# Разбиение на токены / Word tokenization.

**Speech and Language Processing (3rd ed. raft), Dan Jurafsky and James H. Martin**
**Глава 2.3, стр. 11**

Выполнила: Буллиева Дарья
17.03.2017

# Text Normalization

- Every NLP task needs to do text normalization:
  1. Segmenting/tokenizing words in running text
  2. Normalizing word formats
  3. Segmenting sentences in running text

Для чего необходимо решение задач 1-3?

# How many words?

- 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

- Рыбак рыбака видит издалека.
  - Рыбак и рыбака — одна лемма, но разные словоформы.

В чем отличие леммы от словоформы?

# How many words?

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?)

Чем отличается модель от токена?

- Он не мог не ответить на это письмо.

Сколько моделей и токенов?

# How many words?

***N*** = number of tokens

***V*** = vocabulary = set of types

$|V|$ is the size of the vocabulary

Church and Gale (1990): $|V| > O(N^{½})$

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

# Simple Tokenization in UNIX

- (Inspired by Ken Church's UNIX for Poets.)
- Given a text file, output the word tokens and their frequencies

```
tr -sc 'A-Za-z' '\n' < shakes.txt
      | sort
      | uniq -c
```

Change all non-alpha to newlines

Sort in alphabetical order

Merge and count each type

```
1945 A
  72 AARON
  19 ABBESS
   5 ABBOT
 ... ...
```

```
25 Aaron
 6 Abate
 1 Abates
 5 Abbess
 6 Abbey
 3 Abbot
... ...
```

Какие команды UNIX можно использовать для обработки текста?

# The first step: tokenizing

```
tr -sc 'A-Za-z' '\n' <
shakes.txt | head
```

```
THE
SONNETS
by
William
Shakespeare
From
fairest
creatures
We
...
```

Что произошло в результате выполнения команды?

# The second step: sorting

```
tr -sc 'A-Za-z' '\n' <
shakes.txt | sort | head
```

```
A
A
A
A
A
A
A
A
A
...
```

10

Что вывелось в результате выполнения команды?

# More counting

- Merging upper and lower case

```
tr 'A-Z' 'a-z' < shakes.txt | tr -sc 'A-Za-z' '\n' | sort | uniq -c
```

- Sorting the counts

```
tr 'A-Z' 'a-z' < shakes.txt | tr -sc 'A-Za-z' '\n' | sort | uniq -c | sort -n -r
```

```
23243 the
22225 i
18618 and
16339 to
15687 of
12780 a
12163 you
10839 my
10005 in
8954  d
```

Почему “d” вывелось как отдельное слово?

What happened here?

# Issues in Tokenization

- Finland's capital → Finland Finlands Finland's ?
- what're, I'm, isn't → What are, I am, is not
- Hewlett-Packard → Hewlett Packard ?
- state-of-the-art → state of the art ?
- Lowercase → lower-case lowercase lower case ?
- San Francisco → one token or two?
- m.p.h., PhD. → ??
- Красно-желтый → Красно желтый? Красно-желтый?

В чем заключается проблема токенизации?

# Tokenization: language issues

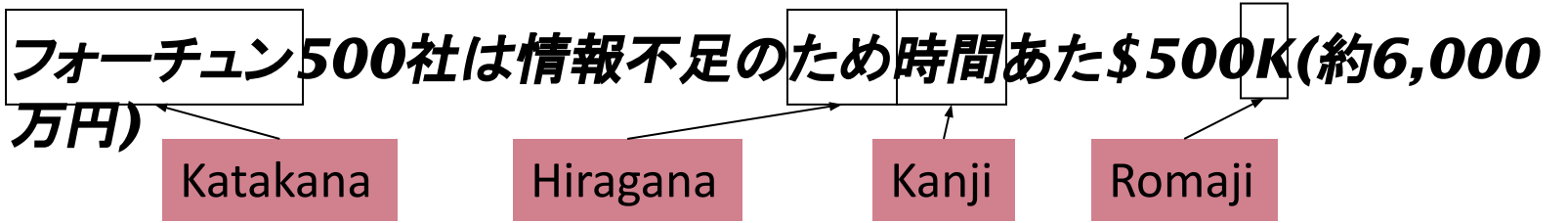
- French
  - ***L'ensemble*** → one token or two?
    - ***L*** ? ***L'*** ? ***Le*** ?
    - Want ***l'ensemble*** to match with ***un ensemble***

- German noun compounds are not segmented
  - ***Lebensversicherungsgesellschaftsangestellter***
  - 'life insurance company employee'
  - German information retrieval needs **compound splitter**

Какие проблемы, связанные с особенностями языков, могут возникнуть?

# Tokenization: language issues

- Chinese and Japanese no spaces between words:
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗
  - Sharapova now lives in US southeastern Florida
- Further complicated in Japanese, with multiple alphabets intermingled
  - Dates/amounts in multiple formats




End-user can express query entirely in hiragana!

# Какие особенности японского языка еще больше осложняют обработку текста?

# Word Tokenization in Chinese

- Also called **Word Segmentation**
- Chinese words are composed of characters
  - Characters are generally 1 syllable and 1 morpheme.
  - Average word is 2.4 characters long.
- Standard baseline segmentation algorithm:
  - Maximum Matching  (also called Greedy)

Какой алгоритм применяется для токенизации  в китайском языке?

# Maximum Matching Word Segmentation Algorithm

- Given a wordlist of Chinese, and a string.

1) Start a pointer at the beginning of the string
2) Find the longest word in dictionary that matches the string starting at pointer
3) Move the pointer over the word in string
4) Go to 2

В чем заключается суть алгоритма Maximum Matching?

# Max-match segmentation illustration

- Thecatinthehat — the cat in the hat
- Thetabledownthere — the table down there
  theta bled own there
- Doesn’t generally work in English!

- But works astonishingly well in Chinese
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
- Modern probabilistic segmentation algorithms even better

# Интересные статьи:

- http://www.dialog-21.ru/digests/dialog2012/materials/pdf/68.pdf
- http://www.dialog-21.ru/media/2213/muravyev.pdf