Words

Sandeep Soni

01/23/2024

Today’s slides draw heavily from David Bamman’s slides on the same topic

Question for the day

"What is the size of a politician's vocabulary?"

Agenda

- What is a word?
- How to segment text into words?
- What are the issues that we should care about?

Words represent concepts

Words represent concepts

Words represent concepts

Words represent concepts

- A concise, shared, and memorable representation for a concept.

Words represent concepts

- A concise, shared, and memorable representation for a concept.
- Words are a form of dimensionality reduction

What are words?

- May the Force be with you
- I'm going to make him an offer he can't refuse.
- And I says, “What kinda action?”
- I’da rather die-died then do that

Word boundaries are not clear-cut in other languages.

Types and tokens

Types and tokens

- Types: Textual representation of real-world concepts
- Tokens: Instances of types

Types and tokens

- Types: Textual representation of real-world concepts
- Tokens: Instances of types

I am as mad as hell and I am not going to take this anymore

Types and tokens

- Types: Textual representation of real-world concepts
- Tokens: Instances of types

> I am as mad as hell and I am not going to take this anymore

- Types = {I, am, as, mad, hell, and, not, going, to, take, this, anymore}
- Tokens = [I, am, as, mad, as, hell, and, I, am, not, going, to, take, this, anymore]

Types and tokens

- Types are the unique tokens in a text
- Types are what we typically use to define a lexicon or a vocabulary.
- How can we measure the verbosity of a text or richness in vocabulary using types and tokens?

How can we segment Text into Tokens?

Just search?

Just search?

- Construct a really large vocabulary and then just search

Just search?

- Construct a really large vocabulary and then just search
- :)
  - Fast: Linear time to tokenize any text
  - Easy: Just lookup in a dictionary

Just search?

- Construct a really large vocabulary and then just search
- :)
  - Fast: Linear time to tokenize any text
  - Easy: Just lookup in a dictionary
- :(
  - Not robust: can match subwords (e.g., the vs them)
  - Brittle: Easy to make up new words (e.g., supercalifragilisticexpialidocious)
  - Unmanageable: Need to store all variations (e.g., move, moves, moving, moved)

Whitespace

```
text.split(" ")
```

It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife.

- Delimiting on whitespace often misses to separate punctuations from words

- `grep -E -o 'wife.?'`

```
853 wife
439 wife,
229 wife.
 93 wife'
 61 wife
 43 wife;
 36 wife?
 28 wife
 27 wife-
 17 wife!
  7 wifey
  7 wifel
  7 wife’
  4 wife_
  2 wifer
  2 wife—
  1 wifeh
  1 wife
```

Sample of ~100 books in PG corpus

Punctuation

- Naively removing punctuation can lead to loss of information about:
  - sentence boundaries
  - pauses
  - parentheticals or asides

Regular expressions

- Instead of simply using whitespace as a marker to separate tokens, we can try to define a pattern for a token
- A pattern like `'wife.?'` is called a regular expression

Regular expressions

- More formally, regular expressions are a compact language to specify search patterns in text

/single/

It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife.

Regular expressions

- More formally, regular expressions are a compact language to specify search patterns in text

/?n/

It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife.

Regular expressions

- More formally, regular expressions are a compact language to specify search patterns in text

`/[Ii]t/`

It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife.

Regular expressions

regex	matches	doesn't match
/the/	the, isothermally	The
/[Tt]he/	the, isothermally, The	
/\b[Tt]he\b/	the, The	—The

Slide credit to David Bamman

Regular expressions

- Use brackets to specify a range or specific alternations

Regex	Range	Example matches
`[Ii]t`	It or it	It, it
`[a-z]`	{a,b,c,…,z}	for, she,hhh
`[0-9]`	{0,1,2,3,…,9}	123, 0123
`[A-Z]`	{A,B,C,…,Z}	FOR, SHE, HHH
`[a-zA-Z0-9]`	{a,b,..,z,A,B,..,Z,0,1,..,9}	1aSBv2

Regular expressions

Term	Meaning	Sample regex	Matches
+	one or more	he+y	hey, heeeeeey
?	optional	colou?r	color, colour
*	zero or more	toys*	toy, toys, toysss

Slide credit to David Bamman

Symbols

Symbol	Function
\b	Word boundary (zero width)
\d	Any decimal digit (equivalent to [0-9])
\D	Any non-digit character (equivalent to [^0-9])
\s	Any whitespace character (equivalent to [\t\n\r\f\v])
\S	Any non-whitespace character (equivalent to [^ \t\n\r\f\v])
\w	Any alphanumeric character (equivalent to [a-zA-Z0-9_])
\W	Any non-alphanumeric character (equivalent to [^a-zA-Z0-9_])
\t	The tab character
\n	The newline character

https://www.nltk.org/book/ch03.html

Slide credit to David Bamman

Composition of Regexes

- You can create complex regular expressions from simple ones by using disjunction

`/single/|[Ii]t`

It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife.

Python

- You can search via regex with the help of the `re` module
- re.findall (regex, text) will find non-overlapping matches
- `re.findall ("[Ii]t", "If it is meant to be it is meant to be")`
- The function returns a list of all the matches: `[It, it]`
- Also see `re.match`, `re.compile`
- https://www.dataquest.io/wp-content/uploads/2019/03/python-regular-expressions-cheat-sheet.pdf

Back to tokenization

https://spacy.io/usage/spacy-101#annotations-token

Sentence segmentation

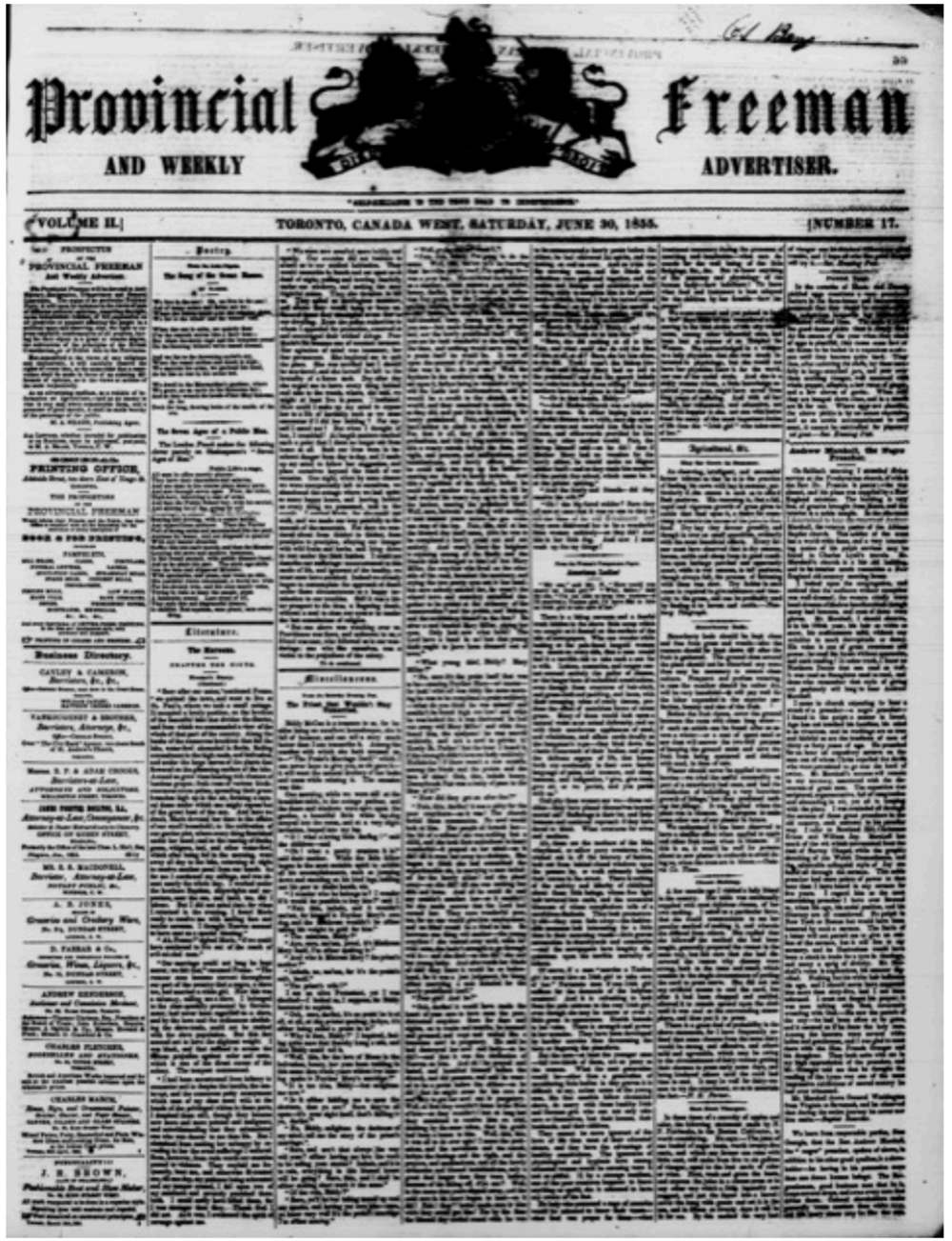
- This is usually not done using regular expressions
- He lives in the U.S. John, however, lives in Canada.
- Approaches include unsupervised methods to smartly detect sentence-initial or sentence-end words or use dependency parsing to find when sentences end

Stemming and Lemmatization

- Languages have rich inflectional and derivational morphology
- Stemming is a heuristic to clip the suffixes
  - arguing, argues, argue, argued -> argu
- Lemmatization is more principled as it collapses inflectional forms to a typical dictionary entry
  - arguing, argues, argue, argued -> argue

Post processing

- What counts as tokens can have important consequences
- In historical texts, tokenization errors need to be fixed as a post-processing step
- `senatoradmits -> senator admits`


Provincial Freeman
AND WEEKLY ADVERTISER.
VOLUME II.] TORONTO, CANADA WEST, SATURDAY, JUNE 30, 1855. [NUMBER 17.

Soni et. al. 2019

In class

- Tokenization demo