MODULE 6 · STUDY NOTES

Introducing Natural Language Processing

How machines analyze human language, the NLP development pipeline from raw text to meaning, and the five AWS managed services — Transcribe, Polly, Translate, Comprehend, and Lex — that put NLP to work

Every time you ask Alexa about the weather, run a document through a spell checker, or read auto-generated subtitles, natural language processing is at work. NLP is the broad field of building computational algorithms that automatically analyze and represent human language — and it is older than machine learning, though most modern NLP now relies on ML. This module has two halves. The first builds intuition for what NLP is, why human language is so difficult for computers, and how an NLP solution follows the same development pipeline as any ML project: preprocess the raw text, tokenize it and engineer features, then analyze the result to classify, match, and extract meaning. The second half is practical: five AWS managed services — Amazon Transcribe, Amazon Polly, Amazon Translate, Amazon Comprehend, and Amazon Lex — that let you add speech recognition, speech synthesis, translation, text insight, and conversational chatbots to an application without building or training your own models, and that compose into powerful multilingual, spoken, and analyzed pipelines.

LEARNING OBJECTIVES

- Describe the natural language processing (NLP) use cases that are solved by using managed Amazon ML services
- Describe the managed Amazon ML services available for NLP
- Use managed Amazon ML services

6.1 What is natural language processing?

NLP develops computational algorithms to automatically analyze and represent human language. By evaluating the structure of language, machine learning systems can process large sets of words, phrases, and sentences. It helps to keep the scope in mind: NLP is a broad term for a general set of business or computational problems that you can solve with machine learning — but **NLP systems predate ML**. Two classic examples that used no machine learning are speech-to-text on an old cell phone and screen readers. Many NLP systems now use some form of machine learning, and that shift is what makes services like the ones in this module possible.

WORKED EXAMPLE – "ALEXA, WHAT'S IT LIKE OUTSIDE?"

Before the formal definition, follow one request end to end. An Amazon device such as an Echo **records your words**, and the recording is sent to Amazon servers to be analyzed more efficiently. Amazon **breaks the phrase into individual sounds**, then connects to a database of word pronunciations to find which words most closely match that combination of sounds. It **identifies the important words** to make sense of the task — noticing words like *outside* or *temperature*, it opens the weather app and carries out the function. Finally, the servers **send information back** to your device, and Alexa speaks the answer. Recording, sound matching, keyword identification, and response generation are all NLP tasks.

LANGUAGE IS A HIERARCHY

- **Words** sit at the lowest layer of the hierarchy.
- A group of words makes a **phrase**.
- Phrases combine to make a **sentence**.
- Ultimately, sentences convey **ideas** — the level NLP is trying to reach.

You can apply NLP to a wide range of problems. Common applications include **search applications** (such as Google and Bing), **human-machine interfaces** (such as Alexa), **sentiment analysis** for marketing or political campaigns, **social research** based on media analysis, and **chatbots** that mimic human speech inside applications. These use cases recur throughout the module and motivate the AWS services in the second half.

6.2 Why NLP is hard: the four challenges

Language is not precise. Words can have different meanings based on the other words that surround them — their context — and the same words or phrases can carry multiple meanings. Consider the word *weather*: you might be *under the weather*, meaning you are sick, yet *there is wonderful weather outside* describes good conditions. The phrase *Oh, really?* might convey surprise, disagreement, or many other meanings depending on context and inflection. This imprecision, context-dependence, complex dependencies between words, and overall lack of structure are exactly what make human language difficult for computers.

TABLE 6.1 The four main challenges for an NLP system

CHALLENGE	WHAT IT MEANS
Discovering the structure of the text	One of the first tasks of any NLP application is to break the text into meaningful units — words, phrases, and sentences
Labeling data	After text is converted to data, apply labels for the various parts of speech; every language needs a different labeling scheme to match its grammar
Representing context	Because word meaning depends on context, the system needs a way to represent context — hard because of the huge number of contexts, and difficult to put in a form computers understand
Applying grammar	Grammar defines a structure, but its application is nearly infinite; handling the variation in how humans use language is where machine learning has a big impact

COMMON PITFALL

Do not assume grammar rules alone can tame language. Although grammar defines a structure, its application is nearly infinite — the same idea can be expressed countless ways. Dealing with that variation is a major challenge, and it is precisely where machine learning, rather than hand-coded rules, makes the difference.

6.3 The NLP development pipeline

You can apply the same ML development pipeline used throughout this course when you develop an NLP solution. The first task is to **formulate the problem**, and then **collect and label data**. For NLP, collecting data consists of breaking the text into meaningful subsets and labeling the sets — a step also called **tagging**, where you assign individual text strings to different parts of speech. **Feature engineering is a large part** of NLP applications, and it gets more complicated with irregular or unstructured text: to classify documents, for example, you must distinguish between words that share a common term but have different meanings. Specialized tools help with NLP labeling.

The first hands-on task is to **convert the text to data** so it can be analyzed, which begins with preprocessing. Common preprocessing steps are removing stop words, normalizing similar text, and standardizing unrecognized text; other steps include encoding and spelling and grammar checks. Multiple libraries and tools are available — for example, the **Natural Language Toolkit (NLTK)** for Python provides functions to remove stop words, normalize, and standardize.

TABLE 6.2 Preprocessing an input string

STEP	WHAT IT DOES	EXAMPLE
Remove stop words	Delete words not needed for the analysis	"This is sample text" → "sample text" (drop "This", "is")
Normalize	Collapse word forms into a common form using stemming and lemmatization	"ran", "running" → "run"
Standardize	Remove or replace words not in the analysis dictionary — acronyms, slang, special characters	"DM" → "direct message"; "ltr" → "later"

After the text is clean, you **load it into a data collection** such as a DataFrame using **tokens** — for example, NLTK's `word_tokenize` function converts words into DataFrame items. You then apply an NLP model to **create features**. Two common feature models are **bag of words**, a simple model that captures the frequency of each word in a document (a key per word, valued by its number of occurrences), and **term frequency and inverse document frequency (TF-IDF)**, which combines a count of how many times a word appears in a document with how many times it occurs across a group of documents to compute a weight — so words that appear in many documents receive a lower weight.

WORKED EXAMPLE – BAG OF WORDS IS A VECTOR MODEL

Bag of words converts each sentence or phrase into a **vector** — a mathematical object that records both directionality and magnitude — where each word is recorded in terms of frequency. Take the sentence *Today is Wednesday and it is my birthday*: the word *is* is recorded with a value of **2** because it appears twice. Bag of words is often used to classify documents into categories, and to derive attributes that feed downstream NLP applications such as sentiment analysis. Notice what it does not capture: word order and meaning are discarded — only counts remain.

6.4 Analyzing text and deriving meaning

Once text is preprocessed and turned into features, analysis falls into three broad categories. **Classifying text** works like other classification systems in this course: text is the input to a process that extracts features, which pass through an ML algorithm that interacts with a classifier model to infer the classification (the NLTK Python library can build such a system). **Discovering similarities** powers text matching — autocorrect, spell check, and grammar check — frequently using the **edit distance**, also known as the **Levenshtein distance**, algorithm. **Deriving relationships** between different words or phrases uses a process called **coreference resolution**, and several NLP systems provide Python libraries for it.

CAPTURING CONTEXT	ENTITY EXTRACTION WITH NER
one of NLP's largest challenges ▪ Tag words with the appropriate **part of speech** to help capture context ▪ NLP libraries provide token functions to help with tagging ▪ Example: the word *tablet* means medicine or a computing device ▪ A qualifying term (medicine, computing) disambiguates it; otherwise, search engines fall back on the most common context	*named entity recognition* ▪ **Identify noun phrases** using dependency charts and part-of-speech tagging ▪ **Classify phrases** with a classification algorithm such as Word2Vec ▪ **Disambiguate entities** using a knowledge graph ▪ A knowledge graph blends expertise with ML to derive meaning

WORKED EXAMPLE – NER ON A SENTENCE

Given *The ocean liner Titanic sank in the North Atlantic*, a NER model extracts the named entities and tags them: "**ship**": "**Titanic**" and "**location**": "**North Atlantic**". After the entities are extracted, a **knowledge graph** is used to extract meaning — combining subject matter expertise with machine learning. The **Amazon recommendations engine** is an example of a knowledge graph in production.

SECTION 1 KEY TAKEAWAYS

- As a domain, **NLP predates machine learning**.
- NLP development **maps directly to the ML development process**.
- Main use cases include **search query analysis, human-machine interaction, and market or social research**.
- NLP is difficult because of the **imprecise nature of human language**.

6.5 Managed services for speech: Amazon Transcribe and Amazon Polly

The rest of the module reviews five managed Amazon ML services that simplify building NLP applications — you consume them as-is, without training your own models. The two speech services move in opposite directions. **Amazon Transcribe** is a fully managed service that uses machine learning to recognize speech in audio files and transcribe it into text. It can recognize specific recorded voices, convert streaming audio to text, support **customized vocabularies** for domain-specific terms, integrate with applications through **WebSockets** (an internet-facing interface for two-way communication with an app), and build **subtitles for multiple languages in real time**.

AMAZON TRANSCRIBE USE CASES

- **Medical transcription** — professionals record notes, and Transcribe captures the spoken notes as text.
- **Video subtitles** — generate subtitles automatically, including real-time closed captioning (CC) for a live feed.
- **Streaming content labeling** — capture and label media content, then feed it into Amazon Comprehend for further analysis.
- **Customer call center monitoring** — record service or sales calls and analyze the results for training or strategy.

Amazon Polly is a managed service that converts text into lifelike speech, supporting multiple languages and voices. You can input either plaintext or **Speech Synthesis Markup Language (SSML)**, a markup language that provides special instructions for how the speech should sound — for example, an SSML tag can instruct Polly to insert a pause between two words. Polly can output speech to **MP3, Vorbis, and pulse-code modulation (PCM)** audio stream formats, and it follows a pay-for-use policy that relies on AWS infrastructure to keep costs low. Polly is eligible for use with regulated workloads under **HIPAA** (the U.S. Health Insurance Portability and Accountability Act of 1996) and **PCI DSS** (the Payment Card Industry Data Security Standard).

AMAZON POLLY USE CASES

- **News service production** — generate vocal content directly from written stories.
- **Language training systems** — build systems for learning a new language.
- **Navigation systems** — Polly is embedded in mapping APIs so developers can add voice to geo-based applications.
- **Animation production** — animators add voices to their characters. (Also: mobile newsreaders, games, eLearning, and accessibility apps for the visually impaired.)

COMMON PITFALL

Keep the direction straight. **Transcribe converts speech to text; Polly converts text to speech.** Exam questions swap the two — anchor on the input: if the input is an audio file, the answer is Transcribe; if the input is written text and the output is spoken, the answer is Polly.

6.6 Managed services for text and conversation: Translate, Comprehend, and Lex

Amazon Translate is a fully managed text translation service that uses machine learning to provide high-quality translation on demand. You can develop multilingual user experiences, translate documents into multiple languages, and analyze incoming text in multiple languages. Its strength is composition: Translate is fully integrated with other Amazon ML services, so you can extract named entities, sentiment, and key phrases by integrating with **Amazon Comprehend**, create multilingual subtitles with **Amazon Transcribe**, and speak translated content with **Amazon Polly**. Common use cases are international websites, software localization (a major cost that Translate reduces), multilingual chatbots, and international media management.

Amazon Comprehend uses NLP to extract insights about the content of documents by recognizing the entities, key phrases, language, sentiments, and other common elements in a document. Concretely, it can **extract key entities** such as people or locations, **identify the language** used, **determine the sentiment** — positive, negative, neutral, or mixed — expressed in a document, and **identify the part of speech** for individual words. In effect, Comprehend implements many of the NLP techniques reviewed earlier in this module as a managed service. Use cases include analysis of legal and medical documents, financial fraud detection across very large transaction datasets, large-scale mobile app usage analysis, and content management (tagging content for analysis).

Amazon Lex is an AWS service for building conversational interfaces for applications using voice and text — with Lex, the same conversational engine that powers **Amazon Alexa** is available to any developer. You can build a chatbot that interacts with voice and text to ask questions, get answers, or complete tasks; automatically scale the chatbot with **AWS Lambda** functions on demand; and store log files of conversations for later analysis. Use cases include frontend interfaces for inventory and sales, customer service interfaces, interactive assistants, and querying databases with human-like language. The module's guided lab uses Lex to build a bot that schedules appointments.

TABLE 6.3 The five AWS managed NLP services

SERVICE	CAPABILITY	DIRECTION / REMEMBER
Amazon Transcribe	Recognizes speech in audio and transcribes it to text; custom vocabularies, WebSockets, real-time multilingual subtitles	Speech → text
Amazon Polly	Converts text into lifelike speech from plaintext or SSML; outputs MP3, Vorbis, PCM; HIPAA and PCI DSS eligible	Text → speech
Amazon Translate	High-quality on-demand translation; integrates with Comprehend, Transcribe, and Polly	Language → language
Amazon Comprehend	Extracts insights from text — entities, key phrases, language, sentiment, parts of speech	Insights from text
Amazon Lex	Builds voice and text chatbots using the Amazon Alexa engine; scales with AWS Lambda; logs conversations	Conversation

SECTION 2 KEY TAKEAWAYS

- **Amazon Transcribe** can automatically convert spoken language to text.
- **Amazon Polly** can convert written text to spoken language.
- **Amazon Translate** can create real-time translation between languages.
- **Amazon Comprehend** automates many of the NLP use cases reviewed in this module.
- **Amazon Lex** can create a human-like interface to your applications.

Module summary

- NLP develops computational algorithms to automatically analyze and represent human language; it evaluates language's hierarchy of words → phrases → sentences → ideas, predates machine learning, and maps directly to the ML development process.
- Human language is hard for computers because it is imprecise and context-dependent; the four main challenges are discovering the structure of text, labeling data, representing context, and applying grammar.
- An NLP solution follows a pipeline: formulate the problem and collect and label (tag) data, preprocess the text (remove stop words, normalize via stemming and lemmatization, standardize), tokenize into a DataFrame, then engineer features with bag of words or TF-IDF.
- Bag of words is a vector model that records word frequency and ignores order; in TF-IDF, words that appear across many documents receive a lower weight.
- Text analysis has three categories: classifying text, discovering similarities (edit distance / Levenshtein distance), and deriving relationships (coreference resolution); context is captured by part-of-speech tagging, and NER plus a knowledge graph derive meaning from entities.
- Amazon Transcribe converts speech in audio to text; Amazon Polly converts text to lifelike speech from plaintext or SSML — the two run in opposite directions.
- Amazon Translate provides high-quality on-demand translation and integrates with Comprehend, Transcribe, and Polly; Amazon Comprehend extracts entities, key phrases, language, sentiment, and parts of speech from text.

- Amazon Lex builds conversational voice and text chatbots using the same engine as Amazon Alexa, scales automatically with AWS Lambda, and stores conversation logs; the five services compose into multilingual, spoken, and analyzed pipelines.

Review questions

1. **Why is human language difficult for NLP systems, and what are the four main challenges they must address?**

   Answer: Language is imprecise and context-dependent — the same words carry different meanings depending on surrounding words and inflection. The four challenges are discovering the structure of the text, labeling data (part-of-speech tagging), representing context, and applying grammar.

2. **Walk text through preprocessing: what do removing stop words, normalizing, and standardizing each accomplish?**

   Answer: Removing stop words deletes low-value words not needed for analysis ("this", "is"). Normalizing collapses word forms into a common form using stemming and lemmatization (ran, running → run). Standardizing removes or replaces words not in the dictionary — acronyms, slang, and special characters ("DM" → "direct message").

3. **How does bag of words represent a sentence, and what is its main limitation? What does TF-IDF add?**

   Answer: Bag of words converts a sentence into a vector of word-frequency counts ("is" scores 2 if it appears twice) but ignores word order and meaning. TF-IDF weights words by frequency within a document versus across many documents, so words common to many documents get a lower weight.

4. **What are the three broad categories of text analysis, and one technique for each?**

   Answer: Classifying text (features → ML algorithm → classifier model, e.g., with NLTK); discovering similarities (edit distance / Levenshtein distance for autocorrect and spell check); deriving relationships (coreference resolution).

5. **Describe the three components of a NER model and the role of a knowledge graph.**

   Answer: NER identifies noun phrases (using dependency charts and part-of-speech tagging), classifies the phrases (with a classification algorithm such as Word2Vec), and disambiguates entities using a knowledge graph. The knowledge graph combines subject matter expertise with ML to derive meaning; the Amazon recommendations engine is an example.

6. **Contrast Amazon Transcribe and Amazon Polly, including one distinctive feature of each.**

   Answer: Transcribe converts speech in audio to text (custom vocabularies, WebSockets, real-time multilingual subtitles). Polly converts text to lifelike speech from plaintext or SSML, outputs MP3/Vorbis/PCM, and is HIPAA and PCI DSS eligible. They run in opposite directions.

7. **A company wants to translate an English article into Spanish and analyze the sentiment of reader comments. Which AWS services fit, and how do they compose?**

   Answer: Amazon Translate for the English-to-Spanish translation, and Amazon Comprehend to determine the sentiment (positive, negative, neutral, or mixed) and extract entities and key phrases from the comments. Translate integrates directly with Comprehend (and with Transcribe and Polly).

8. **What makes Amazon Lex able to build sophisticated chatbots, and how does it scale?**

   Answer: Lex provides the same conversational engine that powers Amazon Alexa, letting a chatbot interact by voice and text to ask questions, get answers, or complete tasks. It scales automatically on demand with AWS Lambda functions and can store conversation log files for analysis.