wildcards.
- You are probably familiar with wildcard notations such as *.txt to find all text files in a file manager.

The regex equivalent is **^.*\.txt$**

Beginning of string

Matches 0 or more char

End of of string

Regular Expression Symbols & Syntax

1. Metacharacters [ ] { } ( ) ^ $. | * + ? \
2. Quantifier Symbols ?, *, +, .
3. Anchor ^, $
4. Other symbols \d \D \w \W \b \B
5. Positive Lookahead (?=) and Negative Lookahead (?!)
6. Positive Lookbehind (?<=) and Negative Lookbehind (?<!)

https://docs.python.org/3/howto/regex.html

Regular Expression: Metacharacters

- Metacharacters don't match themselves.
- They have special meaning.

Symbols	Matches
[ ]	Specify a set of characters to match
^	Negation
\	Represent predefined set of characters
.	Match any single characters

Regular Expressions: [ ]

- Letters inside square brackets [ ]
- Provide a list of potential matching characters at a position in the search text.

Pattern	Matches
[wW]oodchuck	Woodchuck woodchuck
[1234567890]	Any digit
7[Pp][Mm]	7PM 7pm 7pM 7Pm
[123456789][aApP][Mm]	2pm 3Pm 5am 9AM

Regular Expressions: [ ]

- Ranges `[A-Z]`, `[a-z]`, `[0-9]`

Pattern	Matches	Examples
`[A-Z]`	Any upper case letter	`Drenched Blossoms!`
`[a-z]`	Any lower case letter	`Drenched Blossoms!`
`[a-zA-Z]`	Any single character in the range a-z or A-Z	`We'll look at`
`[0-9]`	A single digit	`Chapter 1: Down the Rabbit Hole`

Regular Expressions: The . symbol

- Match any single characters

Pattern	Matches	Examples
.	Match any single characters except a new line	Hola ! Hello
[.]	Match "."	Put a pull stop . here.

Regular Expressions: Negation in [ ]

- Negations [^]

Pattern	Matches	Examples
[^A-Z]	Not an uppercase letter	Halizah
[^Ss]	Neither 'S' nor 's'	reason"
[^e^]	Neither 'e' nor '^'	Look^here
a[^b]	'a' followed by anything except 'b'	about metacharacters
[^0-9A-F]	Any characters except 0-9 and A-F	123DingDong

Regular Expression: Quantifiers

Quantifier Symbols ? * + Repetition characters

Quantifier	Matches	Same as
?	Match zero or one time	{0,1}
*	Match zero or more times	{0, }
+	Match one or more times	{1, }

Regular Expressions: Quantifier ? * + .

Pattern	Matches	Examples
colou?r	Optional previous char	color colour
oo*h!	0 or more of previous char	oh! ooh! oooh! ooooh!
o+h!	1 or more of previous char	oh! ooh! oooh! ooooh!
baa+		baa baaa baaaa baaaaa
beg.n		begin begun begun beg3n

Stephen C Kleene

Kleene *, Kleene +

Regular Expression: String Boundaries

^ : Start of line, not inside [ ]

$: End of line

\A Start of string

\z End of string

Pattern	Matches	Examples
^little	Match "little" at the start of a string / line	little pony little baby
\Alittle	Match "little" at the start of a string	little pony little baby
baby$	Match "baby" at the end of a string /line	little baby cute baby
baby\z	Match "baby" at the end of a string	little baby cute baby

Regular Expressions: Anchors ^ $

Pattern	Matches
^[A-Z]	Palo Alto
^[^A-Za-z]	1 “Hello”
\.$	The end.
\?$	The end?
!$	The end!

Regular Expressions: Disjunction symbol |

- Woodchucks is another name for groundhog!
- The pipe | for disjunction

Pattern	Matches
groundhog|woodchuck	
yours|mine	what are yours are mine too
a|b|c is similar to [abc]	Little pony little baby
[gG]roundhog|[Ww]oodchuck	

Regular Expression: Curly Brackets { }

- Repetition of characters

Pattern	Matches	Examples
`{n}` `o{2}`	*n* times	oh! ooh! oooh! ooooh!
`{n,}` `o{3,}`	At least *n* times, but no upper limit	oh! ooh! oooh! ooooh!
`{n, m}` `o{2,3}`	Between *n* and *m* times	oh! ooh! oooh! oooh!

Regular Expression: Other Symbols

\b Any word boundary character

\B Anything but a word boundary

\W Any non-word character

\w Any word character (letter, number, underscore)

\D Any non-digit

\d Any digit

\S Any non-whitespace character

\s Any whitespace character

https://www.slideshare.net/MesutGne/regular-expression-regex-fundamentals

Regular Expression: Non-Printable Characters

\n → Matches a new line

\t → Matches a tab character

\a → Matches the bell character

\r → Matches a carriage return

\f → Matches form feed

\v → Matches a vertical tab

\u20AC → Euro

\u00A3 → British pound

\u00A5 → Yen

\$ or \u0024 or \x24 → Dollar sign $

Example:

```
r" [RM\$\u20AC\u00A3\u00A5]\d*.\d*
```

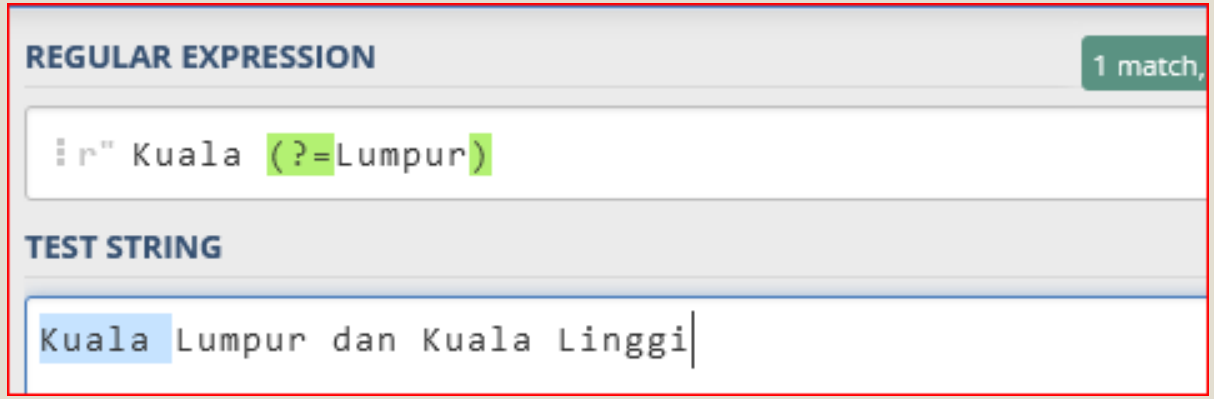
TEST STRING

```
Currency converter: RM1000 is equivalent to $241.51,
€205.21, ¥25446.45, and £185.52
```

Regular Expression: Positive Lookahead Assertion (?=...)

- Matches if ... matches next, but doesn't consume any of the string.
- For example:

Kuala (?=Lumpur) will match 'Kuala ' only if it's followed by 'Lumpur'.



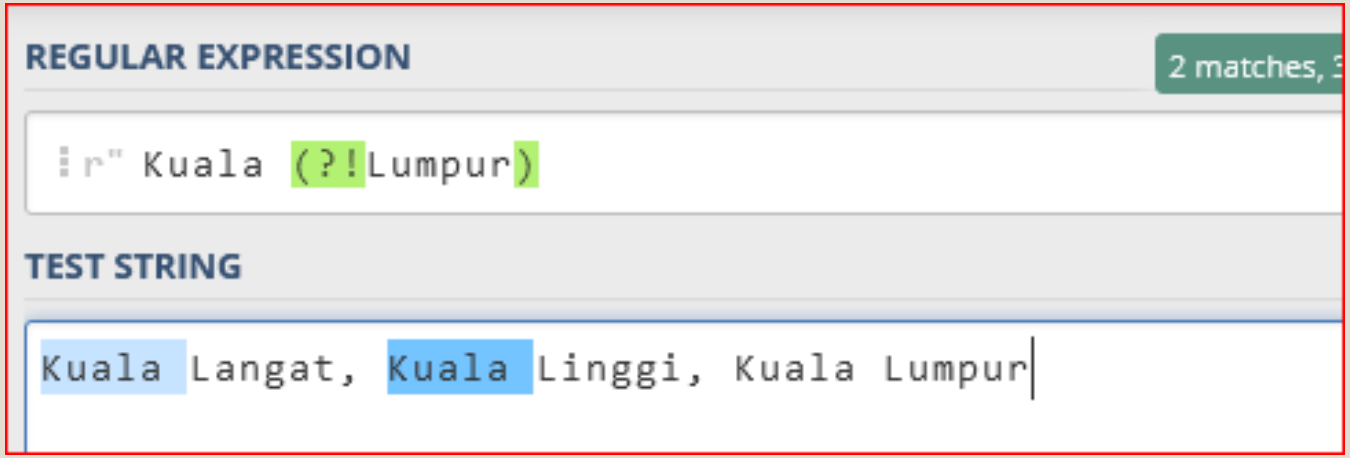
https://docs.python.org/3/library/re.html

Regular Expression: Negative Lookahead Assertion (?!...)

Matches if ... doesn't match next.

For example:

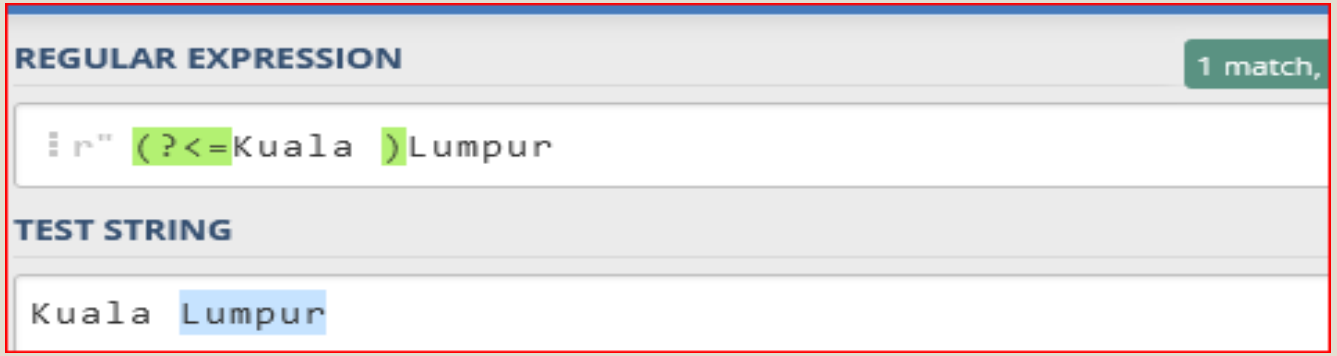
Kuala (?=Lumpur) will match 'Kuala ' only if it's **not** followed by 'Lumpur'.



https://docs.python.org/3/library/re.html

Regular Expression: Positive Lookbehind Assertion (?<=...)

Matches if the current position in the string is preceded by a match for ... that ends at the current position.



For example:

(?<=Kuala)Lumpur will find a match in ‘Kuala Lumpur‘ that means Kuala must be followed by Lumpur.

https://docs.python.org/3/library/re.html

Regular Expression: Positive Lookbehind Assertion (?<=...)

REGULAR EXPRESSION 2 matches,

```
r" (?<=. )[L|l]umpur
```

TEST STRING

```
berada di Kuala Lumpur.
Tanah lumpur
```

Search for any words that followed by ‘Lumpur’ or ‘lumpur’.

This example looks for a word following a hyphen:

```
>>> m = re.search(r'(?<=-)\w+', 'spam-egg')
>>> m.group(0)
'egg'
```

https://docs.python.org/3/library/re.html

Regular Expression: Negative Lookbehind Assertion (?<!...)

Matches if the current position in the string is not preceded by a match for ... .

For example:

(?<=!Kuala)Lumpur will find a match in any words, except Kuala, that followed by Lumpur.

https://docs.python.org/3/library/re.html

Why RE is useful?

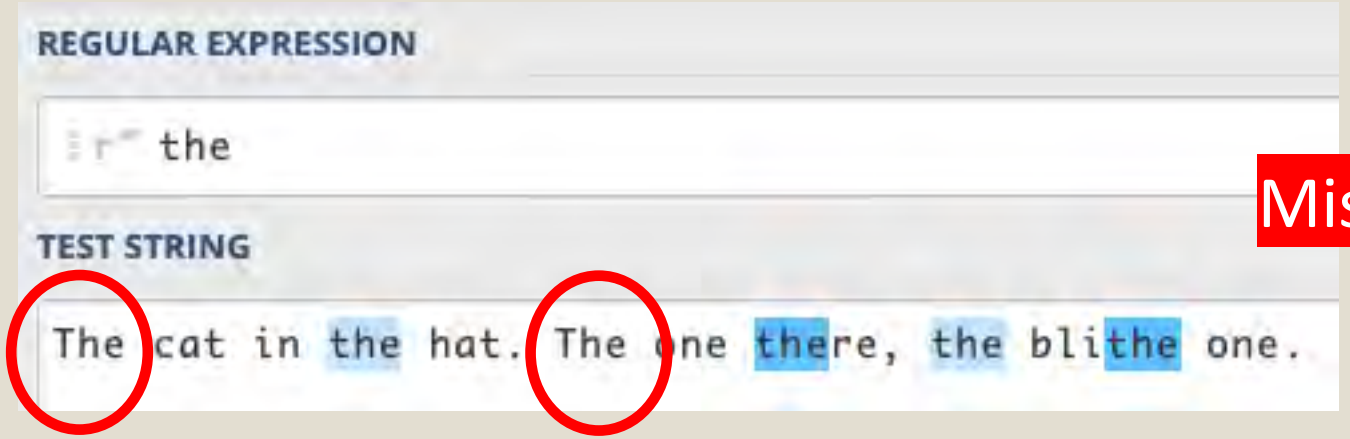
- To fix two kinds of errors
  1. False positives (Type I) - Matching strings that we should not have matched.
  2. False negatives (Type II) - Not matching things that we should have matched.

	Actual -- True/False	
Predicted -- Positive/Negative	True Positive	False Positive (Type I)
	False Negative (Type II)	True Negative

Example:

"The cat in the hat. The one there, the blithe one."

Find me all instances of the word "the" in a text.



Misses capitalized examples

2

Type ~~1~~ error:

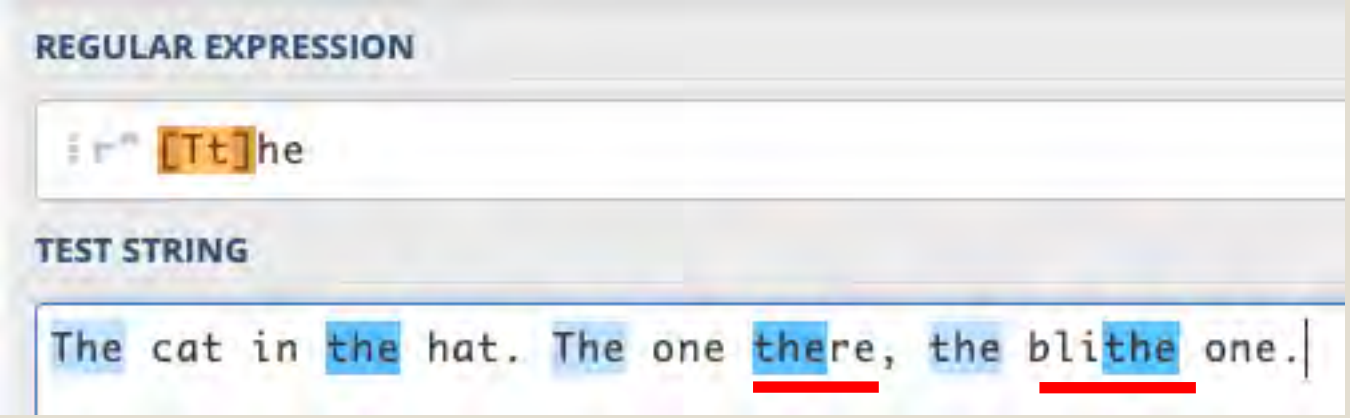
False ~~Positive~~

Negative

Example:

"The cat in the hat. The one there, the blithe one."

Find me all instances of the word "the" in a text.



1

Type ~~2~~ error
(False ~~Negative~~)

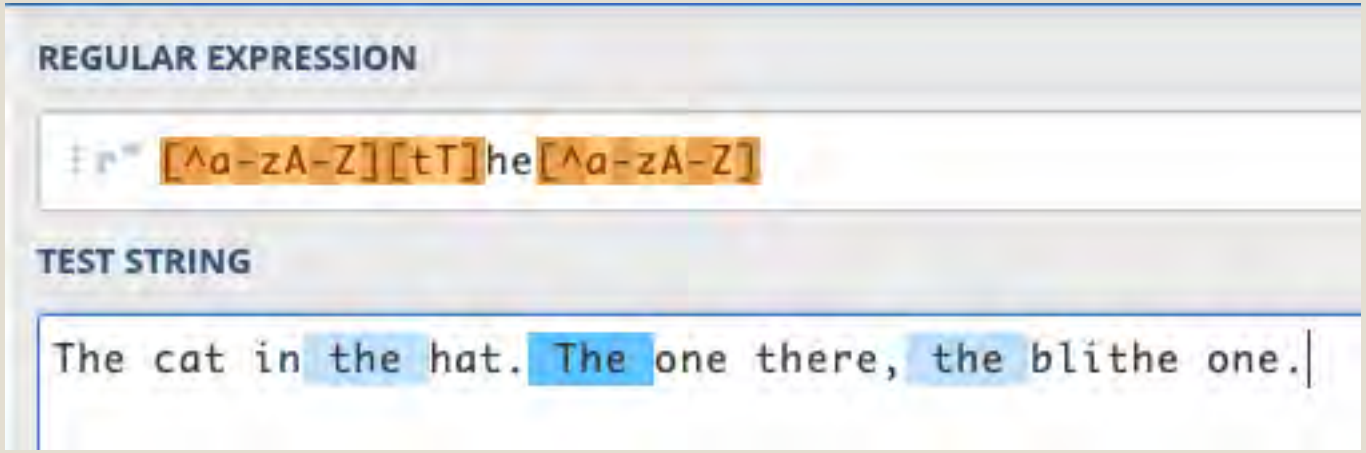
Positive

Incorrectly returns "there" or "blithe".

Example:

"The cat in the hat. The one there, the blithe one."

Find me all instances of the word "the" in a text.



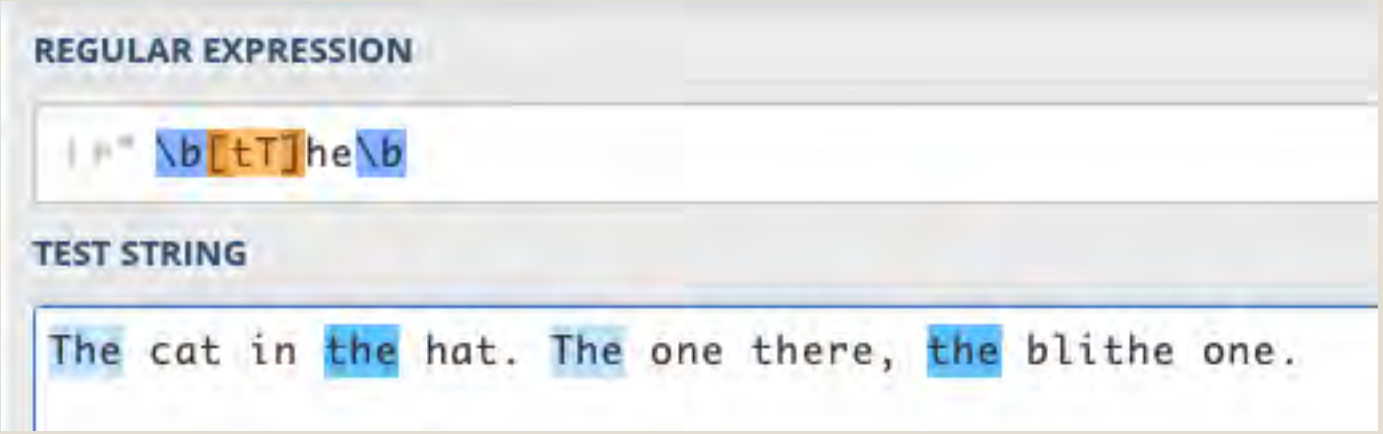
[^a-zA-Z][tT]he[^a-zA-Z]

Not all "the" are captured.

Example:

"The cat in the hat. The one there, the blithe one."

Find me all instances of the word "the" in a text.



\b[tT]he\b

Confusion Matrix- Precision & Recall

	Actual -- True/False	
Predicted -- Positive/Negative	True Positive	False Positive (Type I)
	False Negative (Type II)	True Negative

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

Errors

- In NLP (or ML) we are always dealing with these kinds of errors.
- Reducing the error rate for an application often involves two opposite efforts:
  - Increasing accuracy or precision (minimizing false positives)
  - Increasing coverage or recall (minimizing false negatives).

Outline

What is a Regular Expression?

Regular Expression Syntax

Words and Corpora

How many words in a sentence?

- "I do uh main- mainly business data processing"
  - Fragments, filled pauses
- "Seuss's cat in the hat is different from other cats!"
  - Lemma: same stem, part of speech, rough word sense
    - cat and cats = same lemma
  - Wordform: the full inflected surface form
    - cat and cats = different wordforms

Sentence: they lay back on the San Francisco grass and looked at the stars and their

- Type: the total number of distinct words in a text*.
- Token: the total number of words in a text*.
- How many?
  - 15 tokens
  - 13 types (or 12) (or 11)

How many words in a sentence?

* depending on our goal, different wordforms, same lemma may affect the numbers

How many words in a corpus?

N = number of tokens

V = vocabulary = set of types, $|V|$ is size of vocabulary

There is a relationship between N and V.

According to Heaps Law = Herdan's Law = $$|V| = kN^{\beta}$$

where K is a positive constant and β is between 0 and 1. K is often up to 100 and β is often between $.67 < \beta < .75$.

i.e., vocabulary size grows with > square root of the number of word tokens

How many words in a corpus?

	Tokens = N	Types = |V|
Switchboard phone conversations	2.4 million	20 thousand
Shakespeare	884,000	31 thousand
COCA	440 million	2 million
Google N-grams	1 trillion	13+ million

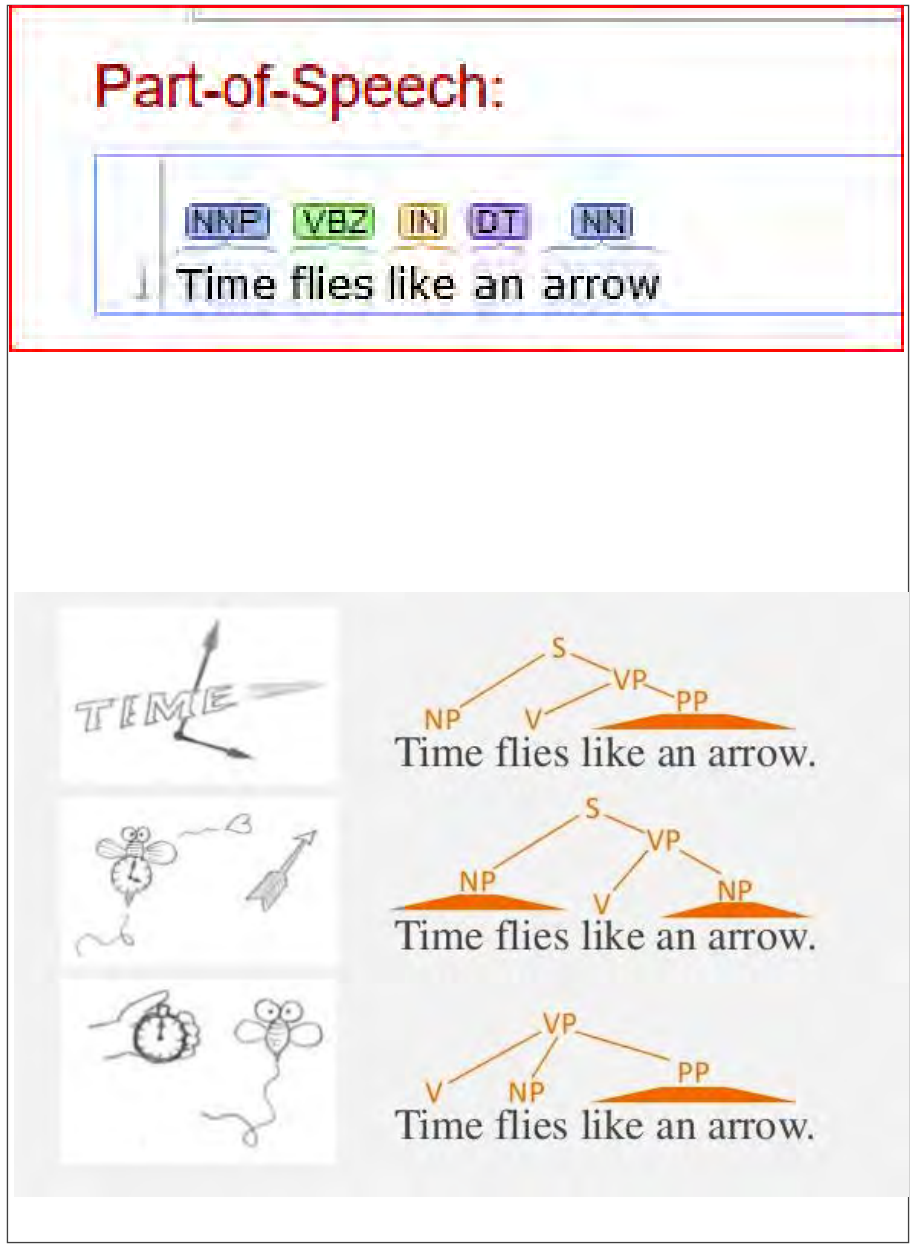
Corpora in NLP

- A corpus (or corpora, in plural) is a big collection of texts on a particular subject that are machine readable and have been published in a natural communication setting.
- It can be derived in a variety of ways, such as from electronic text that was originally written, audio transcriptions, optical character recognition, etc.

Corpora in NLP

- A corpus is not just a random of collection text!

- A text is produced by
  - a specific writer(s),
  - at a specific time,
  - in a specific variety,
  - of a specific language,
  - for a specific function.



Corpora in NLP

- In NLP, corpora are generated with a specific objective in mind: the data should provide the best representation possible for the objective.
- Examples of such tasks are part of speech tagging, coreference resolution, machine translation, and word sense disambiguation.

Examples of Corpora in NLP

1. Gutenberg Corpus:

- Project Gutenberg includes small section of texts from electronic text archive.
- Gutenberg contains some 25,000 free electronic books, hosted as http://www.gutenberg.org/.

2. Web and Chat Text:

- A small collection of web text that is available in NLTK.
- It includes content from a Firefox discussion forum, movie script of Pirates of the Caribbean, personal advertisements, and wine reviews.

Examples of Corpora in NLP

1. Brown Corpus:

- The first million-word electronic corpus in English, created in 1961 at Brown University.
- This corpus contains text from 500 sources, and the sources have been categorized by genre, such as news, editorial, and so on.

2. Reuter Corpus:

- This corpus contains 10,788 news documents totaling 1.3 million words.
- They are classified into 90 topics and they are grouped into two sets, called training and test.

Variety of Corpora

- Language: 7097 languages in the world
- Variety, like African American Language (AAE) varieties.
  - AAE Twitter posts might include forms like "*iont" (I don't)*
- Code switching, e.g., Malay/English, Spanish/English:

  M/E: Sabar tunggu orders from Food Panda and Happy Fresh because you know lazy

  [Patiently waiting for the orders from Food Panda and Happy Fresh because you know, lazy]

  S/E: Por primera vez veo a @username actually being hateful! It was beautiful:)

  [For the first time I get to see @username actually being hateful! it was beautiful:)]
- Genre: newswire, fiction, scientific articles, Wikipedia
- Author Demographics: writer's age, gender, ethnicity

Exercises:

Suppose a text:

This thesis presents a study in the area of computer assisted language learning. The study focusses on the topic of automatic correction of student errors.

How many tokens?

How many types?

- Duit hadiah yg diambil dr yuran kemasukkan tak boleh..klu yuran dikenakan just utk byr kos seliaan, beli minum..bole ...
- tula aku pasan gak tuh sbb tuh aku nak amek sgt jwtn tuh.. aku nampak ble xali pegang byk benda2 yg xelok blaku..

Summary

- Regular expressions play a surprisingly large role
  - Sophisticated sequences of regular expressions are often the first model for any text processing text.
  - Regular expressions can be used as features in classifiers.
- Words
  - Tokens & types
- Corpora
  - Purpose of corpora in NLP

Regular Expression

Regular Expression Syntax

Words & Corpora

Supplementary (Readings)

1. Beginners Guide to Regular Expressions in Natural Language Processing

   https://www.analyticsvidhya.com/blog/2021/03/beginners-guide-to-regular-expressions-in-natural-language-processing/

2. Natural Language Processing Made Simpler with 4 Basic Regular Expression Operators

   https://towardsdatascience.com/natural-language-processing-made-simpler-with-4-basic-regular-expression-operators-5002342cbac1

3. NLP Corpora and Lexical Database

   https://medium.com/@deepak.engg.phd/part-2-nlp-text-corpora-and-lexical-database-ab4749e2ab0b

Supplementary (Videos)

1. Regular Expressions In Python (17:56). https://www.youtube.com/watch?v=WQlKPdKVXfw
2. How to Write and Match Regular Expressions (53:17). https://www.youtube.com/watch?v=K8L6KVGG-7o

Our Next Topics will be on

- Text Normalization
  - Word Tokenization
  - Sentence Segmentation
  - Word Format Normalization

THANK YOU