Tokenization

"Token": a unit of text, considered indivisible by the algorithm

Could be a:

- letter
- word
- phrase?
- morpheme?

See also: segmentation

Motivating application

Sentiment analysis: How did the author of a document feel? *Good* or *bad*?
(binary classification)

Algorithm:

- count the frequency of the tokens in *good* documents and *bad* documents
    - e.g. *good*: 75% "woo", 25% "oow"; *bad*: 25% "woo", 75% "oow"
- count the frequency of the tokens in a candidate document
    - 80% "woo", 20% "oow"
- compare
    - candidate token proportions are closer to *good*

Considerations

We want tokens to:

- bear information
    - Models can be simpler if tokens contain more information
    - Longer tokens bear more information
- occur frequently
    - We can train models if tokens occur frequently
    - Longer tokens occur less frequently

English uses ~26 letters, ~100,000 words, ~5 letters per word

Letter-based tokenization

Pros:

- tokens occur frequently
- easy to compute

Cons:

- very little information per token

Word-based tokenization

Pros:

- good tradeoff between information and frequency

Cons:

- harder to compute
- larger vocabulary

Simple example

Input:

"John does not like dogs"

Desired output:

"John", "does", "not", "like", "dogs"

Simple example, solution

Input:

"John does not like dogs"

Split on `\s+`.

Output:

"John", "does", "not", "like", "dogs"

More troublesome example

Input:

"John Doe has cynophobia, i.e. he doesn't like dogs."

Desired output:

"John Doe", "has", "cynophobia", ",", "i.e.", "he", "does", "n't", "like", "dogs", "."

More troublesome example, solution

Input:

"John Doe has cynophobia, i.e. he doesn't like dogs."

```
segment sentences
detect contractions
detect acronyms
...
```

Output:

"John Doe", "has", "cynophobia", ",", "i.e.", "he", "does", "n't", "like", "dogs", "."

References

- NLTK tokenizers
- Penn Treebank tokenizer
    - Python
    - Javascript
- Stanford tokenizer

Empirical tokenization

Where do heuristic tokenizers fail?

- named entities
  - Will Smith
  - Blue Origin
- lemmas
  - talk, talked, talking
- sub-word translation
  - solar system (English)/Sonnensystem (German)

Some common phrases and morphemes may be good tokens (high-frequency, high-information).

Byte-pair encoding (BPE)

- Developed for general compression
- Iteratively replaces the most frequent pair of symbols with a new combined symbol
- For NLP, instead combine most frequent pair of tokens into a combined token
- Stop when the desired number of merges is completed or a maximum vocabulary size is reached
    - These parameters can be configured

BPE References

- Gage 1994
- Sennrich et al. 2015