Text Normalization and Morphology

23 February 2021
Background

For today, you should have read

the NLTK book, chapter 1 and

*Speech and Language Processing*, 2nd ed., chapter 3 through § 3.7.
Exercise 1

Due Thursday, start of class

Assignment 1

Due March 9, 11:59 p.m. EST
Text normalization
Every NLP task requires text normalization:

- Tokenizing (segmenting) words
- Normalizing word forms
- Segmenting sentences
Tokenization
A significant first step in morphological analysis is *tokenization* – segmenting text into individual words.
When we talk about a “word”, we might mean
an individual *occurrence* of a word or
an abstract *vocabulary item*.

To avoid confusion, we use more precise terminology:

*Word token*: an occurrence of a word

*Word type*: a vocabulary item

The sentence *my dog likes his dog* contains five word tokens, but four word types.
How many words?

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

How many?

15 word tokens (or 14)
13 word types (or 12) (or 11?)
How many words?

Seuss’s *cat in the hat* is different from other *cats*!

*Lemma*: same stem, part of speech, rough word sense

*cat* and *cats* have the same lemma

*Wordform*: the full inflected surface form

*cat* and *cats* have different wordforms
Demo: NLTK word types and tokens
Why not just use white space?

Examples like

> Mr. Sherwood said reaction to Sea Containers' proposal has been “very positive.” In New York Stock Exchange composite trading yesterday, Sea Containers closed at $62.625, up 62.5 cents.

or

> “I said, ‘what’re you? Crazy?’” said Sadowsky. “I can’t afford to do that.”

lead to tokens like:

> cents.
> said,
> positive.”
> Crazy?’”
We also can’t just segment on punctuation:

Word-internal punctuation

- m.p.h
- Ph.D.
- AT&T
- 01/02/06
- Google.com
- 555,500.50
There are more complicated questions of what to count as a word when we tokenize.

E.g., do we expand clitics?

*What’re* → *what are*

*I’m* → *I am*

Many tokenizers don’t expand clitics

Result is arbitrary practices!

*[ca] [n’t] vs [can] [’t]*
What about multi-token words?

New York

Rock ’n’ roll

We can combine them:

New York = [New York]

Thus tied up with the problem of *named entity recognition*.

Wake up, work out

But: I couldn’t work the answer out

Again, varying practice

What about *New York–based*?
Tokenization: Implementation

Tokenization needs to be run before any other language processing.

It needs to be very fast!

Standard method: Use deterministic algorithms based on regular expressions, compiled into very efficient finite-state automata.
Regular expression tokenization in NLTK

```python
>>> text = "That U.S.A. poster-print costs $12.40..."
>>> pattern = r''''(?x) # set flag to allow verbose regexps
... ([A-Z]\.)+ # abbreviations, e.g., U.S.A.
... | \w+(-\w+)* # words with optional internal hyphens
... | \$?\d+(\.\d+)?%? # currency and percentages, e.g. $12.40, 82%
... | \.\.\. # ellipsis
... | \[[.,;"'?():_-_'] # these are separate tokens; includes ], [
... '''
>>> nltk.regexp_tokenize(text, pattern)
['That', 'U.S.A.', 'poster-print', 'costs', '$12.40', '...']
```

From NLTK book, Chapter 3
Tokenization: language issues
Tokenization: language issues

French

$L$'ensemble $\rightarrow$ one token or two?

$L, L', Le?$

Want *l’ensemble* to match with *un ensemble*

Until at least 2003, it didn’t on Google
Tokenization: language issues

French

*L’ensemble* → one token or two?

*L? L’? Le?*

Want *l’ensemble* to match with *un ensemble*

Until at least 2003, it didn’t on Google

German noun compounds are not segmented

*Lebensversicherungsgesellschaftsangestellter*

“life insurance company employee”
Tokenization: language issues

Chinese, Japanese, and Thai have no spaces between words, e.g.,

莎拉波娃现在居住在美国东南部的佛罗里达。
Not always guaranteed a unique tokenization

Further complicated in Japanese, with multiple alphabets intermingled

Dates/amounts in multiple formats

フォーチュン500社は情報不足のため時間あたり$500K（約6,000万円）
Tokenization: language issues

Arabic (or Hebrew) is basically written right to left, but with certain items like numbers written left to right.

Words are separated, but letter forms within a word form complex ligatures.

الجزائر ف-read version 1962 بعد 132 عام من الاحتلال الفرنسي.
← → ← → ← start

“Algeria achieved its independence in 1962 after 132 years of French occupation.”
Word normalization
Sometimes we see *U.S.A.* and other times *USA*.

They’re the same word, right?
Text processing is easier if words are in some standard form.

E.g., in information retrieval, the words in the query and in the index of documents need to match!

We implicitly define equivalence classes of terms, e.g., deleting periods in a term.
Case folding

For information retrieval, we often reduce all letters to lower case.

Because users tend to use lower case
Possible exception: upper case in mid-sentence?
   e.g., General Motors
   Fed vs fed

For sentiment analysis, machine translation, and information extraction, case is helpful (US versus us).
Alternative: asymmetric expansion:

Enter: *window*  Search: *window*, *windows*

Enter: *windows*  Search: *Windows*, *windows*, *window*

Enter: *Windows*  Search: *Windows*

Potentially more powerful, but less efficient
One possibility for further word normalization is **stemming** – chopping off a word’s affixes, e.g.,

*automate, automates, automatic, automation* all reduce to *automat*
One possibility for further word normalization is **stemming** – chopping off a word’s affixes, e.g.,

*automate, automates, automatic, automation* all reduce to *automat*

---

*for example compressed and compression are both accepted as equivalent to compress.*
One possibility for further word normalization is **stemming** – chopping off a word’s affixes, e.g.,

*automate, automates, automatic, automation* all reduce to *automat*

*for example compressed and compression are both accepted as equivalent to compress.*
Porter stemmer

One of the mostly widely used stemming algorithms is the simple and efficient Porter (1980) algorithm, which is based on a series of simple cascaded rewrite rules.

- ATIONAL → ATE (e.g., relational → relate)
- ING → ε if stem contains vowel (e.g., motoring → motor)
- SSES → SS (e.g., grasses → grass)

Problem:

Not perfect: errors of commission, omission
Lemmatization

A nicer – but harder – alternative is to represent all words as their shared root (dictionary headword form).

\[\text{am, are, is} \rightarrow \text{be}\]
\[\text{car, cars, car’s, cars’} \rightarrow \text{car}\]

Spanish \textit{quiero} ("I want"), \textit{quieres} ("you want") same lemma as \textit{querer} ("want")
Lemmatization

the boy’s cars are different colors
→ the boy car be different color

He is reading detective stories
→ he be read detective story
Lemmatization is done by morphological parsing.

We’ll come back to that in a moment.
Sentence segmentation
Why do we want to divide text into sentences?
Why do sentences matter?
! and ? are relatively unambiguous sentence boundaries, but a period “.” is quite ambiguous:

It could be a sentence boundary

Or it could be part of an abbreviations like Inc. or Dr.

Other problems:

“You reminded me,” she remarked, “of its importance.”

“I understand. You reminded me,” she added, “of its importance.”

Nested sentences!
Build a binary classifier:

Looks at a “.”

Decides *EndOfSentence/NotEndOfSentence*

Classifiers: hand-written rules, regular expressions, or machine-learning
Determining if a word is end-of-sentence: a decision tree

Lots of blank lines after me?

YES

Final punctuation is ?, !, or ?:

YES

Final punctuation is period

NO

E-O-S

I am “etc” or other abbreviation

YES

Not E-O-S

NO

E-O-S

Not E-O-S
More sophisticated decision tree features
More sophisticated decision tree features

Case of word with “.”: Upper, Lower, Cap, Number

Dr., it., NASA., 12.
More sophisticated decision tree features

Case of word with “.”: Upper, Lower, Cap, Number

  Dr., it., NASA., 12.

Case of word after “.”: Upper, Lower, Cap, Number

  Dr. Seuss, U.S. economy, etc.
More sophisticated decision tree features

Case of word with “.”: Upper, Lower, Cap, Number

Dr., it., NASA., 12.

Case of word after “.”: Upper, Lower, Cap, Number

Dr. Seuss, U.S. economy, etc.

Numeric features

Length of word with “.”
Probability(word with “.” occurs at end-of-s)
Probability(word after “.” occurs at beginning-of-s)
Implementing decision trees

A decision tree is just an if–then–else statement
Implementing decision trees

A decision tree is just an if–then–else statement

The interesting research is choosing the features
Implementing decision trees

A decision tree is just an if–then–else statement

The interesting research is choosing the features

Setting up the structure is often too hard to do by hand
Implementing decision trees

A decision tree is just an if–then–else statement

The interesting research is choosing the features

Setting up the structure is often too hard to do by hand

  Hand-building only possible for very simple features, domains
Implementing decision trees

A decision tree is just an if–then–else statement

The interesting research is choosing the features

Setting up the structure is often too hard to do by hand

Hand-building only possible for very simple features, domains

For numeric features, it's too hard to pick each threshold
Implementing decision trees

A decision tree is just an if–then–else statement

The interesting research is choosing the features

Setting up the structure is often too hard to do by hand

Hand-building only possible for very simple features, domains

For numeric features, it’s too hard to pick each threshold

Instead, structure usually learned by machine learning from a training corpus
Background: Morphology
Morphology is the study of the ways that words are built up from smaller meaningful units called morphemes.
Example: *Unladylike*

Three morphemes and four syllables

Morpheme breaks:

- *un-* “not"
- *lady* “(well behaved) female adult human”
- *-like* “having the characteristics of”

None of these morphemes can be broken more without losing all sense of meaning

- *la* and *dy* are separate syllables, but have no meaning on their own
Example: *Dogs*

Two morphemes and one syllable:

- *dog*, and
- *-s*, a plural marker on nouns

A morpheme like *-s* can be a single phoneme, does not have to be a whole syllable
Example: *Technique*

One morpheme and two syllables.

Cannot be broken down into smaller meaningful parts.
There are two classes of morphemes:

**Stems**: The core, meaning-bearing units

**Affixes**: Bits and pieces that adhere to stems to change their meanings and grammatical functions

Prefixes

Suffixes

Infixes

Circumfixes
The way that stems and affixes combine is based to a large degree on the *word class* of the stem.

By word class, we have in mind familiar notions like noun and verb. Gory details later!
There are two broad classes of morphology:

*Inflectional*

*Derivational*
Inflectional morphology
**Inflectional morphology** adds tense, number, person, mood, aspect.

The resulting word serves a new grammatical role, but *the word class doesn’t change*.

Examples:

- *come* is inflected for person and number:
  
  *The pizza guy comes at noon.*

- *les* and *rouges* are inflected for agreement with *pommes* in grammatical gender
  
  *Les pommes rouges* ("the red apples") vs
  *La pomme rouge* ("the red apple")
In English, only nouns, verbs, and sometimes adjectives can be inflected, and the number of affixes is quite small.

Inflections of nouns in English:

An affix marking **plural**:

- *cat*(-s), *thrush*(-es), *ox* (oxen), *mouse* (mice)
- *ibis*(-es), *waltz*(-es), *finch*(-es), *box*(-es), *butterfly*(-ies)

An affix marking **possessive**:

- *llama’s*, *children’s*, *llamas’,* *Euripides’* comedies
In English, verbal inflection is more complicated than nominal inflection.

English has three kinds of verbs:

*Main verbs: eat, sleep, impeach*
*Modal verbs: can, will, should*
*Primary verbs: be, have, do*
### Regular (English) verbs

<table>
<thead>
<tr>
<th>Morphological form classes</th>
<th>Regularly inflected verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stem</td>
<td>walk</td>
</tr>
<tr>
<td>-s form</td>
<td>walks</td>
</tr>
<tr>
<td>-ing form</td>
<td>walking</td>
</tr>
<tr>
<td>Past form or -ed participle</td>
<td>walked</td>
</tr>
</tbody>
</table>

These regular verbs and forms are significant in the morphology of English because of their *majority* and being *productive*. 
# Irregular (English) verbs

<table>
<thead>
<tr>
<th>Morphological form classes</th>
<th>Irregularly inflected verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stem</td>
<td>eat</td>
</tr>
<tr>
<td>-s form</td>
<td>eats</td>
</tr>
<tr>
<td>-ing form</td>
<td>eating</td>
</tr>
<tr>
<td>Past form</td>
<td>ate</td>
</tr>
<tr>
<td>-ed participle</td>
<td>eaten</td>
</tr>
</tbody>
</table>
**“To love” in French**

<table>
<thead>
<tr>
<th>Present</th>
<th>Future</th>
<th>Imperfect</th>
<th>Present participle</th>
</tr>
</thead>
<tbody>
<tr>
<td>j'</td>
<td>aime</td>
<td>aimerai</td>
<td>aimaist</td>
</tr>
<tr>
<td>tu</td>
<td>aimes</td>
<td>aimeras</td>
<td>aimaist</td>
</tr>
<tr>
<td>il</td>
<td>aime</td>
<td>aimerait</td>
<td>aimait</td>
</tr>
<tr>
<td>nous</td>
<td>aimons</td>
<td>aimerons</td>
<td>aimions</td>
</tr>
<tr>
<td>vous</td>
<td>aimez</td>
<td>aimeriez</td>
<td>aimiez</td>
</tr>
<tr>
<td>ils</td>
<td>aiment</td>
<td>aimeront</td>
<td>aimaient</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subjunctive</th>
<th>Conditional</th>
<th>Passé simple</th>
<th>Imperfect subjunctive</th>
</tr>
</thead>
<tbody>
<tr>
<td>j'</td>
<td>aime</td>
<td>aimerait</td>
<td>aimai</td>
</tr>
<tr>
<td>tu</td>
<td>aimes</td>
<td>aimerais</td>
<td>aimas</td>
</tr>
<tr>
<td>il</td>
<td>aime</td>
<td>aimerait</td>
<td>aima</td>
</tr>
<tr>
<td>nous</td>
<td>aimons</td>
<td>aimerions</td>
<td>aimames</td>
</tr>
<tr>
<td>vous</td>
<td>aimez</td>
<td>aimeriez</td>
<td>aimâtes</td>
</tr>
<tr>
<td>ils</td>
<td>aiment</td>
<td>aimeraient</td>
<td>aimèrent</td>
</tr>
</tbody>
</table>

| Imperative | | |
| (tu)       | aime        | |
| (nous)     | aimons      | |
| (vous)     | aimez       | |

Most languages are more inflectionally complicated than English.
Acknowledgments

The lecture incorporates material from:

Nancy Ide, Vassar College
Daniel Jurafsky and James Martin, *Speech and Language Processing*