 **Natural Language Processing (NLP):** AI field focused on understanding and responding to written or spoken language.

 **Text Analysis:** NLP process to extract information from unstructured text.

 **Use Cases for NLP:**

* Social media feed analyzer for detecting sentiment in product marketing.
* Document search application for summarizing documents.
* Application for extracting brands and company names from text.

 **Azure AI Language:**

* Cloud-based service for text understanding and analysis.
* Features include sentiment analysis, key phrase identification, text summarization, and conversational language understanding.

**General Principles and Common Techniques in NLP**

* **Statistical Analysis of Text:**
  + **Corpus Analysis:** Analyzing a body of text to infer semantic meaning by identifying commonly used words.
  + **Frequency Analysis:** Counting the occurrences of each token to determine the main subject of a text.
* **Tokenization:**
  + **Definition:** Breaking down a text into tokens, which can be words, partial words, or combinations of words and punctuation.
  + **Example:** "we choose to go to the moon" tokenized as [1,2,3,4,3,5,6].
* **Tokenization Techniques:**
  + **Text Normalization:** Removing punctuation, converting to lower case to improve performance but may lose some semantic meaning.
  + **Stop Word Removal:** Excluding common words (e.g., "the", "a") that add little meaning to focus on important words.
  + **n-grams:** Considering multi-term phrases (e.g., "I have", "he walked") to enhance understanding.
  + **Stemming:** Consolidating words with the same root (e.g., "power", "powered", "powerful") as the same token.
* **Frequency Analysis:**
  + **Single Document Analysis:** Counting token occurrences to infer the main subject.
  + **Multiple Documents:** Using techniques like Term Frequency-Inverse Document Frequency (TF-IDF) to determine token relevance across documents.
* **Machine Learning for Text Classification:**
  + **Classification Algorithms:** Using algorithms like logistic regression to classify text based on categories.
  + **Sentiment Analysis Example:** Training a model with labeled reviews to classify text as positive or negative based on tokenized features.
* **Semantic Language Models:**
  + **Embeddings:** Encoding language tokens as vectors (multi-valued arrays) to represent semantic relationships.
  + **Vector Representation:** Tokens occupy specific locations in multidimensional space, with related words grouped closer together.
  + **Advanced Language Models:** Using high-dimensional vectors and sophisticated methods to calculate embeddings for NLP tasks.
* **Common NLP Tasks Supported by Language Models:**
  + **Text Analysis:** Extracting key terms or identifying named entities.
  + **Sentiment Analysis:** Categorizing text as positive or negative.
  + **Machine Translation:** Automatically translating text between languages.
  + **Summarization:** Summarizing the main points of a large text.
  + **Conversational AI:** Developing bots or digital assistants that interpret natural language and respond appropriately.
* **Azure AI Language Service:**
  + **Capabilities:** Supports various NLP tasks such as text analysis, sentiment analysis, machine translation, summarization, and conversational AI.

**Text Analysis with Azure AI Language**

**Overview**

Azure AI Language provides advanced natural language processing for unstructured text. Its text analysis features include:

* **Named Entity Recognition:** Identifies people, places, events, etc., and allows customization to extract custom categories.
* **Entity Linking:** Links identified entities to Wikipedia for disambiguation.
* **PII Detection:** Identifies personal and sensitive information.
* **Language Detection:** Identifies the language and returns a language code (e.g., "en" for English).
* **Sentiment Analysis:** Identifies positive or negative sentiment.
* **Summarization:** Summarizes text by extracting key information.
* **Key Phrase Extraction:** Lists main concepts from unstructured text.

 **Question Answering in AI:**

* Supports automated conversational AI workloads.
* Used to build bot applications for responding to customer queries.

 **Capabilities:**

* Immediate response and accurate answers.
* Natural multi-turn interactions.

 **Platform Implementation:**

* Bots can be deployed on websites or social media platforms.

 **Benefits:**

* Provides friendly, timely answers outside office hours.
* Escalates to human support for complex queries.

 **Core Services for User Support Bot:**

* **Azure AI Language:** Custom question answering feature for creating a knowledge base with Q&A pairs.
* **Azure AI Bot Service:** Framework for developing, publishing, and managing bots on Azure.

 **Creating a Custom Question Answering Knowledge Base:**

* **Tools:** Use Azure AI Language Studio to create, train, publish, and manage question answering projects.
* **Options:** Write code using Azure AI Language REST API or SDK, but Language Studio is recommended for ease.
* **Initial Step:** Provision a Language resource in your Azure subscription.

 **Defining Questions and Answers:**

* **Methods:** Generate from existing FAQ documents/web pages or enter/edit manually.
* **Combining Techniques:** Often start with existing data and extend with manual entries.
* **Alternative Phrasing:** Add different phrasings for the same question to improve matching.

 **Testing the Project:**

* **Process:** Save the Q&A set, analyze, and apply NLP models for matching.
* **Testing Tool:** Use Language Studio's test interface to submit questions and review answers.

 **Building a Bot with Azure AI Bot Service:**

* **Creating the Bot:** Use the Microsoft Bot Framework SDK or automatic bot creation in Language Studio.
* **Publishing:** Deploy the bot as an Azure AI Bot Service application.

 **Connecting Channels:**

* **Multiple Channels:** Enable interactions through web chat, email, Microsoft Teams, etc.
* **User Interaction:** Users can ask questions and receive answers from the knowledge base via any connected channel.

**Core Concepts of Conversational Language Understanding**

**1. Utterances**

* **Definition:** Examples of what a user might say, which the application must interpret.
* **Examples:**
  + "Switch the fan on."
  + "Turn on the light."

**2. Entities**

* **Definition:** Items referred to in an utterance.
* **Examples:**
  + **Entities in Utterances:**
    - "Switch the fan on." (fan)
    - "Turn on the light." (light)
  + **General Entity:** Fan and light as instances of a general "device" entity.

**3. Intents**

* **Definition:** The purpose or goal expressed in a user's utterance.
* **Examples:**
  + For "Switch the fan on" and "Turn on the light," the intent is to turn a device on.
  + **Example Intents in a Home Assistant Application:**
    - **Greeting:**
      * "Hello"
      * "Hi"
      * "Hey"
      * "Good morning"
    - **TurnOn:**
      * "Switch the fan on" (fan)
      * "Turn the light on" (light)
      * "Turn on the light" (light)
    - **None:**
      * "What is the meaning of life?"
      * "Is this thing on?"

**Application Workflow**

1. **Define Entities and Intents:**
   * Create a model consisting of various intents and entities.
   * Use utterances to train the model to identify the most likely intent and entities based on input.
2. **Training the Model:**
   * Use sample utterances for each intent.
   * Include a "None" intent to handle utterances that don't match any specific intent.
3. **Using the Model:**
   * Train the model to predict intents and entities from user input.
   * Implement the model in a client application to retrieve predictions and respond appropriately.

**Special Notes:**

* **None Intent:**
  + Acts as a fallback to handle unmatched utterances.
  + Provides generic responses to user queries that don't match any defined intents.

**Getting Started with Conversational Language Understanding in Azure**

**Azure Resources for Conversational Language Understanding**

* **Azure AI Language:**
  + Enables building apps with natural language understanding capabilities.
  + Suitable for authoring and prediction.
* **Azure AI Services:**
  + Includes conversational language understanding along with other Azure AI services.
  + Used solely for prediction.

**Authoring a Model**

1. **Define Entities, Intents, and Utterances:**
   * Entities: Items referred to in user utterances.
   * Intents: Purpose or goal expressed in user utterances.
   * Utterances: Examples of user input.
2. **Prebuilt Domains:**
   * Includes pre-defined intents and entities for common scenarios.
   * Can be used as a starting point or create custom ones.
3. **Authoring Interface:**
   * Utilize the Language Studio, a web-based interface for model creation.

**Training the Model**

1. **Iterative Process:**
   * Define intents, entities, and sample utterances.
   * Train the model using sample utterances to match natural language expressions.
   * Test the model for accuracy and make updates as needed.
   * Repeat training and testing until satisfied with results.

**Predicting**

1. **Publishing the Model:**
   * Once satisfied with training and testing, publish the model to a prediction resource.
2. **Client Application Usage:**
   * Client applications connect to the prediction resource's endpoint.
   * Authenticate using appropriate key.
   * Submit user input to get predicted intents and entities.
   * Act on predicted intent based on returned predictions.

**Speech Recognition:**

* **Process:**
  + Converts spoken words into processable data, often transcribing it into text.
  + Analyzes speech patterns in audio to recognize recognizable patterns mapped to words.
* **Models Used:**
  + Acoustic Model: Converts audio signal into phonemes (representations of specific sounds).
  + Language Model: Maps phonemes to words, predicting probable word sequences based on phonemes.
* **Applications:**
  + Providing closed captions for videos.
  + Creating transcripts of phone calls or meetings.
  + Automated note dictation.
  + Interpreting user input for further processing.

**Speech Synthesis:**

* **Process:**
  + Vocalizes data, usually by converting text to speech.
  + Requires text to be spoken and voice selection for vocalization.
* **Steps:**
  + Tokenizes text into words.
  + Assigns phonetic sounds to each word.
  + Breaks phonetic transcription into prosodic units (phrases, clauses, sentences).
  + Converts phonemes to audio format.
* **Applications:**
  + Generating spoken responses to user input.
  + Creating voice menus for telephone systems.
  + Reading email or text messages aloud.
  + Broadcasting announcements in public locations.

**Azure AI Speech Service**

**APIs:**

* **Speech to Text API:**
  + Converts audio into text format.
  + Supports real-time and batch transcription.
  + Real-time transcription for scenarios like presentations, demos, etc.
  + Batch transcription for audio recordings stored on various platforms.
* **Text to Speech API:**
  + Converts text into audible speech.
  + Speech synthesis voices include pre-defined options for multiple languages and regional pronunciation.
  + Supports custom voices for personalized speech synthesis.

**Azure Resources:**

* **Speech Resource:**
  + For standalone use of Azure AI Speech.
  + Allows separate management of access and billing.
* **Azure AI Services Resource:**
  + For combined use with other Azure AI services.
  + Enables centralized management of access and billing for multiple services.

**Real-time Transcription:**

* **Scenario:**
  + Ideal for presentations, demos, or live events.
  + Transcribes text from real-time audio streams.

**Batch Transcription:**

* **Scenario:**
  + Suitable for processing audio recordings stored in various locations.
  + Transcribes text asynchronously from audio files using SAS URI.

**Speech Synthesis Voices:**

* **Variety:**
  + Offers multiple pre-defined voices supporting various languages and regional pronunciations.
  + Includes neural voices for more natural intonation.
  + Supports customization for developing and using custom voices.

.