

Natural Language Processing

Corpora and Preprocessing

Ramaseshan Ramachandran

NATURAL LANGUAGE

- ▶ Ambiguous, context-dependent
- ▶ Expresses thoughts, emotions, ideas
- ▶ Multiple ways to convey same idea
- ▶ Handles sarcasm, metaphor, humor
- ▶ Can be poetic, narrative, abstract
- ▶ Possible to Express simple ideas in complex forms
- ▶ Non-algorithmic, non-deterministic

PROGRAMMING LANGUAGE

- ▶ Algorithmic and precise
- ▶ Deterministic execution
- ▶ Strict syntax and semantics
- ▶ Designed for unambiguous logic
- ▶ Single best way to express a given task
- ▶ Machine-readable instructions

THE GAP

- ▶ NL: Flexible, rich, human-oriented
- ▶ PL: Rigid, exact, machine-oriented
- ▶ Challenge: Mapping ambiguity to precision
- ▶ Requires preserving meaning while enforcing structure

Bridging the Gap

BRIDGING THE GAP: NATURAL VS. PROGRAMMING LANGUAGE

- ▶ Natural: Ambiguous, flexible, human-focused
- ▶ Programming: Precise, deterministic, machine-focused
- ▶ Steps to convert fuzzy descriptions to executable code

UNDERSTAND THE INTENT

- ▶ Gap: Vague goals hidden in emotional wording
- ▶ Example: "Make the program smarter about choices" – unclear if basic logic or advanced features
- ▶ Address: Read multiple times, identify core objective (e.g., handle inputs predictably)

CLARIFY AMBIGUITIES

- ▶ Gap: Metaphors or sarcasm create confusion
- ▶ Example: "Code should fly through data" – literal speed? Or efficiency?
- ▶ Address: Ask specifics (e.g., "Optimize for speed or large datasets?") to fix meaning

BREAK IT DOWN

- ▶ Gap: Non-linear ideas vs. sequential code
- ▶ Example: "Check rain, suggest umbrella if not windy" – jumps without order
- ▶ Address: Split into inputs (weather), conditions (rain/wind), outputs (suggestion)

MAP TO STRUCTURES

- ▶ Gap: Non-deterministic "gut feel" vs. fixed logic
- ▶ Example: "Decide best full stack for this complex application" – varies by person
- ▶ Address: Use if-else or scoring for consistent decisions

USE SIMPLE TEMPLATES

- ▶ Gap: Complex sentences obscure simple ideas
- ▶ Example: "Initiate a procedural amalgamation of disparate data elements to facilitate the derivation of aggregated insights" – hides basic summation
- ▶ Address: Apply pseudocode: Input: data; Process: sum/group; Output: report

TEST AND REFINE

- ▶ Gap: Flexible interpretations vs. exact code
- ▶ Example: "Handle errors gracefully" – log, retry, or ignore?
- ▶ Address: Code outline, test scenarios, adjust for reliable match

DOCUMENT CLEARLY

- ▶ Gap: Ambiguity returns without notes
- ▶ Example: "This part is just brilliant" – sarcastic issue?
- ▶ Address: Add comments (e.g., "// Optimized 'brilliant' with caching")

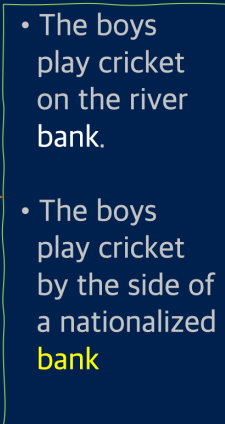
KEY BENEFITS

- ▶ Turns creative language into reliable code
- ▶ Ensures precision without losing intent
- ▶ Systematic for any description

Corpus Capital

IDEAL PROPERTIES OF A CORPUS

- Corpus is huge – Several billions of words
- Useful to verify a hypothesis about a language
- Find usage of a particular sound, word, or syntactic construction varies in different contexts
- Collection of most of the words of a language
- Even distribution of texts from all domains of language use

- 
- The boys play cricket on the river bank.
 - The boys play cricket by the side of a nationalized bank

LEXICAL RESOURCES

- ▶ The Brown Corpus contains a collection of written American English
- ▶ SUSANNE is a freely available, annotated subset of the Brown Corpus
- ▶ The Canadian Hansards, a bilingual parallel corpus, contains English and French transcripts of parliamentary proceedings
- ▶ The Penn Treebank contains syntactically annotated text from the Wall Street Journal
- ▶ NLP toolkits such as NLTK and spaCy ship with several corpora for learning purposes
- ▶ [Hugging Face](#) and [Kaggle](#) host thousands of text, image, and multimodal datasets
- ▶ Wikipedia dumps are available for hundreds of languages
- ▶ Web-scale corpora used for LLM pretraining: Common Crawl, C4, The Pile, RefinedWeb, FineWeb

DISAMBIGUATION OF BANK

Synset('bank.n.01') sloping land (especially the slope beside a body of water)

Synset('depository-financial-institution.n.01') a financial institution that accepts deposits and channels the money into lending activities

Synset('bank.n.03') a long ridge or pile

Synset('savings-bank.n.02') a container (usually with a slot in the top) for keeping money at home

Synset('bank.n.10') a flight maneuver (manoeuvre); aircraft tips laterally about its

longitudinal axis (especially in turning) Synset('bank.v.01') tip laterally

Synset('bank.v.02') enclose with a bank

Synset('bank.v.03') do business with a bank or keep an account at a bank

Synset('bank.v.04') act as the banker in a game or in gambling

Synset('bank.v.05') be in the banking business

Synset('deposit.v.02') put into a bank account

Synset('trust.v.01') have confidence or faith in

COMMON LAYERS OF NLP APPLICATIONS

- ▶ **Preprocessing layer**
- ▶ Data extraction layer
- ▶ Analysis of extracted information
- ▶ Semantic understanding
- ▶ Human/automatic evaluation of word meaning, sentence structure using the content obtained from the previous layers

OPERATIONS ON A CORPUS

- ▶ Text normalization converts text into a single canonical form: case folding, Unicode normalization, handling of foreign words, . . .
- ▶ Tokenization divides input text into tokens by identifying word boundaries (modern LLMs use subword tokenizers such as BPE)
- ▶ Identification/extraction picks out particular tokens, sentences, and paragraphs
- ▶ Corpus statistics cover token count, vocabulary size and type–token ratio

Preprocessing

PREPROCESSING

- ▶ Corpus creation
- ▶ Modifications required to make raw text suitable for processing
- ▶ Understanding the problem is crucial in choosing the preprocessing steps
- ▶ Preprocessing steps are specific to the problem at hand
- ▶ Improper preprocessing may destroy lexical context (e.g., case folding loses “US” vs. “us”)
- ▶ Note: modern LLM pipelines do minimal preprocessing. They keep case and punctuation and rely on subword tokenization; aggressive cleaning is mostly for classical/statistical pipelines

COMMONLY USED PREPROCESSING STEPS FOR ENGLISH

Preprocessing consists of (a) tokenization, (b) normalization and (c) substitution

- | | |
|---|--|
| <ul style="list-style-type: none">▶ Tokenization splits text into words/tokens▶ Case folding converts all text to lower case▶ Stemming turns running → run (crude suffix stripping)▶ Lemmatization turns best → good (dictionary-based root form)▶ Correct misspellings | <ul style="list-style-type: none">▶ Strip punctuation▶ Normalize white space, newlines, tabs, ...▶ Expand contractions, so isn't → is not and I'd → I would▶ Remove scripts and form variables from HTML/XML▶ Remove stop words (task-dependent) |
|---|--|

- ▶ An airplane accelerates down a runway at **3.20 m/s^2** for **32.8 s** until it finally lifts off the ground. Determine the distance traveled before takeoff
- ▶ How far will a car travel in **25 min** at **12 km/h** ?
- ▶ A jalopy with an initial speed of 23.7 km/h accelerates at a uniform rate of **0.92 m/s^2** for **3.6 s** . Find the final speed and the displacement of the jalopy during this time
- ▶ A college student wants to toss a textbook to his roommate who is leaning out of a window directly above him. He throws the book upwards with an initial velocity of **8.0 m/s** . The roommate catches it while it is traveling at **3.0 m/s** [up]. a) How long was the book in the air? b) How far vertically did the book travel? A car accelerates in a straight line from rest at the rate of **2.3 m/s^2** . What is its final velocity after **55 m** ? What is its time?
- ▶ 4. What height will a dart achieve **7 seconds** after being blown straight up at **50 m/s** ?

PREPROCESSING SAMPLE - FSHARP CODE

```
(true, PPSError.NoRegexError, ( doc |>  
    removeNewLine |>  
    removeHints |>  
    removeMathMinus |>  
    separateNumberEqualToSignBySpace |>  
    removeDashBetweenNumberAlpha |>  
    separateNumberAlphaBySpace |>  
    replaceTenPowerNotationToENotation |>  
    fixMSPattern |>  
    regexReplace directionReplacementTuples |>  
    regexReplace unitReplacementTuples |>  
    identifyAndMarkDegree |>  
    singlespace).ToLower())
```

PROCESSED CORPUS

Raw text	Preprocessed Text
An airplane accelerates down a runway at 3.20m/s² for 32.8 s until is finally lifts off the ground. Determine the distance traveled before takeoff	An airplane accelerates down a runway at 3.20 < m/s**2 > for 32.8 < s > until is finally lifts off the ground. Determine the distance traveled before takeoff
A college student wants to toss a textbook to his roommate who is leaning out of a window directly above him. He throws the book upwards with an initial velocity of 8.0 m/s . The roommate catches it while it is traveling at 3.0 m/s [up]. a) How long was the book in the air? b) How far vertically did the book travel?	A college student wants to toss a textbook to his roommate who is leaning out of a window directly above him. He throws the book upwards with an initial velocity of 8.0 < m/s > . The roommate catches it while it is traveling at 3.0 < m/s > [up]. a) How long was the book in the air? b) How far vertically did the book travel?

HTML PREPROCESSING

1. Case folding - convert content to lower case
2. Remove scripts
3. Tokenize
4. Term Frequency
5. Extract Vocabulary
6. Remove stop words
7. Stemming removes suffixes (e.g., the Porter stemmer)
8. Lemmatization maps a word to its dictionary root (e.g., Stanford CoreNLP, spaCy)
 - ▶ Stemming: running → run
 - ▶ Stemming: studies, studied, studying → studi
 - ▶ Lemmatization: best → good
9. Correct misspellings
10. Strip punctuation
11. Normalize white space, newlines, tabs, ...
12. Expand contractions, so isn't → is not and I'd → I would

PREPROCESSING EXAMPLE - HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="description" content="A sample HTML page for NLP preprocessing.">
  <meta name="keywords" content="HTML, NLP, preprocessing">
  <title>Sample NLP Page</title>
  <style>
    .highlight {
      color: red;
    }
  </style>
</head>
<body>
  <header>
    <h1>Welcome to the Sample Page</h1>
  </header>
  <div id="content">
    <p>This is some sample text for NLP processing. It includes <span class="highlight">important</span> keywords.</p>
    <p>Another paragraph with more text. We want to extract this text.</p>
    <script>
      function sayHello() {
        console.log("Hello from JavaScript!");
      }
    </script>
  </div>
  <footer>
    <p>&copy; 2023 Sample Site</p>
  </footer>
</body>
</html>
```

Processed text

Welcome to the Sample Page This is some sample text for NLP processing.
It includes important keywords. Another paragraph with more text.
We want to extract this text. 2023 Sample Site

PREPROCESSING HTML

```
import re
from bs4 import BeautifulSoup
def preprocess_html(html_file):
    with open(html_file, 'r', encoding='utf-8') as f:
        html_content = f.read()
    soup = BeautifulSoup(html_content, 'html.parser')
    # Remove script and style elements
    for script in soup(["script", "style", "meta"]):
        script.decompose()
    # Get extracted text
    text = soup.get_text()
    # Remove extra whitespace and newlines
    text = re.sub(r'\s+', ' ', text).strip()
    return text
```