

Natural Language Processing Data Preprocessing

Lecture 06

Prepared by

Dr. Hager Saleh

What is NLP (Natural Language Processing)?

Natural Language Processing (NLP) is a field of Artificial Intelligence (AI) that focuses on enabling computers to understand, interpret, generate, and respond to human languages in a way that is both meaningful and useful.

Different Applications of NLP Classification

- Sentiment Analysis (Emotion Classification)
- Spam Detection (Email & Message Classification)
- Topic Classification (News and Document Categorization)

Main steps of developing NLP for classification

- Data Collection
- Data Preprocessing
- Feature Extraction
- Model Selection and Training
- Evaluation

The main types of data preprocessing

1. Text Cleaning: Preprocessing the raw text to remove noise and irrelevant content.
 - Lowercasing: Convert all text to lowercase to ensure uniformity. Example: "Hello World" → "hello world"

- Removing Punctuation: Eliminating symbols like .,!@#. Example: "hello, world!" → "hello world"
 - Removing Special Characters: Remove non-alphanumeric characters. Example: "I love #AI & NLP" → "I love AI NLP"
 - Removing Numbers (optional): If numbers are not meaningful for the task. Example: "Today is 13th" → "Today is"
 - Removing URLs, Emails, Mentions: For social media, forums, emails. Example: "Check <http://example.com>" → "Check"
 - Removing HTML tags:
2. Tokenization: Splitting text into units: words, subwords, or sentences.
 - Word Tokenization: "I love NLP." → ["I", "love", "NLP", "."]
 - Sentence Tokenization: "I love NLP. It is powerful." → ["I love NLP.", "It is powerful."]
 3. Normalization: Reducing text to a canonical form.
 - Stemming: Reduce words to root form (may not be meaningful). Example: "playing", "played", "plays" → "play" Tool: PorterStemmer, SnowballStemmer
 - Lemmatization: Reduce words to dictionary form using context. Example: "am", "are", "is" → "be" Tool: WordNetLemmatizer
 4. Stop Words Removal: Eliminate common but insignificant words (e.g., "the", "is", "in").

Example: "This is a book" → "book"

```
#pip install nltk spacy

import re
import nltk
import spacy
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer

# Download NLTK resources (only first time)
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

stemmer = PorterStemmer()

# Stop words list
stop_words = set(stopwords.words('english'))
```

```

# -----
# Function to apply all preprocessing steps
# -----

def preprocess_text(text):
    print(f"\nOriginal Text: {text}")

    # 1. Lowercasing
    text = text.lower()
    print(f"Lowercased Text: {text}")

    # 2. Remove URLs
    text = re.sub(r'http\S+|www.\S+', '', text)
    print(f"Removed URLs: {text}")

    # 3. Remove HTML tags
    text = re.sub(r'<.*?>', '', text)
    print(f"Removed HTML Tags: {text}")

    # 4. Remove Special Characters, Numbers, Punctuation
    text = re.sub(r'^a-zA-Z\s]', '', text)
    print(f"Removed Special Characters: {text}")

    # 5. Tokenization
    tokens = word_tokenize(text)
    print(f"Tokens: {tokens}")

    # 6. Remove Stop Words
    tokens = [word for word in tokens if word not in stop_words]
    print(f"Removed Stopwords: {tokens}")

    # 7. Stemming
    stemmed_tokens = [stemmer.stem(word) for word in tokens]
    print(f"Stemmed Tokens: {stemmed_tokens}")

    # 8. Final clean text reconstruction (optional)
    clean_text = ' '.join(stemmed_tokens)
    print(f"Final Preprocessed Text: {clean_text}")

    return clean_text

```

```

[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\compumarts\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\compumarts\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!

```

```
[nltk_data] Downloading package wordnet to
[nltk_data] C:\Users\compumarts\AppData\Roaming\nltk_data...
[nltk_data] Package wordnet is already up-to-date!
```

```
sample_text = """
Hello! This is a test document. Visit https://example.com for more
info.
I bought 5 products from Amazon. Let's meet at 5:00 PM. I can't wait
to see you! 😊
"""
```

```
final_text = preprocess_text(sample_text)
```

Original Text:

```
Hello! This is a test document. Visit https://example.com for more
info.
I bought 5 products from Amazon. Let's meet at 5:00 PM. I can't wait
to see you! 😊
```

Lowercased Text:

```
hello! this is a test document. visit https://example.com for more
info.
i bought 5 products from amazon. let's meet at 5:00 pm. i can't wait
to see you! 😊
```

Removed URLs:

```
hello! this is a test document. visit for more info.
i bought 5 products from amazon. let's meet at 5:00 pm. i can't wait
to see you! 😊
```

Removed HTML Tags:

```
hello! this is a test document. visit for more info.
i bought 5 products from amazon. let's meet at 5:00 pm. i can't wait
to see you! 😊
```

Removed Special Characters:

```
hello this is a test document visit for more info
i bought products from amazon lets meet at pm i cant wait to see you
```

```
Tokens: ['hello', 'this', 'is', 'a', 'test', 'document', 'visit',
'for', 'more', 'info', 'i', 'bought', 'products', 'from', 'amazon',
'lets', 'meet', 'at', 'pm', 'i', 'cant', 'wait', 'to', 'see', 'you']
```

```
Removed Stopwords: ['hello', 'test', 'document', 'visit', 'info',
'bought', 'products', 'amazon', 'lets', 'meet', 'pm', 'cant', 'wait',
'see']
```

```
Stemmed Tokens: ['hello', 'test', 'document', 'visit', 'info',
'bought', 'product', 'amazon', 'let', 'meet', 'pm', 'cant', 'wait',
'see']
```

Final Preprocessed Text: hello test document visit info bought product
amazon let meet pm cant wait see