Lesson 3: Topic Modeling and Text Summarization

In this lesson, we'll dive into two powerful unsupervised NLP techniques: Topic Modeling and Text Summarization. Both help us make sense of large volumes of text by extracting the most relevant information.

---

1. What is Topic Modeling?

Topic Modeling is an unsupervised technique that uncovers the hidden thematic structure in large text collections. It clusters words into "topics" and assigns documents to these topics based on word usage.

Common Algorithms:

- **LDA (Latent Dirichlet Allocation)** – the most widely used probabilistic model for topic discovery.
- **NMF (Non-negative Matrix Factorization)** – matrix factorization approach for identifying topics.

Example Use Cases:

- **Grouping customer reviews by underlying themes.**
- **Analyzing research papers for dominant topics.**
- **Organizing news articles.**

---

2. Implementing LDA with Gensim

```
import gensim
from gensim import corpora
from gensim.models.ldamodel import LdaModel
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import nltk
nltk.download('punkt')
nltk.download('stopwords')
# Sample Documents
documents = [
   "Machine learning is a field of artificial intelligence.",
   "Natural language processing involves text and speech.",
   "Deep learning models are used for image and speech recognition.",
   "Transformers have revolutionized NLP tasks like translation and summarization."
]
# Preprocessing
stop_words = set(stopwords.words('english'))
texts = [
   [word for word in word_tokenize(doc.lower()) if word.isalpha() and word not in stop_words]
   for doc in documents
]
# Create Dictionary and Corpus
dictionary = corpora.Dictionary(texts)
corpus = [dictionary.doc2bow(text) for text in texts]

# LDA Model
lda = LdaModel(corpus=corpus, id2word=dictionary, num_topics=2, passes=10)

# Print Topics
for idx, topic in lda.print_topics(-1):
   print(f"Topic {idx}: {topic}")
```

---

3. Evaluating Topic Models

- **Coherence Score: Measures how semantically related the words in a topic are.**
- **Perplexity: Lower values generally indicate a better model (used less frequently now).**
- **Code Example**

```
from gensim.models.coherencemodel import CoherenceModel

coherence_model = CoherenceModel(model=lda, texts=texts, dictionary=dictionary, coherence='c_v')
print("Coherence Score:", coherence_model.get_coherence())
```

---

4. What is Text Summarization?

Text summarization is the process of distilling the most important information from a text.

Two Types:

- **Extractive Summarization: Selects key sentences from the text (e.g., TextRank).**
- **Abstractive Summarization: Generates new sentences that summarize the content, like humans do (e.g., Transformers like BART, T5).**

---

5. TextRank for Extractive Summarization (spaCy + NetworkX)

```
import spacy
import networkx as nx
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer
import numpy as np

nlp = spacy.load("en_core_web_sm")
text = """
Natural Language Processing (NLP) is transforming how we interact with machines. From chatbots to voice assistants, NLP helps machines understand human language. It's a critical component of AI systems used in healthcare, finance, and customer service.
"""

# Sentence Tokenization
doc = nlp(text)
sentences = [sent.text.strip() for sent in doc.sents]

# TF-IDF Vectorization
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(sentences)

# Cosine Similarity Matrix
sim_matrix = cosine_similarity(X)

# Build Graph and Rank Sentences
nx_graph = nx.from_numpy_array(sim_matrix)
scores = nx.pagerank(nx_graph)

# Rank Sentences
ranked_sentences = sorted(((scores[i], s) for i, s in enumerate(sentences)), reverse=True)

# Summary (top 2 sentences)
summary = " ".join([s for _, s in ranked_sentences[:2]])
print("Summary:", summary)
```

---

6. Abstractive Summarization with Hugging Face Transformers

```
from transformers import pipeline

summarizer = pipeline("summarization")
text = """
The field of Natural Language Processing has seen tremendous growth. With the advent of transformer-based models like BERT and GPT, machines are now able to generate human-like text, translate languages, and answer questions more accurately than ever before.
"""

summary = summarizer(text, max_length=50, min_length=25, do_sample=False)
print("Summary:", summary[0]['summary_text'])
```

---

7. Comparison Table

Technique	Type	Tools/Libraries	Best For
LDA	Unsupervised	Gensim	Discovering hidden themes in large corpora
TextRank	Extractive	spaCy, NetworkX, sklearn	Quick summaries from raw text
Transformer Summarizers	Abstractive	Hugging Face Transformers	High-quality, human-like summaries

---

8. Summary

- Topic Modeling helps discover hidden themes in documents.
- LDA is a powerful unsupervised algorithm for topic discovery.
- Text Summarization can be extractive (TextRank) or abstractive (Transformers).
- spaCy, Gensim, NetworkX, and Hugging Face provide robust tools for these tasks.