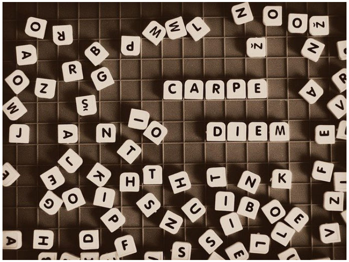
Advanced NN Architectures

Tokenization



Learning goals

- Understand the the process of text tokenization
- Learn the various types of text tokenization

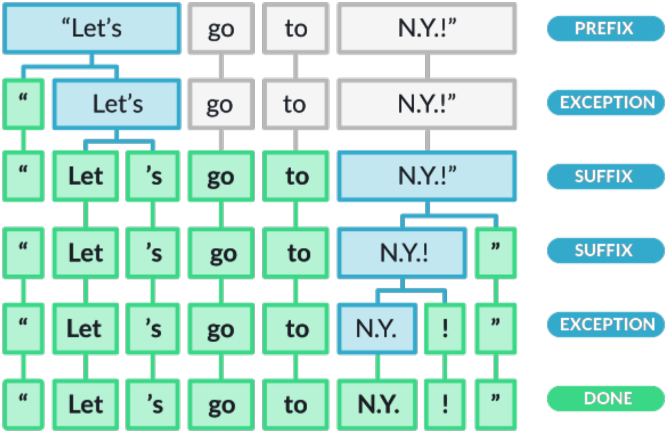
PROCESS OF TEXT TOKENIZATION

- Breaking text into smaller units called tokens
  - Tokens are discrete text units (letters, words, etc.)
  - They are the building blocks of natural language
- Encoding each token with unique IDs (numbers)
- Performed on the entire corpus of documents
  - Corpus vocabulary of unique tokens is obtained
- Mandatory preprocessing step for most of NLP tasks

WHY TOKENIZE?

- Computers must understand text
  - Text encoding is necessary
  - Encode small rather than large units
- Corpus documents can be large and hard to interpret
  - Working with tokens is easier
  - Building meaning in bottom-up fashion
- Text may contain extra whitespaces
  - Tokenization removes them

TOKENIZATION IN ACTION



Source: *spaCy*

TOKENIZATION TYPES

- Paragraph tokenization
  - Breaking doucments in paragraphs
  - Rarely used
- Sentence tokenization
  - Breaking text in sentences
- Word tokenization
  - Breaking text in words
  - The most common
- Subword tokenization
  - Breaking words in morphemes
- Character tokenization
  - Breaking text in individual characters
- Whitespace tokenization
  - Typical whitespaces: `" "`, `\t`, `\n`

WORD TOKENIZATION

- Most popular type of tokenization
  - Applied as preprocessing step in most NLP tasks
- Considers dictionary words and several delimiters
  - Accuracy depends on dictionary used for training
  - Tradeoff between accuracy and efficiency
- Whitespaces and punctuation symbols are used
  - They determine word boundaries
- Available in many NLP libraries

Example:

What is the tallest building? => ‘What’, ‘is’, ‘the’, ‘tallest’, ‘building’, ‘?’

SUBWORD TOKENIZATION

- Finer grained than word tokenization
  - Breaks text into words
  - Breaks words into smaller units (root, prefix, suffix, etc.)
  - Uses more complex linguistic rules
- More important for highly flective languages
  - Words have many forms
  - Prefixes and suffixes are added
  - Word meaning and function changes
- Helps to disambiguate meaning
- Helps to reduce out of vocabulary words

Example:

What is the tallest building? => ‘What’, ‘is’, ‘the’, ‘tall’, ‘est’, ‘build’, ‘ing’, ‘?’

CHARACTER TOKENIZATION

- Creates smaller vocabulary
  - Same as the number of lettters
- Helps with out of vocabulary words
  - Retains their character composition
- More complex process
  - The output becomes 5 or 6 times bigger

Example:

What is the tallest building? => ‘W’, ‘h’, ‘a’, ‘t’, ‘i’, ‘s’, ‘t’, ‘h’, ‘e’, ‘t’, ‘a’, ‘l’, ‘l’, ‘e’, ‘s’, ‘t’, ‘b’, ‘u’, ‘i’, ‘l’, ‘d’, ‘i’, ‘n’, ‘g’, ‘?’