Basics of Text Processing

✧ Words
✧ Tokenization
✧ Social signals
✧ Resources to dealing with text

Some slides are adapted based on *Text Processing from Speech and Language Processing* (3rd ed. draft) Dan Jurafsky and James H. Martin (https://web.stanford.edu/~jurafsky/slp3/)
I like the San Francisco airport
ttyl, lol
This no there is no typo
Punctuation

Punctuation can be important
  Signals boundaries (sentence, clausal boundaries, etc)
  Has illocutionary force, like exclamation points (!) and question marks (?)
Emoticons are strong signals of sentiment
How many words in a sentence?

"I do uh main- mainly business data processing"

"Emily’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
How many words in a sentence?

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

Type: an element of the vocabulary.
Token: an instance of that type in running text.

How many?
  15 tokens (or 14)
  13 types (or 12) (or 11?)
How many words in a corpus?

\[ N = \text{number of tokens} \]
\[ V = \text{vocabulary} = \text{set of types}, |V| \text{ is size of vocabulary} \]

| Corpus                  | Tokens=\(N\)       | Types=|\(V|\)      |
|-------------------------|---------------------|--------|
| Switchboard phone conversation | 2.4 million         | 20 thousand |
| Shakespeare             | 884,000             | 31 thousand |
| COCA                    | 440 million         | 2 million |
| Google N-grams          | 1 trillion          | 13+ million |
I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!
Bag of Words Representation

A fixed-length representation, consisting of a **vector of word counts**
The vector length is the size of the vocabulary

\[
\mathbf{w} = \text{It was the best of times, it was the worst of times}
\]

\[
\mathbf{x} = [\text{aardvark}, \ldots, \text{best}, \ldots, \text{it}, \ldots, \text{of}, \ldots, \text{zyther}]
\]
Unigrams/Bigrams/Trigrams

*It was the best of times, it was the worst of times.*

Unigrams: it, was, the, best, ...
Bigrams: it was, was the, the best, best of, ...
Trigrams: it was the, was the best, the best of, ...
Corpora

Words don't appear out of nowhere!

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 vary along dimensions like

**Language** 7097 languages in the world

**Variety** like African American Language varieties.

AAE Twitter posts might include forms like "iont" (I don’t)

**Code switching**, e.g., Spanish/English, Hindi/English:

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, SES
Corpus Datasheets (Gebru et al (2020), Bender and Friedman (2018))

Motivation:
   Why was the corpus collected?
   By whom?
   Who funded it?

Situation: In what situation was the text written?

Collection process: If it is a subsample how was it sampled? Was there consent? Pre-processing?

Annotation process, language variety, demographics, etc.
Text Normalization

Every NLP task requires text normalization:

- Tokenizing (segmenting) words
- Normalizing word formats
- Segmenting sentences
Space-based tokenization

A very simple way to tokenize

For languages that use space characters between words

   Arabic, Cyrillic, Greek, Latin, etc., based writing systems

Segment off a token between instances of spaces
Issues in Tokenization

Can't just blindly remove punctuation:
  m.p.h., Ph.D., AT&T, cap’n
  prices ($45.55)
  dates (01/02/06)
  URLs (http://stanford.edu)
  hashtags (#nlproc)
  email addresses (someone@stanford.edu)

Clitic: a word that doesn't stand on its own
  "are" in we're, French "je" in j'ai, "le" in l'honneur

When should multiword expressions (MWE) be words?
  New York, rock ‘n’ roll
Tokenization using 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', '...']
```
Tokenization in languages without spaces

Many languages (like Chinese, Japanese, Thai) don't use spaces to separate words!

How do we decide where the token boundaries should be?
Word Normalization

Putting words/tokens in a standard format

U.S.A. or USA
uhhuh or uh-huh
Fed or fed
am, is, be, are
Case Folding

Applications like IR: reduce all letters to lower case
Since users tend to use lower case
Possible exception: upper case in mid-sentence?
   e.g., General Motors
   Fed vs. fed
   SAIL vs. sail

For sentiment analysis, MT, Information extraction
   Case is helpful (US versus us is important)
Sentence Segmentation

!, ? mostly unambiguous but period “.” is very ambiguous

Sentence boundary
Abbreviations like Inc. or Dr.
Numbers like .02% or 4.3

Common algorithm: Tokenize first: use rules or ML to classify a period as either (a) part of the word or (b) a sentence-boundary.

An abbreviation dictionary can help

Sentence segmentation can then often be done by rules based on this tokenization.
Get to know your data!

✧ What’s the format?
✧ In which language?
✧ What’s the genre?
✧ Where does it come from?
✧ Can I trust the data source?
Dealing with Social Media Data

Emojis: 🙏🤷🥰

Special characters: ' }{ [ ] # @ ! * < > ~

Out of vocabulary words: icebucketchallenge, wowwwwww

URLs: https://www.nytimes.com/

Typos or spelling errors: typs, tpos, ...

Social media features: @user, RT, #hashtags

Slang words: chill, slay, sick ...
Multilinguality in Social Data

Language identification has very high accuracy for long texts, but struggles with social media (short informal) text

Code switching:  I have 2 friends *due estudiaron la contabilidad*
Data Preprocessing Matters

To remove or not to remove, that’s up to your goal

A brief review to a movie:

Very interesting, 😊

Very interesting, 😞
Emojis Might Help Prediction

<table>
<thead>
<tr>
<th>Irony?</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>I just love being ignored😊 #not 😓</td>
<td>Yes</td>
</tr>
<tr>
<td>Love it when my mans on a cleaning spree.. Saves me doing it 👏😢</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sentiment</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>@Paul O'Connor187 hi we going to see ted 2 at the Odeon cinemas at Glasgow on Wednesday 😊</td>
<td>Positive</td>
</tr>
<tr>
<td>Serato DJ isn’t compatible with Windows 10 yet 😞 ...got to spin on my old laptop Saturday.</td>
<td>Negative</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Avg. Rec.</th>
<th>Acc.</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baziots et al. (2017)</td>
<td>0.681</td>
<td>0.651</td>
<td>0.677</td>
</tr>
<tr>
<td>Cliche (2017)</td>
<td>0.681</td>
<td>0.658</td>
<td>0.685</td>
</tr>
<tr>
<td>Rouvier (2017)</td>
<td>0.676</td>
<td>0.661</td>
<td>0.674</td>
</tr>
<tr>
<td>EMJ-EMBED</td>
<td>0.703</td>
<td>0.689</td>
<td>0.691</td>
</tr>
<tr>
<td>EMJ-DESC</td>
<td><strong>0.728</strong></td>
<td><strong>0.704</strong></td>
<td><strong>0.703</strong></td>
</tr>
</tbody>
</table>

Results on sentiment analysis (three-way classification: positive, neutral or negative)

Resources to Check Out!

- NLTK
- Beautiful Soup
- SpaCy
- Spacymoji
- https://github.com/steve-wilson/nlpcss201-sm-preprocessing