Chapter 11: Text Analysis II: Advanced Methods and Applications

School of International Liberal Studies
Waseda University
Introductory Data Science

Chapter Objective

- Move beyond basic preprocessing to structured insights.
- Learn **Named Entity Recognition (NER)** for extracting people, places, and organizations.
- Explore **word frequencies** and **n-grams** for context and phrase patterns.
- Apply **vectorization** methods (BoW, TF–IDF) to represent text numerically.
- Introduce **topic modeling** (LDA) to uncover latent themes in documents.
- Reflect on **ethics and bias** in text analysis workflows.

Named Entity Recognition (NER)

"Marie Curie won the Nobel Prize in Physics in 1903."

A Named Entity Recognizer might identify:

- `Marie Curie` → **PERSON**
- `Nobel Prize` → **WORK_OF_ART** or **ORG**
- `Physics` → may not be recognized (model-dependent)
- `1903` → **DATE**

NER with spaCy

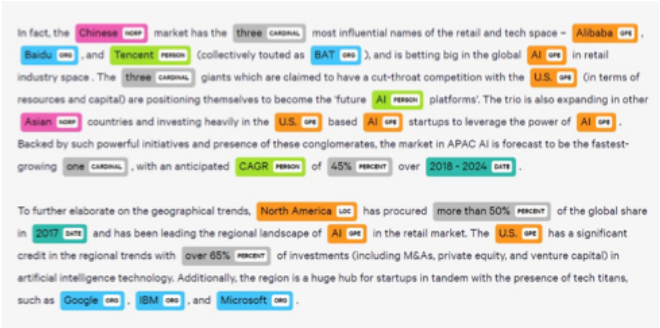
```
import spacy
nlp = spacy.load("en_core_web_sm")

text = "Barack Obama was born in Hawaii and served as President of the United
    States."
doc = nlp(text)

for ent in doc.ents:
    print(ent.text, ent.label_)
```

Visualizing Entities

- Automatic detection of entities in text.
- Recognizes people, countries, companies, dates, locations.
- Trained with supervised machine learning on labeled corpora.

Instructor Tip

Show how NER converts unstructured text into structured knowledge (like a spreadsheet).



Example of spaCy NER highlighting people, companies, and locations

Interpreting Common Labels

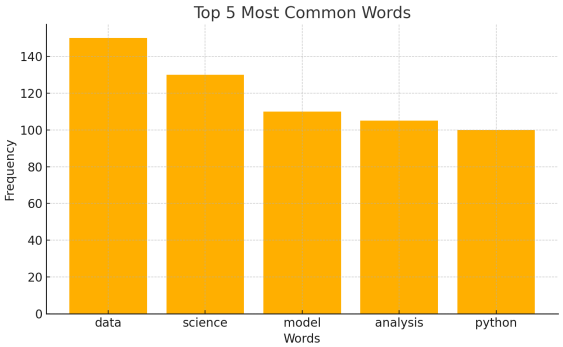
- **PERSON**: people, fictional or real.
- **ORG**: organizations.
- **GPE**: geopolitical entities (countries, cities).
- **LOC**: non-GPE locations (mountains, rivers).
- **NORP**: nationalities, religious, or political groups.
- **DATE**, **TIME**, **MONEY**, **PERCENT**, etc.

Word Frequency

Word frequency = how often a particular word appears in a document or across documents.

```
from collections import Counter
import spacy

nlp = spacy.load("en_core_web_sm")
text = "Data science is science that uses data."
doc = nlp(text)

tokens = [t.lemma_.lower() for t in doc
          if not t.is_stop and not t.is_punct]
freq = Counter(tokens)
print(freq)
```

Quick Bar Plot of Top Words



Top 5 most common words with their frequencies

```python
import matplotlib.pyplot as plt

# Word frequencies
freq = {"analysis": 105, "data": 150,
        "python": 100, "science": 130,
        "model": 110}

# Sort by frequency and take top 5
top = sorted(freq.items(),
             key=lambda x: x[1],
             reverse=True)
words, counts = zip(*top)

plt.bar(words, counts, color="orange")
plt.title("Top 5 Most Common Words")
plt.xlabel("Words")
plt.ylabel("Frequency")
```

What Are N-grams?

- **N-gram** = sequence of `n` units (usually words).
- Unigram = 1 word; Bigram = 2 words; Trigram = 3 words.
- Capture common phrases and local context.

Example:

```
"The quick brown fox jumps over the lazy dog."
```

Bigrams:

```
[('the','quick'), ('quick','brown'), ('brown','fox'), ...]
```

Creating Bigrams with `nltk`

```
from nltk import bigrams
from nltk.tokenize import word_tokenize

text = "The quick brown fox jumps over the lazy dog."
tokens = word_tokenize(text.lower())

bi = list(bigrams(tokens))
print(bi)
```

Why N-grams Matter

- Identify common phrases and collocations.
- Capture context that single words miss (e.g., "climate change").
- Provide useful features for machine learning models.

Vectorizing Text

- Convert text into numerical vectors.
- **Bag-of-Words (BoW)**: counts word frequency, ignores order.
- **TF-IDF**: reweights words frequent in one document but rare globally.

Bag-of-Words with CountVectorizer

```
from sklearn.feature_extraction.text import CountVectorizer

docs = ["data science is fun", "data science uses data"]
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(docs)

print(vectorizer.get_feature_names_out())
print(X.toarray())
```

TF-IDF with TfidfVectorizer

```
from sklearn.feature_extraction.text import TfidfVectorizer

docs = ["data science is fun", "data science uses data"]
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(docs)

print(vectorizer.get_feature_names_out())
print(X.toarray())
```

Topic Modeling

- **Topic modeling** groups words that frequently occur together across documents.
- Each "topic" is a cluster of co-occurring terms, discovered automatically.
- Human interpretation is needed to assign meaningful labels.

Example

Suppose a model outputs the following cluster of words:

`Topic 1: economy, inflation, market, prices, stocks`

This cluster could be interpreted as the topic "**finance**" **or** "**economic news**".

LDA with gensim

```
from gensim import corpora, models
from nltk.tokenize import word_tokenize

texts = [
  word_tokenize("I love data science and machine learning."),
  word_tokenize("Python is great for data analysis."),
  word_tokenize("Stocks and markets rise with economic data.")
]

dictionary = corpora.Dictionary(texts)
corpus = [dictionary.doc2bow(t) for t in texts]

lda = models.LdaModel(corpus, num_topics=2, id2word=dictionary)
print(lda.print_topics())
```

Applications of Text Analysis

- Journalism/media: trend detection, discourse analysis.
- Humanities: style/themes/authorship, character networks.
- Social science: framing, ideology, debates.
- Business: reviews, surveys, brand sentiment.
- Everyday tech: spam filters, autocomplete, assistants.

Ethical Considerations

- Privacy & consent in data collection.
- Bias propagation and amplification.
- Over-interpretation of statistical artifacts.
- Cultural/linguistic transfer issues.

Common Types of Bias

- Historical and Social Bias
- Representation and Sampling Bias
- Measurement Bias
- Aggregation Bias
- Evaluation Bias
- Learning Bias
- Deployment Bias

Responsible Practices

- Transparency about methods/assumptions.
- Human validation and close reading.
- Think critically about inclusion/exclusion.
- Avoid deterministic conclusions.

Summary

- NER extracts structured entities for downstream analysis.
- Frequency & n-grams reveal themes and phrases.
- Vectorization (BoW/TF-IDF) enables ML on text.
- LDA surfaces latent topics; interpret carefully.
- Ethics/bias are integral to responsible NLP.