สำนักงานวิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

การทดสอบประสิทธิภาพการสื่อสารกับหุ่นยนต์สนทนา

เมื่อรวมตัวแปรคำสั่ง (Interpreter) เข้ากับโปร แกรมตัดคำและส่วนแก้ไขคำผิดอัตโนมัติแล้ว จะได้หุ่นยนต์สนทนาที่มีความยืดหยุ่นต่อการพิมพ์ผิดแบบไม่ตั้งใจได้ คือ แม้ว่าจะมีการพิมพ์ผิดเพียงเล็กน้อยหุ่นยนต์ก็จะสามารถโต้ตอบได้อย่างมีประสิทธิภาพ โดยหาองค์ประกอบมาช่วยคาดเดาคำผิดให้หุ่นยนต์สามารถคาดเดาคำผิดได้เหมือนมนุษย์ ซึ่งได้มาจากการสังเกตพฤติกรรมการพิมพ์ดังที่ได้แสดงในบทที่ 3

การทดสอบประสิทธิภาพการสื่อสารกับหุ่นยนต์สนทนาทำได้โดย เริ่มจากสร้างฐานความรู้เตรียมไว้ก่อน (ซึ่งประโยคที่ใช้ในการสร้างกฎ แสดงไว้ในภาคผนวก ค) เพื่อเตรียมไว้ใช้ในการทดสอบผลการจับคู่แพทเทิร์น หลังจากนั้นทำการเปรียบเทียบว่าเมื่อมีการแก้ไขคำผิดแล้วสามารถจับคู่ได้ประสิทธิภาพดีขึ้นกว่าเดิมมากเท่าใดในรูปของเปอร์เซ็นต์ (ซึ่งตัวอย่างประโยคสนทนาที่มีคำผิดแบบไม่ตั้งใจที่นำมาใช้ในการทดสอบทั้งหมดนั้น ได้แสดงไว้ในภาคผนวก ข) โดยมีรายละเอียดของแต่ละขั้นตอน ดังนี้

5.1 การเลือกข้อความที่นำมาทดสอบ

ข้อความที่นำมาทดสอบจะเป็นข้อความที่นำมาจากการสนทนาจริงของผู้ใช้ผ่านทางโปรแกรมเอ็มเอสเอ็นแมสเซนเจอร์ (msn Messenger) โดยเก็บมาจากบันทึกบทสนทนาที่เก็บไว้ในเครื่องคอมพิวเตอร์จากอาสาสมัคร (ซึ่งเป็นนิสิตและอาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย) จำนวน 30 คน ซึ่งสุ่มเลือกเฉพาะประโยคที่มีคำผิดมาคนละ 4 ประโยค (ประโยคทั้งหมดที่เลือกมา แสดงไว้ในภาคผนวก ข) และประโยคที่เลือกมานั้นต้องเกิดจากการประกอบกันของคำที่มีความหมายในพจนานุกรมเท่านั้น

ประโยคที่มีคำผิดนั้นจะต้องเป็นประโยคที่มีคำผิดซึ่งมีระยะแก้ไขเป็น 1 และเกิดจากการพิมพ์ผิดแบบไม่ตั้งใจเท่านั้น (แต่ไม่รวมกรณีพิมพ์ตก) เนื่องจากอัลกอทึมการแก้ไขคำผิดในงานวิจัยนี้ไม่ได้ครอบคลุมถึงกรณีที่มีระยะแก้ไขมากกว่าหนึ่งพิมพ์และกรณีที่เกิดจากการพิมพ์ตกไว้ ดังที่เคยได้กล่าวไปแล้ว นอกจากนี้ประโยคที่มีคำผิดนั้นจะต้องยังคงให้ขอบเขตของคำผิดได้ถูกต้องเมื่อนำไปผ่านโปรแกรมตัดคำแล้ว (เพราะอัลกอริทึมแก้ไขคำผิดนั้นทำงานหลังการตัดคำ) *และการแก้ไขคำผิดนั้นจะไม่ครอบคลุมถึงภาษาแชท (Chat) ที่วัยรุ่นนิยมใช้ เช่น เดี๋ยว จงใจพิมพ์ผิดเป็น เด๋ว* (แต่ถ้าต้องการเพิ่มศัพท์หรือคำใหม่ๆ ลงไปในพจนานุกรมที่ใช้อยู่ ก็สามารถเพิ่มลงไปในพจนานุกรมเดิมที่เก็บอยู่ในรูปของแฟ้มข้อความได้ทันที โดยไม่ต้องทำการแก้ไขโค้ดโปรแกรมส่วนอื่น ทำให้หุ่นยนต์สามารถรู้จักคำศัพท์ใหม่ๆ โดยไม่มองว่าคำนั้นเป็นคำผิดอีกต่อไป)

5.2 การสร้างฐานความรู้ให้กับหุ่นยนต์สนทนา

เริ่มจากสร้างชุดกฎคำถาม-คำตอบ เตรียมไว้ก่อนเป็นจำนวน 120 กฎ ซึ่งนำมาจากบันทึกบทสนทนาของโปรแกรมเอ็มเอสเอ็นแมสเซนเจอร์ (ดังรายละเอียดในข้อ 5.1) โดยที่จะนำประโยคที่มีคำผิดจากภาคผนวก ข มาทำให้เป็นประโยคที่ถูกต้องไม่มีคำผิดเสียก่อน ดังที่แสดงไว้ในภาคผนวก ค แล้วสมมติคำตอบให้เข้ากับคำถามนั้นขึ้นมาเอง ดังในตัวอย่างที่ 5.1

ตัวอย่างที่ 5.1

1. สมมติว่าประโยคที่ได้มาจากบทสนทนาจริงของผู้ใช้ คือ

เรียนเสร้จแล้วหรือจ๊ะ

2. นำมาแก้ไขให้เป็นประโยคที่ถูกต้อง ได้เป็น

เรียนเสร็จแล้วหรือจ๊ะ

3. สมมติคำตอบให้เข้ากับคำถามขึ้นมาเอง (เพื่อเตรียมไว้นำไปสร้างกฎ) ดังในตัวอย่างนี้ได้เป็น

เสร็จแล้วจ้ะ

เมื่อเตรียมคำถาม-คำตอบไว้ครบแล้ว ต่อมาก็จะนำข้อความแต่ละชุดมาสร้างกฎหรือในภาษาเอไอเอ็มแอลเรียกว่าแคตากอรี เพื่อเตรียมไว้เป็นฐานความรู้ให้กับหุ่นยนต์สนทนา *แต่คำถามต้องเขียนเหมือนรูปแบบที่ได้จากการตัดคำด้วยโปรแกรม [5]* ดังในตัวอย่าง 5.2 เป็นตัวอย่างการเขียนคำถาม-คำตอบให้อยู่ในรูปแบบของภาษาเอไอเอ็มแอล

ตัวอย่างที่ 5.2 จากข้อความในตัวอย่างที่ 5.1 สามารถนำมาสร้างกฎได้ดังนี้

```
<category>
<pattern>เรียน เสร็จ แล้ว หรือ จ๊ะ</pattern>
<template>เสร็จแล้วจ้ะ</template>
</category>
```

การเขียนแพทเทิร์นของแต่ละแคตากอรีในฐานความรู้เอไอเอ็มแอลภาษาไทย ต้องมีการเว้นวรรคระหว่างคำในประโยคเองเหมือนรูปแบบที่ได้จากการตัดคำด้วยโปรแกรม SWATH [5] ซึ่งมีการเว้นวรรคระหว่างคำในประโยคโดยใช้อักขระว่าง (Space character) แต่การเขียนเทมเพลตหรือคำตอบนั้นไม่จำเป็นต้องมีการเว้นวรรคระหว่างคำในประโยค

สัญลักษณ์คั่นระหว่างคำนั้นเป็นคุณสมบัติสำคัญที่ ALICE ต้องพึ่งพาในการแยกแยะวิธีการโต้ตอบกลับ ดังนั้นภาษาที่ไม่มีสัญลักษณ์คั่นระหว่างคำในประโยคจึงจำเป็นที่จะต้องมีการตัดคำเสียก่อน จึงจะทำให้สามารถนำ ALICE และเอไอเอ็มแอล ไปใช้สร้างหุ่นที่โต้ตอบด้วยภาษาไทยได้

5.3 การทดสอบสื่อสารกับหุ่นยนต์สนทนา

หลังจากเตรียมชุดกฎคำถาม-คำตอบ ไปสร้างไว้เป็นฐานความรู้เป็นจำนวน 120 กฎ เรียบร้อยแล้ว จากนี้หุ่นยนต์ก็พร้อมที่จะสนทนา แต่สามารถสนทนาโต้ตอบได้เพียง 120 ชุด คำถาม (ตามประโยคในภาคผนวก ค) เท่านั้น

ความสามารถในการพูดคุยโต้ตอบนั้น จะขึ้นอยู่กับข้อมูลในไฟล์ AIML เป็นหลัก ซึ่งผู้ใช้สามารถที่จะเพิ่มข้อมูลการพูดคุย และรูปแบบประโยคสนทนาใหม่ๆ เข้าไปในฐานข้อมูล AIML ได้ตามต้องการ

บทสนทนาแต่ละบทจะถูกกำหนดด้วยแคตากอรี (category) เมื่อ A.L.I.C.E Bot รับประโยคพูดคุยเข้ามา จะนำประโยคนั้นมาเปรียบเทียบ (Matching) กับฐานข้อมูล AIML หากพบว่าตรงกับบทสนทนาอันไหน ก็จะเอาคำตอบของบทสนทนานั้นส่งออกไป ดังตัวอย่าง 5.2 ถ้าหากข้อความรับเข้าเป็น "เรียนเสร็จแล้วหรือจ๊ะ" ก็จะถูกจับคู่กับคำตอบ "เสร็จแล้วจ้ะ" ส่งเป็นผลลัพธ์ออกไป

A.L.I.C.E Bot [2] ที่มีให้ดาวน์โหลดได้ทางอินเตอร์เน็ตนั้นมีอยู่ด้วยกันหลายเวอร์ชัน ใช้ได้กับหลายๆ ภาษา แต่เวอร์ชันที่ใช้ในงานวิจัยชิ้นนี้ ได้เลือกใช้ A.L.I.C.E Bot - Program D เนื่องจากเป็นเวอร์ชันที่เพิ่งได้รับการพัฒนาขึ้นมาใหม่ที่สุด และมีความสามารถในการทำให้เป็นเซิร์ฟเวอร์ได้ ซึ่งทั้งหมดถูกเขียนขึ้นโดยใช้ภาษาจาวา เพราะฉะนั้นเครื่องที่จะรัน A.L.I.C.E Bot เวอร์ชันนี้ได้จึงจำเป็นที่จะต้องลงจาวาเวอร์ชวลแมชชีน (Java Virtual Machine) เสียก่อน

จากนั้นเริ่มการทำงานของ A.L.I.C.E Bot (ที่งานวิจัยนี้ได้เพิ่มเติมเรื่องการจับคู่อักษรที่เป็นไทยและได้เพิ่มเติมคุณสมบัติเรื่องการแก้ไขคำผิดลงไปเรียบร้อยแล้ว) นั้นขึ้นมา หน้าจอจะมีข้อความดังรูปที่ 5.1

เมื่อขึ้นหน้าจอดังรูปที่ 5.1 แล้ว A.L.I.C.E Bot ทำงานเป็นเซิร์ฟเวอร์พร้อมแล้วสำหรับการติดต่อที่พอร์ตหมายเลข 2001 การติดต่อกับ A.L.I.C.E Bot สามารถทำได้โดยสนทนาผ่านเบราว์เซอร์โดยจำลองตนเองเป็นเซิร์ฟเวอร์แล้วระบุพอร์ตไปด้วย ซึ่งเขียนเป็นยูอาร์แอลได้เป็น http://localhost:2001

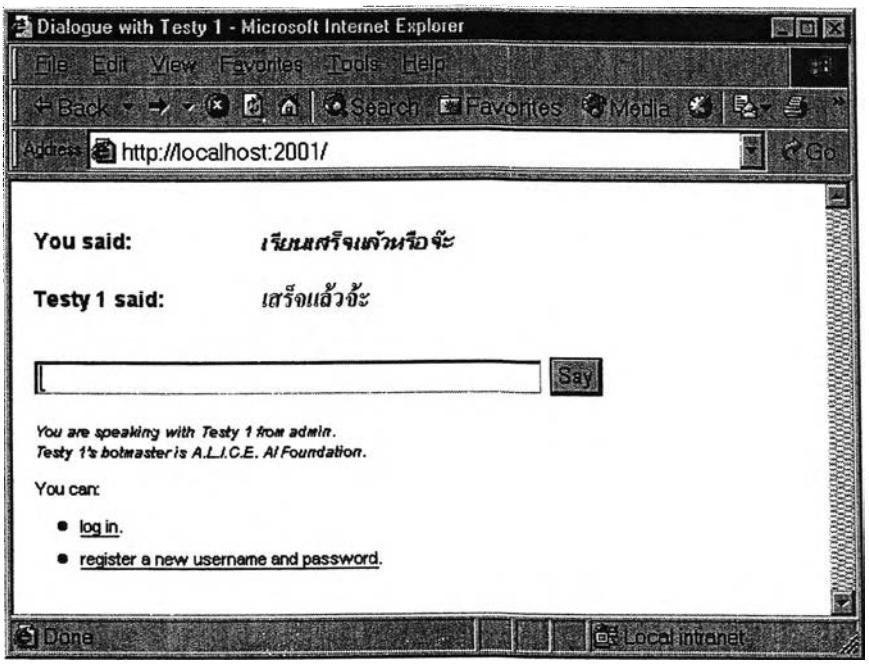
```
[18:02:18] Starting Alicebot Program D version 4.1.5
[18:02:19] Using Java VM 1.5.0_02-b09 from Sun Microsystems Inc.
[18:02:19] On Windows XP version 5.1 (x86)
[18:02:19] Predicates with no values defined will return: "undefined".
[18:02:19] Initializing Multiplexor.
[18:02:20] Loading Graphmaster.
[18:02:20] Starting up with "D:\Work\Thesis\ALICE_Project\conf\startup.xml".
[18:02:21] Configuring bot "TestBot-1".
[18:02:22] Loaded 287 input substitutions.
[18:02:22] Loaded 19 gender substitutions.
[18:02:22] Loaded 9 person substitutions.
[18:02:22] Loaded 60 person2 substitutions.
[18:02:22] Loaded 4 sentence-splitters.
[18:02:23] There is no "sr" element in AIML.
[18:02:23] 6000 categories loaded so far.
[18:02:25] 12000 categories loaded so far.
[18:02:26] 18000 categories loaded so far.
[18:02:26] 1 bots thinking with 23358 categories.
[18:02:26] Alicebot Program D (c) 1995-2002 A.L.I.C.E. AI Foundation
[18:02:26] All Rights Reserved.
[18:02:26] This program is free software; you can redistribute it and/or
[18:02:26] modify it under the terms of the GNU General Public License
[18:02:26] as published by the Free Software Foundation; either version 2
[18:02:26] of the License, or (at your option) any later version.
[18:02:26] Alicebot Program D version 4.1.5 Build [00]
[18:02:26] 23358 categories loaded in 6.54 seconds.
[18:02:26] The AIML Watcher is not active.
[18:02:26] HTTP server listening at http://virtuallife:2001
[18:02:28] Interactive shell: type "/exit" to shut down; "/help" for help.
[18:02:28] Testy 1> My favorite movie is Memento or The Matrix. Have you seen either?

[18:02:28] [Testy 1] user>
```

รูปที่ 5.1 แสดงหน้าจอการทำงานของ A.L.I.C.E Bot

การสนทนากับ A.L.I.C.E Bot ทำได้โดยป้อนข้อความที่ต้องการสนทนาลงไปผ่านเบราว์เซอร์ แต่ไม่ต้องมีการเว้นวรรคระหว่างคำเหมือนการสร้างแคตากอรี เนื่องจากโปรแกรมที่งานวิจัยนี้ได้พัฒนาจะไปทำการตัดคำให้เองก่อนที่จะส่งไปเปรียบกับกับฐานข้อมูล AIML *ถ้าหากพบก็จะเอาคำตอบของบทสนทนานั้นส่งออกไป แต่ถ้าไม่พบก็จะตอบกลับไปว่า "ไม่เข้าใจ"*

ซึ่งการติดต่อกับ A.L.I.C.E Bot เวอร์ชันภาษาไทย (ที่งานวิจัยนี้พัฒนาขึ้น) แสดงได้ดังในรูปที่ 5.2



รูปที่ 5.2 แสดงหน้าจอการติดต่อกับ A.L.I.C.E Bot เวอร์ชันภาษาไทยผ่านเบราว์เซอร์

เมื่อหุ่นยนต์พร้อมที่จะสนทนาผ่านเบราว์เซอร์ ดังในรูปที่ 5.2 เรียบร้อยแล้ว สามารถเริ่มทดสอบประสิทธิภาพได้โดยนำประโยคที่มีคำผิดจากภาคผนวก ข มาลองสนทนากับหุ่นยนต์ หากสามารถแก้ไขจากประโยคอินพุตที่มีคำผิดแล้วตอบออกมาได้เป็นประโยคที่ถูกต้อง ก็จะนับคะแนนผลการทดสอบเป็น 1 สะสมไปเรื่อยๆ จนกว่าจะทำการสนทนาครบ 120 ประโยค แล้วสุดท้ายก็จะคิดประสิทธิภาพออกมาในรูปแบบของเปอร์เซ็นต์

5.4 ผลการทดสอบประสิทธิภาพการสื่อสารกับหุ่นยนต์สนทนา

เริ่มจากการสร้างฐานความรู้เอไอเอ็มแอล เตรียมไว้ก่อนเป็นจำนวน 120 แคตากอรี แล้วนำประโยคที่มีคำผิดจากภาคผนวก ข มาลองสนทนากับหุ่นยนต์ เพื่อทดสอบประสิทธิภาพของอัลกอริทึมแก้ไขคำผิดที่งานวิจัยนี้ได้พัฒนาขึ้นมา

จากการทดสอบประสิทธิภาพในการจับคู่แพทเทิร์นด้วยบทสนทนาที่มีคำผิดแบบไม่ตั้งใจทั้งหมดจำนวน 120 ประโยค พบว่าหุ่นยนต์สามารถตอบได้คิดเป็น 95 เปอร์เซ็นต์จากคำผิดแบบไม่ตั้งใจทั้ง 3 กรณี (ซึ่งในงานวิจัยไม่ได้นำกรณีพิมพ์ตกมาใช้ทดสอบ) แสดงให้เห็นว่าหุ่นยนต์สามารถทำงานได้มากขึ้นจริงหากใช้อัลกอริทึมแก้คำผิดที่ได้จากงานวิจัยนี้

หลังจากทดสอบประสิทธิภาพแล้ว พบว่ายังมีกรณีที่หุ่นยนต์ไม่สามารถตอบได้คิดเป็น 5 เปอร์เซ็นต์ เมื่อทำการวิเคราะห์ความผิดพลาดที่ไม่สามารถจับคู่คำตอบให้หุ่นยนต์ได้นั้นสามารถแบ่งออกเป็น 2 สาเหตุ คือ

1. พจนานุกรมที่ใช้ยังไม่มีคำสมาส

- เช่น โปรแกรมตัดคำตัดเป็น **ทำงาน** แต่ในพจนานุกรมมีแต่คำว่า **ทำ+งาน** จึงมองคำว่า ทำงาน เป็นคำผิด จึงทำการแก้ไขคำนี้ออกมาเป็นคำว่า **ทำงน** (ซึ่งใกล้เคียงที่สุด และมีความหมายในพจนานุกรม) *หรือกล่าวอย่างสรุปได้ว่า โปรแกรมตัดคำมักตัดเป็นคำใหญ่ แต่พจนานุกรมจะประกอบด้วยคำย่อย*

2. ตัวอักษรที่พิมพ์ผิดนั้นไม่ได้อยู่ใกล้เคียงในระยะ 1 ปุ่มรอบตัวอักษรเดิมที่ควรจะเป็น

- เช่น **กรรม** พิมพ์ผิดเป็น **กรรป** ซึ่งตัวอักษรสุดท้ายที่ต้องการนั้นจะต้องพิมพ์ด้วยนิ้วกลางขวา แต่ไปใช้นิ้วกลางซ้ายพิมพ์มาแทน