

1. Understanding Intents and Entities (หน้า 2)

- ความหมายของ Intents และ Entities:
 - Intents หมายถึงวัตถุประสงค์หรือความต้องการของผู้ใช้ เช่น การค้นหาร้านอาหารหรือการถามชื่อ
 - Entities หมายถึงข้อมูลเฉพาะในประโยคที่เกี่ยวข้องกับ Intents เช่น ประเภทอาหาร สถานที่ หรือวันที่
 - การใช้งาน: แหบทบจะต้องเข้าใจ "Intent" ของผู้ใช้เพื่อระบุความต้องการ จากนั้นจะสกัดข้อมูลสำคัญ (Entities) ออกมา เช่น ชนิดของอาหาร หรือสถานที่ตั้ง
 - ตัวอย่าง: ผู้ใช้พูดว่า “ฉันกำลังหาฟาสต์ฟู้ดอเมริกันในใจกลางเมือง”
 - Intent: restaurant_search
 - Entities: Cuisine: ฟาสต์ฟู้ดอเมริกัน, Area: ใจกลางเมือง
 - ความสำคัญ: Intents และ Entities ช่วยให้แหบทบสามารถเข้าใจและตอบสนองได้ตรงตามความต้องการของผู้ใช้มากขึ้น
-

2. Intents (หน้า 4-5)

- ความหมายของ Intents: เป็นคำอธิบายที่กว้างๆ เกี่ยวกับสิ่งที่ผู้ใช้ต้องการ เช่น คำว่า “hello”, “hi” อาจถูกจัดว่าเป็น greet intent ซึ่งหมายถึงการทักทาย
 - การใช้งาน: การจัดกลุ่ม Intents ช่วยให้บอทสามารถเข้าใจได้ว่าแต่ละประโยคมีความหมายว่าอย่างไร
 - ตัวอย่าง:
 - ประโยค “I want to take my friend out for sushi” สามารถถูกจัดว่าเป็น restaurant_search intent หรือถ้าแหบทบสามารถจองโต๊ะได้ ประโยคนี้อาจจะจัดว่าเป็น request_booking intent
 - ข้อดี: ช่วยให้บอทเข้าใจและสามารถตอบสนองได้ตามความต้องการของผู้ใช้
 - ข้อเสีย: การจัดกลุ่ม Intents ต้องออกแบบให้ครอบคลุมทุกความต้องการ มิฉะนั้นบอทอาจจะไม่เข้าใจ
-

3. Entities (หน้า 6)

- ความหมายของ Entities: หมายถึงข้อมูลเฉพาะในประโยค เช่น วันที่ ชนิดอาหาร หรือสถานที่ ซึ่งเป็นข้อมูลที่สามารถสกัดออกมาเพื่อใช้ในการตอบสนอง
- การใช้งาน: ใช้ในระบบเช่นการค้นหาร้านอาหารที่ต้องการข้อมูลเฉพาะ เช่น “วันที่ 10 มิถุนายน” หรือ “ร้านซูชิในนิวยอร์ก”

- ความสำคัญ: ช่วยให้แชทบอทสามารถจัดการข้อมูลเฉพาะได้
 - ข้อดี: ทำให้การโต้ตอบมีความแม่นยำมากขึ้นเมื่อบอทสามารถเข้าใจข้อมูลสำคัญได้
 - ข้อเสีย: ต้องการการออกแบบและการจับคู่ข้อมูลที่ถูกต้องเพื่อให้การทำงานเป็นไปอย่างราบรื่น
-

4. Using Regular Expressions (หน้า 7-8)

- การใช้งาน Regular Expressions (regex): ใช้ regex ในการค้นหาคำที่เกี่ยวข้องในข้อความ เพื่อช่วยบอพระบุ Intent และ Entities
- การทำงาน: ใช้คำสั่ง `re.search()` เพื่อจับคู่คำที่ต้องการ เช่น "hello", "hi", "hey" ในข้อความ
- ตัวอย่าง:

python

Copy code

```
re.search(r"\b(hello|hey|hi)\b", "hey there!") # ผลลัพธ์: ไม่เท่ากับ None
```

```
re.search(r"\b(hello|hey|hi)\b", "which one?") # ผลลัพธ์: เท่ากับ None
```

5. Intent Classification with Regex (หน้า 10-14)

- การทำงานของการจัดกลุ่ม Intent ด้วย Regex: สร้างดิกชันนารีของคีย์เวิร์ดที่ใช้ใน Intent เช่น "hello", "hi" ใน Intent greet
 - การใช้งาน: ใช้ฟังก์ชัน `re.compile()` เพื่อสร้างแพทเทิร์นจากดิกชันนารีและจับคู่ข้อความกับ Intents
 - ข้อดี: วิธีนี้ง่ายและรวดเร็วในการจัดการกับข้อความที่มีรูปแบบชัดเจน
 - ข้อเสีย: จำกัดสำหรับการตอบสนองที่ซับซ้อนมากกว่า เช่น การเรียนรู้จากการสนทนาก่อนหน้า
-

6. Entity Extraction with Regex (หน้า 15-18)

- การสกัด Entities ด้วย regex: ใช้ regex เพื่อค้นหาชื่อของบุคคลจากประโยค เช่น "my name is David Copperfield"
-

7. Building a Virtual Assistant (หน้า 19-35)

- การสร้าง Virtual Assistant (VA): ตัวช่วยอัจฉริยะที่สามารถทำงานหรือให้บริการตามคำสั่ง เช่น การถามคำถาม หรือการควบคุมอุปกรณ์อัตโนมัติในบ้าน
 - การทำงาน: Virtual Assistant ที่จะสามารถเข้าใจ Intent และ Entities รวมถึงบริบทของคำถามหรือการสนทนาได้
 - การใช้งาน: ใช้คำสั่งเสียงหรือข้อความเพื่อสั่งงาน เช่น ควบคุมอุปกรณ์ในบ้าน หรือค้นหาข้อมูลในอินเทอร์เน็ต
 - โมดูลที่ใช้: Python มีไลบรารีหลายตัวที่ใช้ในการสร้าง VA เช่น SpeechRecognition สำหรับการจดจำเสียง และ Pyttsx3 สำหรับการแปลงข้อความเป็นเสียง
-

8. The Audio Input Function (หน้า 29-30)

- การรับคำสั่งด้วยเสียง: ฟังก์ชัน `audioinput()` ใช้ในการรับและประมวลผลเสียงจากไมโครโฟน แล้วใช้ Google API เพื่อแปลงเสียงเป็นข้อความ
 - การใช้งาน: เมื่อผู้ใช้พูด เช่น “What is your name?” ระบบจะประมวลผลและตอบกลับด้วยข้อความที่ตั้งไว้
-

9. Functions for Day, Time, and Web Search (หน้า 32-35)

- ฟังก์ชันบอกวันและเวลา: ฟังก์ชัน `theDay()` และ `theTime()` ใช้ในการบอกวันและเวลาปัจจุบัน
- การเปิดเว็บไซต์: ใช้โมดูล `webbrowser` เพื่อเปิดเว็บไซต์ เช่น Google หรือ Wikiped