Natural Language Processing

Last update: 20 September 2024

Academic Year 2024 – 2025

Alma Mater Studiorum · University of Bologna

Contents

1 Basic text processing

Text normalization Operations such as:

Tokenization Split a sentence in tokens. Tokenization

| **Remark.** Depending on the approach, a token is not always a word.

Lemmatization/stemming Convert words to their canonical form. Lemmatization/stemming

| **Example.** $\{\texttt{sang}, \texttt{sung}, \texttt{sings}\} \mapsto \texttt{sing}$

Sentence segmentation Split a text in sentences. Sentence segmentation

| **Remark.** A period does not always signal the end of a sentence.

1.1 Regular expressions

Regular expression (regex) Formal language to describe string patterns. Regular expression (regex)

1.1.1 Basic operators

Disjunction (brackets) Match a single character between square brackets `[]`.

| **Example.** `/[wW]oodchuck/` matches `Woodchuck` and `woodchuck`.

Range Match a single character from a range of characters or digits.

| **Example.**
- `/[A-Z]/` matches a single upper case letter.
- `/[a-z]/` matches a single lower case letter.
- `/[0-9]/` matches a single digit.

Negation Match the negation of a pattern.

| **Example.** `/[^A-Z]/` matches a single character that is not an upper case letter.

Disjunction (pipe) Disjunction of regular expressions separated by `|`.

| **Example.** `/groundhog|woodchuck/` matches `groundhog` and `woodchuck`.

Wildcards

Optional A character followed by `?` can be matched optionally.

| **Example.** `/woodchucks?/` matches `woodchuck` and `woodchucks`.

Any `.` matches any character.

Kleene `*` A character followed by `*` can be matched zero or more times.

Kleene `+` A character followed by `+` must be matched at least once.

Counting A character followed by `{n,m}` must be matched from n to m times.

Example.

- `{n}` matches exactly n instances of the previous character.
- `{n,m}` matches from n to m instances of the previous character.
- `{n,}` matches at least n instances of the previous character.
- `{,m}` matches at most m instances of the previous character.

Anchors

Start of line `^` matches only at the start of line.

Example. `/^a/` matches a but not `ba`.

End of line `$` matches only at the end of line.

Example. `/a$/` matches a but not `ab`.

Word boundary `\b` matches a word boundary character.

Word non-boundary `\B` matches a word non-boundary character.

Aliases

- `\d` matches a single digit (same as `[0-9]`).
- `\D` matches a single non-digit (same as `[^\d]`).
- `\w` matches a single alphanumeric or underscore character (same as `[a-zA-Z0-9_]`).
- `\w` matches a single non-alphanumeric and non-underscore character (same as `[^\w]`).
- `\s` matches a single whitespace (space or tab).
- `\S` matches a single non-whitespace.

Capture group Operator to refer to previously matched substrings.

Example. In the regex `/the (.*)er they were, the \1er they will be/`, `\1` should match the same content matched by `(.*)`.

1.2 Tokenization

Lemma Words with the same stem and roughly the same semantic meaning. Lemma

Example. `cat` and `cats` are the same lemma.

Wordform Orthographic appearance of a word. Wordform

Example. `cat` and `cats` do not have the same wordform.

Vocabulary Collection of text elements, each indexed by an integer. Vocabulary

Remark. To reduce the size of a vocabulary, words can be reduced to lemmas.

Type / Wordtype Element of a vocabulary (i.e., wordforms in the vocabulary). Type / Wordtype

Token Instance of a type in a text. Token

Genre Topic of a text corpus (e.g., short social media comments, books, Wikipedia pages, ...). Genre

Remark (Herdan's law)**.** Given a corpus with N tokens, a vocabulary V over that corpus roughly have size:

$$|V| = kN^{\beta}$$

where the typical values are $10 \leq k \leq 100$ and $0.4 \leq \beta \leq 0.6$.

Stopwords Frequent words that can be dropped. Stopwords

Remark. If semantics is important, stopwords should be kept. LLMs keep stopwords.

Remark. For speed, simple tokenizers use regex.