Natural Language Processing

Basic Text Processing

Joakim Nivre

Uppsala University
Department of Linguistics and Philology
joakim.nivre@lingfil.uu.se

Basic text processing

```
J.P. Bolduc, vice chairman of W.R. Grace & Co., which
holds a 83.4% interest in this energy-services company,
was elected a director. Clark J. Vitulli was named senior
vice president and general manager of this U.S. sales and
marketing arm of Japanese auto maker Mazda Motor Corp.
```

- Before we start processing a piece of text:
  1. Segment text into tokens (words and other symbols)
  2. Group tokens into sentences

Preprocessing

- Texts come in many shapes and forms
- They may need to be
  - digitized using OCR or other techniques
  - separated from images and other content
  - stripped of markup in HTML or XML
- Most NLP tools require plain text (sometimes with markup)

Tokenization

```
J.P. Bolduc, vice chairman of W.R. Grace & Co., which
holds a 83.4% interest in this energy-services company,
was elected a director. Clark J. Vitulli was named senior
vice president and general manager of this U.S. sales and
marketing arm of Japanese auto maker Mazda Motor Corp.
```

- Segment text into tokens
  - Words: chairman, of, energy-services
  - Numerical expressions: 83.4, 10,000
  - Punctuation: ., ?
  - Other symbols: &, %
- Information sources: whitespace, character classes, dictionaries

Normalization

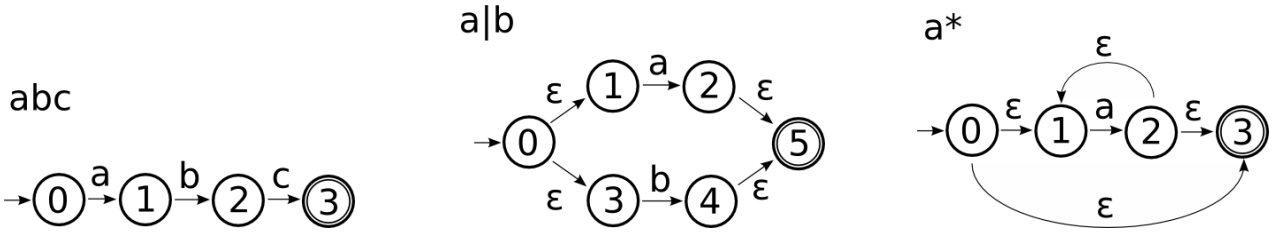
- After tokenization, we sometimes need to "normalize" tokens
  - Abbreviations: U.S., US → U.S.
  - Case folding: Window, window → window
  - Lemmatization: window, windows → window
  - Stemming: compute, computing, computer → comput
- Need for normalization is highly dependent on application
  - Is it always a good idea to lowercase Apple and Windows?
  - When should we regard run and runs as the same word?

Sentence segmentation

```
J.P. Bolduc, vice chairman of W.R. Grace & Co., which
holds a 83.4% interest in this energy-services company,
was elected a director. Clark J. Vitulli was named senior
vice president and general manager of this U.S. sales and
marketing arm of Japanese auto maker Mazda Motor Corp.
```

- Group tokens into sentences
  - Sentence-final punctuation: ., !, ?
  - The period (.) is highly ambiguous: 83.4, U.S., Corp.
  - Final punctuation is often missing (headlines, lists, etc.)

Finite state technology



- Transduction = mapping input sequence to output sequence
- Performed efficiently using finite state automata (FSA, FST)
- Specified using regular expressions (RE)

Coming up next

1. Regular expressions
2. Tokenization

Natural Language Processing

Regular Expressions

Joakim Nivre

Uppsala University
Department of Linguistics and Philology
joakim.nivre@lingfil.uu.se

Regular Expressions

- A formal language for specifying text strings
- REs can be converted into efficient finite state automata

- How do we search for these?

```
raccoon
raccoons
Raccoon
Raccoons
```

Literals

- Most characters represent themselves

Pattern	Matches
raccoon	raccoon
123	123

- Some characters have to be escaped

Pattern	Matches	Example
\.	.	This is a sentence.
\$	$	$100,000

Disjunctions

- Characters inside square brackets `[ ]`

Pattern	Matches
`[Rr]accoon`	`Raccoon`, `raccoon`
`[1234567890]`	Any digit

- Ranges

Pattern	Matches	Example
`[A-Z]`	An uppercase letter	`Charlie Brown`
`[a-z]`	A lowercase letter	`Charlie Brown`
`[0-9]`	A single digit	`Chapter 1`

Character classes

- Ranges work well for English, but what about é, č, ø, Å?
- Often better to use (LOCALE dependent) character classes
- In Python

Pattern	Matches	Closest ASCII equivalent
`\w`	Alphanumeric	`[0-9A-Za-z_]`
`\d`	Digit	`[0-9]`
`\s`	Whitespace	`[ \t\n\r\f\v]`

Negation in disjunction

- Caret ^ means negation only when first in [ ]

Pattern	Matches	Example
[^A-Z]	Not an uppercase letter	R2-D2
[^Ss]	Neither S nor s	R2-D2
[^e^]	Neither e nor ^	R2-D2
a\^b	The string a^b	Look up a^b

More disjunction

- Another word for raccoon is coon
- The pipe | for disjunction

Pattern	Matches
`raccoon|coon`	`raccoon`, `coon`
`football|soccer`	`football`, `soccer`
`a|b|c`	Same as `[abc]`
`[Rr]accoon|[Cc]oon`	`Raccoon`, `raccoon`, `Coon`, `coon`

Quiz 1

- Which strings match the regular expression `abc|xyz`?
  1. `abc`
  2. `ABC`
  3. `acb`
  4. `xyz`
- Which strings match the regular expression `[A-Z][0-9]`?
  1. `B1`
  2. `C`
  3. `4B`
  4. `A4`

Operators

- Operators for optional or repeated patterns
- Often called Kleene operators after Stephen C. Kleene

Pattern	Matches	Explanation
`aa?`	a, aa	0 or 1
`aa*`	a, aa, aaa, aaaa, ...	0 or more
`a+`	a, aa, aaa, aaaa, ...	1 or more
`a.`	aa, ab, ac, ...	any character

Anchors

- Beginning of string ^
- End of string $

Pattern	Examples
^[A-Z]	Charlie, Brown
^[A-Z][a-z]	Raccoon, raccoon
\.$	The end.
.$	The end!, The end?

Example

- Find all instances of the word the in a text

Pattern	Analysis
`the`	Misses capitalized examples
`[Tt]he`	Matches `other` and `theory`
`[^A-Za-z][Tt]he[^A-Za-z]`	Good enough?

Errors

- This process is based on fixing two types of errors:
  - Matching strings we shouldn't match – false positives (Type I)
  - Missing strings we should match – false negatives (Type II)
- Reducing the error rate often involves two antagonistic efforts
  - Improving precision (minimizing false positives)
  - Improving recall (minimizing false negatives)

Greedy left-to-right matching

- Sometimes there are several (overlapping) matches in a string
- By default, the longest match starting from the left wins

Pattern	String	Match
`ab|a|b`	`abc`	`abc` (not `abc` or `abc`)
`port|trait`	`portrait`	`portrait` (not `portrait`)
`\w+`	`cat`	`cat` (not `cat` or `cat`)

Precedence and grouping

- Operators take precedence over concatenation
- Concatenation takes precedence over disjunction
- Brackets can be used to change precedence

Pattern	String	Match
ab*	abab	abab (a + 0 or more b)
(ab)*	abab	abab (0 or more ab)
ac|b	abab	abab (ac or b)
a(c|b)	abab	abab (a + c or b)

Quiz 2

- Regular expression: `a*(a|b)*c`
- String: aabaac

- Which of the following are valid matches?
  1. aabaac
  2. aabaac
  3. aabaac
  4. aabaac
- Which is the longest left-to-right match?
  1. aabaac
  2. aabaac
  3. aabaac
  4. aabaac

Natural Language Processing

Tokenization

Joakim Nivre

Uppsala University
Department of Linguistics and Philology
joakim.nivre@lingfil.uu.se

Tokenization

```
J.P. Bolduc, vice chairman of W.R. Grace & Co., which
holds a 83.4% interest in this energy-services company,
was elected a director. Clark J. Vitulli was named senior
vice president and general manager of this U.S. sales and
marketing arm of Japanese auto maker Mazda Motor Corp.
```

- Segment text into tokens (words and other symbols)

Output formats

- How do we represent the output of tokenization?

Plain text	Text with markup
J.P.	<w>J.P.<\w> <w>Bolduc<\w> <w>,<\w>
Bolduc	<w>vice<\w> <w>chairman<\w> <w>of<\w>
,	<w>W.R.<\w> <w>Grace<\w> <w>&<\w>
vice	<w>Co.<\w> <w>,<\w> <w>which<\w>
chairman	<w>holds<\w> <w>a<\w> <w>83.4<\w>
of	<w>%<\w> <w>interest<\w> <w>in<\w>
...	...

Words

- Strings of alphanumeric characters
- Whitespace bounds greedy left-to-right matching
- Punctuation stripped off automatically
- What about hyphens?

Pattern	String	Matches
`\w+`	`back in 1962`	`back`, `in`, `1962`

Words

- Strings of alphanumeric characters
- Whitespace bounds greedy left-to-right matching
- Punctuation stripped off automatically
- What about hyphens?

Pattern	String	Matches
\w+	back in 1962	back, in, 1962
\w+	“Come on!,” she said.	Come, on, she, said

Words

- Strings of alphanumeric characters
- Whitespace bounds greedy left-to-right matching
- Punctuation stripped off automatically
- What about hyphens?

Pattern	String	Matches
\w+	back in 1962	back, in, 1962
\w+	“Come on!,” she said.	Come, on, she, said
\w+	two-step process	two, step, process

Words

- Strings of alphanumeric characters
- Whitespace bounds greedy left-to-right matching
- Punctuation stripped off automatically
- What about hyphens?

Pattern	String	Matches
\w+	back in 1962	back, in, 1962
\w+	“Come on!,” she said.	Come, on, she, said
\w+	two-step process	two, step, process
\w+(-\w+)*	two-step process	two-step, process

Other symbols

- Punctuation marks can be enumerated
- But some characters should be grouped together
- Logographic symbols can be handled the same way

Pattern	String	Matches
[.,:;!?‘’]	‘‘Come on!,’’ she said.	‘, ‘, !, ,, ’, ’, .

Other symbols

- Punctuation marks can be enumerated
- But some characters should be grouped together
- Logographic symbols can be handled the same way

Pattern	String	Matches
[.,:;!?‘’]	‘‘Come on!,’’ she said.	‘, ‘, !, ,, ’, ’, .
‘‘|’’|[.,:;!?‘’]	‘‘Come on!,’’ she said.	‘‘, !, ,, ’’, .

Other symbols

- Punctuation marks can be enumerated
- But some characters should be grouped together
- Logographic symbols can be handled the same way

Pattern	String	Matches
[.,:;!?‘’]	‘‘Come on!,’’ she said.	‘, ‘, !, ,, ’, ’, .
‘‘|’’|[.,:;!?‘’]	‘‘Come on!,’’ she said.	‘‘, !, ,, ’’, .
[%#$]	$20 (5%)	$, %

Special expressions

- Abbreviations mix alphanumeric characters and punctuation
- Match patterns or specific cases

Pattern	String	Matches
`\w+|[.,:;!?‘’]`	`e.g. Ph.D.`	`e, ., g, ., Ph, ., D, .`

Special expressions

- Abbreviations mix alphanumeric characters and punctuation
- Match patterns or specific cases

Pattern	String	Matches
`\w+|[.,:;!?‘’]`	`e.g. Ph.D.`	`e, ., g, ., Ph, ., D, .`
`e\.g\.|Ph\.D\.`	`e.g. Ph.D.`	`e.g., Ph.D.`

Special expressions

- Abbreviations mix alphanumeric characters and punctuation
- Match patterns or specific cases

Pattern	String	Matches
`\w+|[.,:;!?‘’]`	`e.g. Ph.D.`	`e, ., g, ., Ph, ., D, .`
`e\.g\.|Ph\.D\.`	`e.g. Ph.D.`	`e.g., Ph.D.`
`\w+\.(\w|\.)*`	`e.g. Ph.D.`	`e.g., Ph.D.`

Quiz 1

- Which of the following numerical expressions match the regular expression `\d+[\.,]\d+`?
  1. `314`
  2. `3.14`
  3. `3,143.14`

- Which of the following regular expressions matches the email address `joakim.nivre@lingfil.uu.se`?
  1. `\w+@(\w|\.)+`
  2. `(\w|\.)+@(\w|\.)+`
  3. `\w+(\.\w+)*@\w+\.\w+(\.\w+)*`

Issues in tokenization

- Word boundaries without space (clitics, contractions):
  - English: `don't` = `do` + `n't`; `Sam's` = `Sam` + `'s`
  - Spanish: `al` = `a` + `el`; `dámelo` = `da` + `me` + `lo`
- Words with spaces (multiword expressions):
  - Fixed expressions: `ad hoc`; `by and large`
  - Compounds: `cell phone` (no spaces in many languages)
  - Idioms: `pushing up daisies`

Other writing systems

- Many writing systems do not put spaces between words
- This makes tokenization much more difficult

Thai	Chinese	Japanese
นรชาติวางวาย มลายสิ้นทั้งอินทรีย์ สถิตทั่วแต่ชั่วดี ประดับไว้ในโลกา	床前明月光 疑是地上霜 举头望明月 低头思故乡	水は使用する前に煮沸してください。

我喜欢新西兰花

我喜欢新西兰花

我 | 喜欢 | 新西兰 | 花
"I like New Zealand flowers"

我 | 喜欢 | 新 | 西兰花
"I like fresh broccoli"

Chinese word segmentation

- Words are composed of 1–4 characters
- Some kind of dictionary is needed
- Baseline algorithm: maximum matching

Algorithm: segment string C into word list W using dictionary D

```
while C is not empty:
    find longest match w in D from start of C
    if w is non-empty:
        remove w from C and add to W
    else:
        remove first character from C and add to W
return W
```

Maximum matching in practice

- Baseline accuracy ≈ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	
`thetabledownthere`	

Maximum matching in practice

- Baseline accuracy ≈ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	`the`
`thetabledownthere`	

Maximum matching in practice

- Baseline accuracy $\approx$ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	`the cat`
`thetabledownthere`	

Maximum matching in practice

- Baseline accuracy ≈ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	`the cat in`
`thetabledownthere`	

Maximum matching in practice

- Baseline accuracy ≈ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	`the cat in the`
`thetabledownthere`	

Maximum matching in practice

- Baseline accuracy ≈ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	`the cat in the hat`
`thetabledownthere`	

Maximum matching in practice

- Baseline accuracy ≈ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	`the cat in the hat`
`thetabledownthere`	`theta`

Maximum matching in practice

- Baseline accuracy ≈ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	`the cat in the hat`
`thetabledownthere`	`theta bled`

Maximum matching in practice

- Baseline accuracy ≈ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	`the cat in the hat`
`thetabledownthere`	`theta bled own`

Maximum matching in practice

- Baseline accuracy ≈ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	`the cat in the hat`
`thetabledownthere`	`theta bled own there`

Maximum matching in practice

- Baseline accuracy ≈ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	`the cat in the hat`
`thetabledownthere`	`theta bled own there`
	`the`

Maximum matching in practice

- Baseline accuracy ≈ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	`the cat in the hat`
`thetabledownthere`	`theta bled own there`
	`the table`

Maximum matching in practice

- Baseline accuracy ≈ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	`the cat in the hat`
`thetabledownthere`	`theta bled own there`
	`the table down`

Maximum matching in practice

- Baseline accuracy ≈ 90% boundaries, 80% words
- Does not always work for English

String	Word segmentation
`thecatinthehat`	`the cat in the hat`
`thetabledownthere`	`theta bled own there`
	`the table down there`

Take-home message

- Tokenization is a crucial first step in almost all NLP tasks
- Different tasks may require different tokenizations
- Tokenization is often harder than we think
- Even more so for languages that don't mark word boundaries
- If we get it wrong, we are screwed ("garbage in, garbage out")

Quiz 2

- Consider the dictionary $D = \{$ a, ab, abc $\}$
- What is the max-match segmentation of the string abcba?
  1. abc, ba
  2. abc, b, a
  3. a, b, c, b, a