CSCI E-89B Introduction to Natural Language Processing

Harvard Extension School

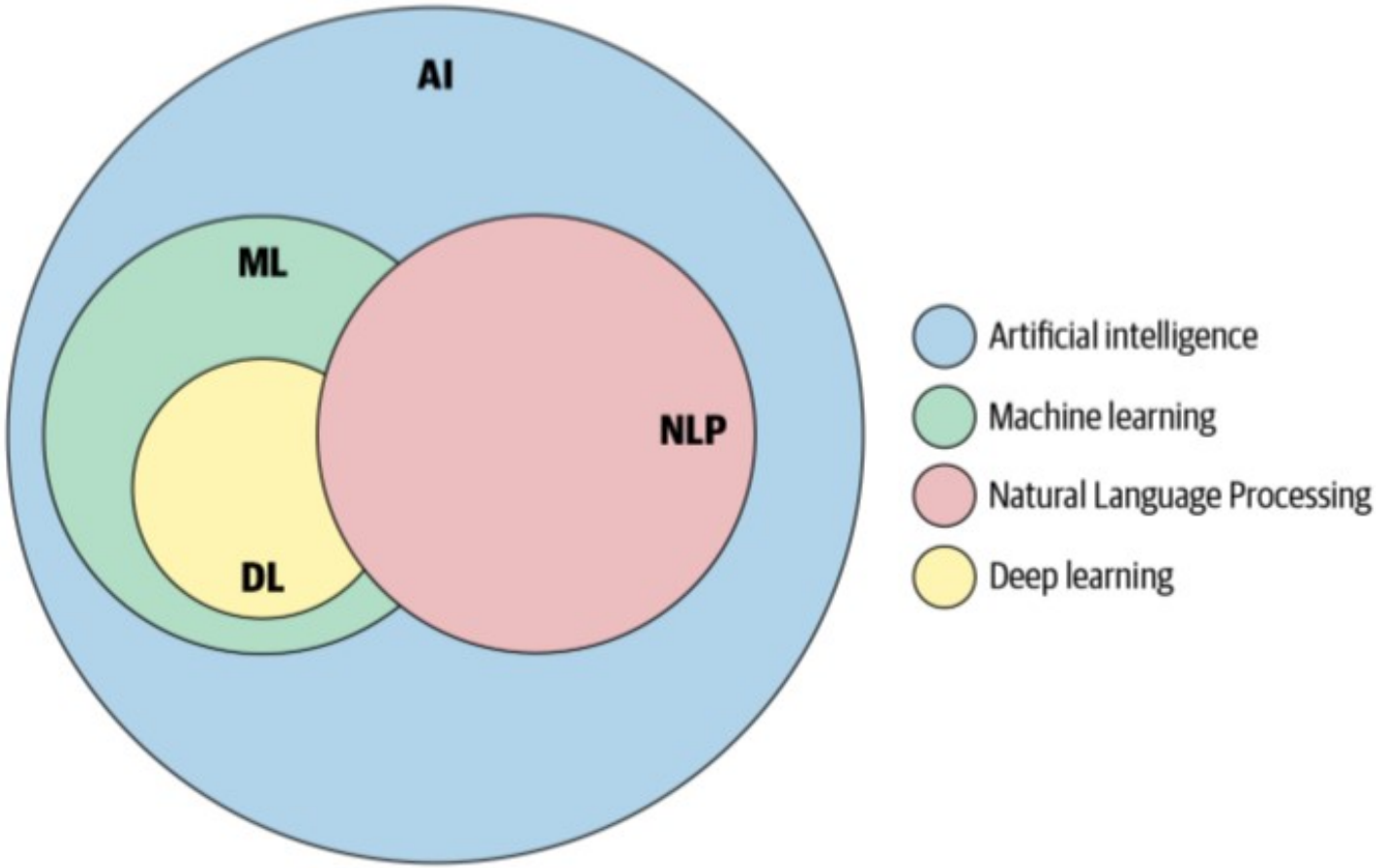
Dmitry Kurochkin

Fall 2024
Lecture 3

Contents

Contents

What is Natural Language Processing (NLP)?



Source: *Practical Natural Language Processing* by Vajjala et al.

What is Natural Language Processing (NLP)?

Natural Language Processing (NLP) is a multidisciplinary field integrating techniques from

- Linguistics (syntax, morphology, semantics, etc.) and
- Artificial Intelligence.

The main objective of NLP is to enable computers to understand, interpret, and generate human language, thereby facilitating natural and intuitive human-computer interactions.

Contents

Challenges the Natural Language Presents

Examples of ambiguous headlines:

- "Enraged Cow Injures Farmer with Ax"
- "Miners Refuse to Work after Death"
- "Squad Helps Dog Bite Victim"
- "Teacher Strikes Idle Children"
- "The Pope's Baby Steps on Gays"
- "New Study of Obesity Looks for Larger Test Group"
- "Kids Make Nutritious Snacks"
- "Hospitals Are Sued by 7 Foot Doctors"
- "Stolen Painting Found by Tree"
- "Two Sisters Reunite after 8 Years at Checkout Counter"
- "Juvenile Court to Try Shooting Defendant"

Applications of NLP

1. Text Classification
   - Examples: Spam detection, topic categorization, sentiment classification
   - Description:
     - ★ Assigning categories to text documents based on labeled data.
     - ★ Foundational NLP task leveraging supervised learning algorithms.

2. Named Entity Recognition (NER)
   - Examples: Extracting names of people, organizations, and locations
   - Description:
     - ★ Identifying and categorizing key entities within a corpus.
     - ★ Implemented using sequence labeling techniques.

Applications of NLP

3. Sentiment Analysis
   - Examples: Social media sentiment tracking, customer review analysis
   - Description:
     - ★ Discern the subjective sentiment expressed in text.
     - ★ Uses methods from lexicon-based approaches to deep learning models.

4. Information Retrieval
   - Examples: Search engines, document retrieval systems
   - Description:
     - ★ Indexes and retrieves relevant documents in response to queries.
     - ★ Techniques include TF-IDF, BM25, and rank-learning methods.

Applications of NLP

5. Optical Character Recognition (OCR)
   - Examples: Digitizing printed documents, automated data entry
   - Description:
     - ★ Transforms printed or handwritten text in images into machine-readable text.
     - ★ Modern OCR systems use convolutional neural networks (CNNs).

6. Machine Translation
   - Examples: Google Translate, DeepL
   - Description:
     - ★ Converts text from one language to another.
     - ★ Uses neural machine translation (NMT) models like Transformer.

Applications of NLP

7. Text Summarization
   - Examples: News aggregation, automatic report generation
   - Description:
     - ★ Condenses long documents into shorter, essential summaries.
     - ★ Includes extractive (key sentences) and abstractive (new sentences) methods.

8. Speech Recognition
   - Examples: Voice assistants, meeting transcriptions
   - Description:
     - ★ Converts spoken language into text.
     - ★ Combines acoustic models with language models.

Applications of NLP

9. Question Answering Systems
   - Examples: IBM Watson, SQuAD challenge systems
   - Description:
     - ★ Provides precise answers to natural language queries.
     - ★ Combines information retrieval and sophisticated NLP techniques.

10. Chatbots and Virtual Assistants
    - Examples: Apple Siri, Amazon Alexa, customer service bots
    - Description:
      - ★ Interacts conversationally with users.
      - ★ Uses natural language understanding (NLU) and natural language generation (NLG).

Applications of NLP

11. Document Summarization and Topic Modeling
    - Examples: Legal and clinical document analysis, news aggregation
    - Description:
        - ★ Involves summarizing texts and identifying key themes.
        - ★ Uses Latent Dirichlet Allocation (LDA) and advanced summarization algorithms.

12. Clinical Applications
    - Examples: EHR analysis, medical literature summarization
    - Description:
        - ★ Extracts and analyzes information from medical documents.
        - ★ Aids in clinical decision support and summarization of patient records.

Applications of NLP

13. Language Generation
    - Examples: Automated content creation, creative writing aids
    - Description:
        - ★ Creates coherent and contextually relevant text from prompts.
        - ★ Uses deep learning models like GPT-3 to assist in writing and generating dialogue.

Contents

Basic Text Processing

- Text processing is an essential step in the pipeline of Natural Language Processing (NLP).
- It prepares raw textual data for subsequent analysis and modeling.
- Key techniques:
  - Tokenization
  - Stemming
  - Lemmatization

Contents

Basic Text Processing

- Tokenization
  - Tokenization is the process of segmenting text into indivisible units called tokens.
  - It serves as the cornerstone for all subsequent text analysis tasks.
  - Tokens can represent phrases, words, subwords, or even characters.

- Importance of Tokenization
  - Facilitates language modeling by creating a vocabulary.
  - Enables subsequent tasks like parsing, stemming, and lemmatization.
  - Helps in handling linguistic intricacies such as:
    - ★ Handling punctuation
    - ★ Recognizing entities
    - ★ Processing contractions and compound words

Tokenization Techniques

- **Word Tokenization:**
  - Breaks text into words.
  - Example:
    "Karl Benz invented the first car."
    → ["Karl", "Benz", "invented", "the", "first", "car", "."]
- **Sentence Tokenization:**
  - Breaks text into sentences.
  - Example:
    "Henry Ford created assembly lines. This revolutionized car production."
    → ["Henry Ford created assembly lines.", "This revolutionized car production."]
- **Subword Tokenization:**
  - Breaks text into subword units.
  - Example:
    "Hydrogen-powered"
    → ["Hydrogen", "-", "powered"]

Contents

Stemming

- Stemming aims to reduce words to their root form, stripping affixes (prefixes and suffixes).
- Important for reducing morphologically similar words to a common base form.
- Example: "running" $\rightarrow$ "run", "jumps" $\rightarrow$ "jump"

Common Stemming Algorithms

- **Porter Stemmer:**
  - Developed By: Martin Porter in 1979.
  - Heuristic-based rules to strip suffixes. Widely used in English text processing.
  - Strengths: Simple and effective.
  - Weaknesses: May produce non-root forms that aren't actual words
  - Example: ["electric", "transportation"] $\rightarrow$ ["electr", "transport"]
- **Snowball Stemmer:**
  - Also developed by Martin Porter, an improvement over the original.
  - Supports multiple languages, Improved accuracy and flexibility.
  - Weaknesses: More complex than Porter Stemmer.
- **Lancaster Stemmer (Paice/Husk):**
  - Developed By: Chris Paice.
  - Highly aggressive stemming approach, can strip away more of the word's characters.
  - Weaknesses: May over-stem, producing non-intuitive results

Contents

Lemmatization

- Lemmatization reduces words to their base or dictionary form (lemma) by considering context and part of speech.
- It provides more accurate and linguistically valid base forms compared to stemming.
- Example: "cars" (noun) $\rightarrow$ "car", "driving" (verb) $\rightarrow$ "drive"
- Examples on Lemmatization vs. Stemming:
  - "cars" (noun) $\rightarrow$ "car" (lemma), "car" (stem)
  - "driving" (verb) $\rightarrow$ "drive" (lemma), "driv" (stem)
  - "gears" (noun) $\rightarrow$ "gear" (lemma), "gear" (stem)
  - "faster" (adjective) $\rightarrow$ "fast" (lemma), "faster" (stem)
  - "inflated" (verb) $\rightarrow$ "inflate" (lemma), "infl" (stem)
  - "engines" (noun) $\rightarrow$ "engine" (lemma), "engin" (stem)

Lemmatization Techniques

- **WordNet Lemmatizer:**
  - Utilizes the WordNet lexical database to find lemmas.
  - Considers part of speech (POS) tags for accurate lemmatization.
  - Example: "innovations" (noun) $\rightarrow$ "innovation"
- **SpaCy Lemmatizer:**
  - Integrated into SpaCy, an industrial-strength NLP library.
  - Example: "was" (verb, past tense) $\rightarrow$ "be", "automobiles" (noun) $\rightarrow$ "automobile"

Contents

Tokenization with NLTK - Words

```
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize

text = "Henry Ford's innovation, the assembly line process,
        changed the car industry's dynamics profoundly."

# Tokenize into words
word_tokens = word_tokenize(text)
print(word_tokens)
```

```
['Henry', 'Ford', "'s", 'innovation', ',', 'the', 'assembly',
'line', 'process', ',', 'changed', 'the', 'car', 'industry',
"'s", 'dynamics', 'profoundly', '.']
```

Tokenization with NLTK - Sentences

```
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize

text = "Henry Ford revolutionized the car industry with the
        introduction of assembly lines. Karl Benz pioneered
        the automotive age with the invention of the first
        practical automobile. Ferdinand Porsche revolutionized
        sports cars with the development of the Volkswagen
        Beetle and the Porsche 911."

# Tokenize into sentences
sent_tokens = sent_tokenize(text)
print(sent_tokens)
```

```
['Henry Ford revolutionized the car industry with the introduction
of assembly lines.', 'Karl Benz pioneered the automotive age with
the invention of the first practical automobile.', 'Ferdinand
Porsche revolutionized sports cars with the development of the
Volkswagen Beetle and the Porsche 911.']
```

Contents

Stemming with NLTK

```
from nltk.stem import PorterStemmer, SnowballStemmer

text = "Henry Ford's innovation, the assembly line process,
changed the car industry's dynamics profoundly."

# Tokenize into words
words = word_tokenize(text)
print("Word Tokens:", words)

# Apply Porter Stemmer
porter = PorterStemmer()
porter_stems = [porter.stem(word) for word in words]
print("Porter Stemmer Results:", porter_stems)

# Apply Snowball Stemmer
snowball = SnowballStemmer("english")
snowball_stems = [snowball.stem(word) for word in words]
print("Snowball Stemmer Results:", snowball_stems)
```

Stemming with NLTK (Continued)

```
Word Tokens:
['Henry', 'Ford', "'s", 'innovation', ',', 'the', 'assembly',
'line', 'process', ',', 'changed', 'the', 'car', 'industry',
"'s", 'dynamics', 'profoundly', '.']

Porter Stemmer Results:
['henri', 'ford', "'s", 'innov', ',', 'the', 'assembl', 'line',
'process', ',', 'chang', 'the', 'car', 'industri', "'s", 'dynam',
'profoundli', '.']

Snowball Stemmer Results:
['henri', 'ford', "'s", 'innov', ',','the', 'assembl', 'line',
'process', ',', 'chang', 'the', 'car', 'industri', "'s", 'dynam',
'profound', '.']
```

Contents

Lemmatization with SpaCy

```
!pip install spacy
!python -m spacy download en_core_web_sm

from nltk.corpus import wordnet
from nltk.stem import WordNetLemmatizer
import spacy

nltk.download('punkt')
nltk.download('wordnet')

text = "Henry Ford's innovation, the assembly line process,
changed the car industry's dynamics profoundly."

# Tokenize the sentence into words
words = word_tokenize(text)
print("Word Tokens:", words
```

Lemmatization with SpaCy (Continued)

```
# Initialize WordNet Lemmatizer
wordnet_lemmatizer = WordNetLemmatizer()

# Apply WordNet Lemmatizer
wordnet_lemmas = [wordnet_lemmatizer.lemmatize(word, pos='v')
                  for word in words]
print("WordNet Lemmatizer Results:", wordnet_lemmas)

# Using SpaCy for Lemmatization
nlp = spacy.load("en_core_web_sm")
doc = nlp(text)
spacy_lemmas = [token.lemma_ for token in doc]
print("SpaCy Lemmatizer Results:", spacy_lemmas)
```

Lemmatization with SpaCy (Continued)

```
Word Tokens:
['Henry', 'Ford', "'s", 'innovation', ',', 'the', 'assembly',
'line', 'process', ',', 'changed', 'the', 'car', 'industry',
"'s", 'dynamics', 'profoundly', '.']

WordNet Lemmatizer Results:
['Henry', 'Ford', "'s", 'innovation', ',', 'the', 'assembly',
'line', 'process', ',', 'change', 'the', 'car', 'industry',
"'s", 'dynamics', 'profoundly', '.']

SpaCy Lemmatizer Results:
['Henry', 'Ford', "'s", 'innovation', ',', 'the', 'assembly',
'line', 'process', ',', 'change', 'the', 'car', 'industry',
"'s", 'dynamic', 'profoundly', '.']
```