IBM Watson and Cognitive Computing

IBM Watson and Cognitive Computing

1. IBM Watson Overview

- Cloud-based cognitive computing platform.
- o Known for defeating human champions in *Jeopardy!* using Al.
- Uses hundreds of language-analysis algorithms to process vast amounts of data (e.g., 200M pages, including Wikipedia).
- Trained using machine learning and reinforcement learning.

2. Cognitive Computing

- Simulates human brain functions like pattern recognition and decision-making.
- Learns as it processes more data.
- Used in diverse real-world applications (e.g., healthcare, customer service, fraud detection).

3. Watson Services (Lite Tier)

- Free tier available for experimentation ("no credit card required").
- o Services include:
 - **Speech to Text** Converts spoken audio to text.
 - **Text to Speech** Synthesizes speech from text (supports multiple languages).
 - Language Translator Translates text between languages and identifies languages.
 - Natural Language Understanding Analyzes sentiment, emotion, and keywords.
 - Visual Recognition Identifies objects, faces, and text in images/videos.
 - Watson Assistant Builds chatbots and virtual assistants.
 - Personality Insights Analyzes personality traits from text.
 - Tone Analyzer Detects emotional and social tones in text.

4. IBM Cloud Account Setup

- Required to access Watson services.
- Free tier available with usage limits.
- o Dashboard: IBM Cloud Console.

5. Watson Developer Cloud Python SDK

- o Enables programmatic access to Watson services.
- o Install via:

bash			
Сору			

o Supports synchronous and asynchronous APIs.

6. Traveler's Companion Translation App (Case Study)

- o Mashup of three Watson services:
 - 1. **Speech to Text** Transcribes spoken English/Spanish.
 - 2. **Language Translator** Translates between English and Spanish.
 - 3. **Text to Speech** Converts translated text back to speech.
- o Steps:
 - Record a question (English) → Transcribe → Translate → Speak (Spanish).
 - Record a response (Spanish) → Transcribe → Translate → Speak (English).
- o Requires API keys for each service (stored in keys.py).

2. Additional Watson Tools

- Watson Studio For data science, machine learning, and deep learning projects.
- Knowledge Studio Custom model training for domainspecific applications.
- Watson Machine Learning Integrates with TensorFlow, Keras, and scikit-learn.
- Cognos Analytics Al-powered business intelligence and visualization.

3. Key Terms

- o **Mashup** Combining multiple services into one application.
- SSML (Speech Synthesis Markup Language) Controls speech synthesis (e.g., pitch, pauses).
- Cognitive Computing Al systems that mimic human reasoning.

1. Top-Down Design Approach

Problem Breakdown:

o **Top-level goal**: Create a translator app for English/Spanish communication.

o Refinement:

- 1. Translate English speech → Spanish speech.
- 2. Translate Spanish speech → English speech.

o 10-Step Implementation:

- Steps 1–5: English → Spanish (record, transcribe, translate, synthesize, play).
- Steps 6-10: Spanish → English (repeat process).

2. Watson Service Classes

- SpeechToTextV1:
 - o Transcribes audio files to text (e.g., english.wav → English text).
 - o Uses predefined models:
 - en-US BroadbandModel (English).
 - es-ES BroadbandModel (Spanish).
 - JSON Response: Contains nested results → alternatives → transcript.

LanguageTranslatorV3:

- o Translates text between languages.
- o Predefined models:
 - en-es (English → Spanish).
 - es-en (Spanish → English).
- o **JSON Response**: translations list with translation key.
- TextToSpeechV1:
 - o Converts text to speech (e.g., Spanish text → spanish.wav).
 - o Voices:
 - en-US AllisonVoice (English).
 - es-US SofiaVoice (Spanish).

3. Key Functions

speech_to_text(file_name, model_id)

- Process:
 - 1. Opens audio file ('rb' mode for binary read).
 - 2. Calls Watson's recognize method with:
 - audio: File object.
 - content type: 'audio/wav'.
 - model: Language model (e.g., en-US_BroadbandModel).
 - 3. Parses ISON response to extract transcript.

translate(text_to_translate, model)

- Process:
 - 1. Calls Watson's translate method with:
 - text: String to translate.
 - model id: Translation model (e.g., en-es).
 - 2. Returns translated text from ISON (translations[0]['translation']).

text_to_speech(text_to_speak, voice_to_use, file_name)

- Process:
 - 1. Opens output file ('wb' mode for binary write).
 - 2. Calls Watson's synthesize method with:
 - text: String to speak.
 - accept: 'audio/wav'.
 - voice: Voice model (e.g., es-US_SofiaVoice).
 - 3. Writes audio bytes to file.

record audio(file name)

- Uses PyAudio:
 - o Configures audio stream (44.1 kHz, 16-bit, stereo).
 - Records 5 seconds of audio in chunks (1024 frames per chunk).
 - o Saves as WAV file using wave module.

play_audio(file_name)

- Uses PyDub:
 - o Loads WAV file with AudioSegment.from wav().
 - o Plays audio with pydub.playback.play().

4. Technical Details

- Audio Configuration:
 - o **Frame rate**: 44.1 kHz (CD quality).
 - o **Chunk size**: 1024 frames per read.
 - o **Format**: 16-bit integers (pyaudio.paInt16).
 - o Channels: 2 (stereo).

- File Handling:
 - o **WAV files**: Saved/read in binary mode ('wb'/'rb').
 - o JSON Parsing:
 - Speech-to-text: result['results'][0]['alternatives'][0] ['transcript'].
 - Translation: translated text['translations'][0]['translation'].

5. Key Terms

- Media type: audio/wav (formerly MIME type).
- **Divide-and-conquer**: Breaking the app into 10 manageable steps.
- **Mashup**: Combining Watson services (Speech-to-Text, Translator, Text-to-Speech).

Self-Check Answers

- 1. False (Watson returns JSON, not XML).
- 2. en-es (English-to-Spanish model).
- 3. **Media** type (audio/wav).
- 4. **False** (44.1 kHz is CD quality, not BluRay).
- 5. **content**: Bytes of the synthesized audio file.
- 6. **model_id**: Specifies the translation model (e.g., en-es).

1. Watson Developer Resources

- Documentation Hub:
 - o Watson Services Documentation provides:
 - Getting started tutorials.
 - Video overviews and demos.
 - API references (Python, Java, etc.).
 - Sample apps and advanced guides.
- SDKs & Tools:

- o SDKs for multiple languages/platforms:
 - Python (used in this chapter), Java, Node.js, etc.
 - List: Watson SDKs.

Learning Resources:

- o Blogs, GitHub repos, and Watson YouTube channel.
- Code Patterns: Reusable solutions for complex problems (some in Python).

IBM Redbooks:

- o In-depth guides on Watson services:
 - Essentials of Application Development on IBM Cloud.
 - Building Cognitive Applications with IBM Watson Services (7 volumes covering Assistant, Visual Recognition, Language Translator, etc.).

2. Hands-On Exercises

Core Watson Services Practice

- 1. Speech-to-Text:
 - o Record audio, test transcription accuracy (<u>Demo</u>).
 - Detect multiple speakers in conversations.

2. Visual Recognition:

o Identify objects in photos using the <u>Visual Recognition demo</u>.

3. Language Translator:

- o Enhance the Traveler's Companion app:
 - Add bilingual prompts (Exercise 14.4).
 - Experiment with voices (Exercise 14.5).
 - Extend to new languages (Exercise 14.6).
- o **Challenge**: Chain translations (e.g., English → Spanish → French → German → Italian → English) to observe accuracy loss (Exercise 14.7).

4. Text-to-Speech (TTS):

o Use SSML for voice control (pitch, pauses, etc.) (Exercises 14.14–14.15).

Project-Based Learning

- **Pizza Parlor Chatbot** (Exercise 14.8):
 - Combine Speech-to-Text and Text-to-Speech for voice-driven pizza orders.
 - Advanced: Integrate Watson Assistant for a conversational interface.
- **Evesight Tester** (Exercise 14.13):
 - o Simulate a Snellen chart; validate spoken letter responses with Watson.
- Metric Conversion App (Exercise 14.21):
 - Use speech recognition/synthesis to convert units (e.g., meters to inches).
- Voice-Driven Text Editor (Exercise 14.22):
 - Accessibility-focused app for hands-free text editing via voice commands.

Advanced Challenges

- **Real-Time Audio Handling** (Exercise 14.18):
 - Use PyAudio to detect speech start/stop for dynamic recording.
- **Sentiment Analysis** (Exercise 14.23):
 - Compare Watson's Natural Language Understanding with manual NLP techniques.

3. Key Tools & Techniques

- Speech Synthesis Markup Language (SSML):
 - o Customize voice output (e.g., singing "Happy Birthday" in Exercise 14.15).
 - o Docs: W3C SSML Spec.
- Watson Assistant:
 - Build chatbots for customer service, FAQs, etc. (Exercises 14.19-14.20).
- IoT Integration:
 - Explore live sensor data with <u>Watson IoT Platform</u> (Exercise 14.10).

4. Entrepreneurial Applications

- **Bot Development** (Exercise 14.20):
 - o Automate call centers, healthcare Q&A, travel planning, etc.
 - o Bots can route calls to humans when needed.

Self-Check Answer

• **Code patterns**: IBM's reusable solutions for complex programming challenges.