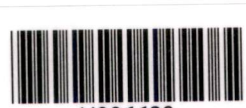
ระบบสร้าง CAPTCHA ภาษาไทย

THAI CAPTCHA GENERATION SYSTEM



โดย

กนกวุธ ฉนัดการ

KANOKWUT THANADKARN

อาจารย์ที่ปรึกษา

รศ.ดร.วรพจน์ กรีสุระเดช

รายงานนี้เป็นส่วนหนึ่งของวิชาการศึกษาอิสระ

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 1 ปีการศึกษา 2553

THAI CAPTCHA GENERATION SYSTEM

KANOKWUT THANADKARN

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS OF THE COURSE

INDEPENDENT STUDY

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1/ 2010

หัวข้อ	ระบบสร้าง CAPTCHA ภาษาไทย
นักศึกษา	นาย กนกวุธ ถนัดการ
รหัสนักศึกษา	51066501
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2553
อาจารย์ที่ปรึกษา	รศ.ดร.วรพจน์ กรีสุระเดช

บทคัดย่อ

โดยทั่วไป CAPTCHA (การทดสอบทัวริงแบบอัตโนมัติเพื่อแยกแยะคอมพิวเตอร์และคน) มักจะใช้ตัวอักษรภาษาอังกฤษเพื่อสร้างชุดตัวอักษรที่ใช้ทดสอบภายใต้สองรูปแบบคือชุดตัวอักษรที่สุ่มแบบเสมอกัน (uniform random) และชุดตัวอักษรที่สุ่มตามความน่าจะเป็นเพื่อให้ดูแล้วคุ้นเคย ซึ่งเป็นที่ยอมรับกันแล้วว่าคนสามารถอ่านชุดตัวอักษรในแบบหลังได้ง่ายกว่า ดังนั้นการสร้างชุดตัวอักษรที่ดูแล้วคุ้นเคยจึงเป็นที่นิยมใช้ในการสร้าง CAPTCHA บนเว็บไซต์ต่างๆ อย่างไรก็ตามยังไม่พบว่ามีการใช้ภาษาไทยในการสร้าง CAPTCHA ในลักษณะดังกล่าวและเป็นเหตุให้การสร้างชุดตัวอักษรที่ดูแล้วคุ้นเคยไม่สามารถให้ผลลัพธ์ที่ดีที่สุดแก่ผู้ใช้ภาษาไทยเป็นภาษาหลักได้เนื่องจากภาษาที่ใช้ในการทดสอบไม่ใช่ภาษาที่คุ้นเคย

รายงานฉบับนี้เสนอผลการศึกษาและพัฒนาระบบสร้าง CAPTCHA ภาษาไทยซึ่งอาศัยไตรแกรมเป็นพื้นฐานในการสุ่มสร้างชุดตัวอักษรตามความน่าจะเป็นเพื่อให้ดูแล้วเป็นที่คุ้นเคย ทั้งนี้ก็เพื่อประโยชน์ในการใช้งานแก่ผู้ที่คุ้นเคยกับภาษาไทยมากกว่าภาษาอื่น

Title	Thai CAPTCHA Generation System
Student	Ms. Kanokwut Thanadkarn
Student ID.	51066501
Degree	Master of Science
Program	Information Technology
Major	Information Science
Academic Year	2010
Advisor	Assoc.Prof. Dr.Worapoj Kreesuradej

ABSTRACT

Typically, CAPTCHA (Completely Automated Public Test to tell Computers and Humans Apart) often use English letters to generate challenge string based on two patterns: uniform random text and familiar text. It is known that humans reading familiar text perform better than on unfamiliar text. Therefore, familiar text generation is popularly used for CAPTCHA on many websites, even though today Thai language have not been found in this kind of test and, thus cannot yield benefit optimal result of familiar text generation for Thai native speakers because testing language is not a familiar language.

This report describes a result from the study and development of Thai CAPTCHA generation system. Tri-gram model is employed to generate familiar text for Thai CAPTCHA. The users who familiar with Thai language more than other languages will gain benefit from this system.

กิตติกรรมประกาศ

รายงานฉบับนี้สำเร็จลุล่วงได้ด้วยความกรุณาจาก รศ.ดร.วรพจน์ กรีสุระเดช ซึ่งเป็นอาจารย์ที่ปรึกษาในการศึกษาอิสระ ผู้เขียนรู้สึกซาบซึ้งในความอนุเคราะห์ของท่านที่ให้คำปรึกษา และสนับสนุนข้อมูลทางวิชาการที่เป็นประโยชน์อย่างยิ่งกับการศึกษาค้นคว้าของข้าพเจ้า จึงขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ ครอบครัวซึ่งเป็นที่รัก ที่คอยห่วงใย เป็นกำลังใจ และสนับสนุนด้านการศึกษาเป็นอย่างดีมาโดยตลอด

ขอขอบพระคุณคณาจารย์ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติหรือเนคเทคที่อนุเคราะห์ให้ใช้คลังข้อความภาษาไทยในการพัฒนา

คุณค่าและประโยชน์อันพึงมาจากการศึกษาฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

กนกวุธ ถนัดการ

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ซอฟต์แวร์โรบ็อทหรือบ็อทหมายถึงโปรแกรมคอมพิวเตอร์ซึ่งสามารถทำงานได้โดยอัตโนมัติด้วยจุดมุ่งหมายที่มักไม่เป็นที่พึงประสงค์ของคนส่วนใหญ่ บ็อทดังกล่าวสร้างปัญหาและความรำคาญให้กับผู้ใช้งานและผู้ให้บริการออนไลน์บนอินเตอร์เน็ตจำนวนมาก แล้วยังมีแนวโน้มว่าบ็อทเหล่านี้จะมีจำนวนเพิ่มมากขึ้นเรื่อยๆ จากการสำรวจในรายงานของไซแมนเทค (Symantec, 2004) พบว่าในช่วงครึ่งแรกของปี 2004 ตรวจพบบ็อทได้เพิ่มขึ้นจากเดิมเฉลี่ยประมาณ 2,000 ครั้งต่อวันเป็นเฉลี่ยประมาณ 30,000 ครั้งต่อวัน

แฮกเกอร์สามารถควบคุมบ็อทผ่านทางเครือข่ายอินเตอร์เน็ต โดยบ็อทที่ถูกติดตั้งไว้บนคอมพิวเตอร์เครื่องใดเครื่องหนึ่งหรือหลายๆ เครื่องอาจถูกสั่งให้ทำงานที่ไม่ถูกต้องอาทิเช่น ส่งอีเมล์ขยะจำนวนมากผ่านบริการอีเมล์ฟรี เขียนโฆษณาหลอกลวงตามกระดานสนทนาบนเว็บ หรือทำการโจมตีแบบกระจายเพื่อหยุดการทำงานของบริการสืบค้นข้อมูลด้วยการส่งคำสั่งสืบค้นจำนวนมากจากหลายๆ เครื่อง

จากปัญหาดังกล่าวทำให้ CAPTCHA หรือการทดสอบแบบอัตโนมัติเพื่อแยกแยะคอมพิวเตอร์และคนได้ถูกนำมาใช้งานด้วยการทดสอบว่าผู้ใช้งานสามาถระบุตัวอักษรที่ใช้ทดสอบได้อย่างถูกต้องหรือไม่ ซึ่งถือเป็นข้อบ่งชี้ว่าเป้าหมายที่ทำการทดสอบคือคนจริงๆ ไม่ใช่บ็อทที่พยายามเลียนแบบคน เพราะตัวอักษรได้ถูกลดความชัดเจนไปจนน่าเชื่อว่าเกินกว่าขีดความสามารถที่กระบวนการรู้จำตัวอักษร (OCR) ด้วยคอมพิวเตอร์จะระบุตัวอักษรได้

อย่างไรก็ตามในบางครั้งแม้แต่คนเองก็ไม่สามารถระบุตัวอักษรในการทดสอบ CAPTCHA ได้อย่างถูกต้อง สาเหตุก็เพราะว่าภาษาที่ใช้ไม่ใช่ภาษาที่คุ้นเคย และตัวอักษรที่ถูกสร้างในการทดสอบด้วยการสุ่มแบบเสมอกัน (Uniform Random) จนเป็นชุดตัวอักษรที่ไร้ระเบียบทำให้ยากต่อการระบุตัวอักษรได้

เพื่อให้คนไทยสามารถระบุตัวอักษรที่ใช้ทดสอบ CAPTCHA ได้ง่ายขึ้น จึงควรใช้ภาษาไทย และควรใช้ชุดตัวอักษรที่สร้างด้วยการสุ่มตามความน่าจะเป็นซึ่งทำให้ดูแล้วคุ้นเคยมากกว่า ในโครงการนี้จึงได้พัฒนาระบบสร้าง CAPTCHA ด้วยชุดตัวอักษรภาษาไทยที่ถูกสุ่มขึ้นโดยอาศัยความน่าจะเป็นในแบบจำลองไตรแกรมที่สกัดขึ้นจากคลังข้อความภาษาไทย

1.2 วัตถุประสงค์

1. เพื่อศึกษาหลักการทำงานของ CAPTCHA
2. เพื่อพัฒนาระบบสร้าง CAPTCHA ภาษาไทยโดยอาศัยแบบจำลองไตรแกรมเป็นพื้นฐานในการสุ่มสร้างชุดตัวอักษร
3. เพื่อให้องค์กรหรือผู้ใช้ที่ต้องการใช้ภาษาไทยใน CAPTCHA สามารถทำได้ง่ายขึ้น

1.3 ขอบเขตของการพัฒนาระบบสร้าง CAPTCHA ภาษาไทย

1. ระบบสร้าง CAPTCHA ภาษาไทยสามารถทำงานร่วมกับคลังข้อความ (corpus) ที่มีรูปแบบตามมาตราฐานทั่วไปได้
2. ระบบสร้าง CAPTCHA ภาษาไทยสามารถสุ่มชุดตัวอักษรตามค่าความน่าจะเป็นทางภาษาด้วยแบบจำลองไตรแกรมที่สกัดจากคลังข้อความได้
3. ชุดตัวอักษรที่สร้างขึ้นด้วยระบบสร้าง CAPTCHA ภาษาไทยต้องสามารถผ่านขั้นตอนตรวจสอบการป้อนอินพุต (ISC) ของระบบปฏิบัติการได้
4. ชุดตัวอักษรที่สร้างขึ้นด้วยระบบสร้าง CAPTCHA ภาษาไทยต้องไม่ตรงกับคำในพจนานุกรม
5. ระบบสร้าง CAPTCHA ภาษาไทยสามารถลดความชัดเจนของภาพชุดตัวอักษรที่สร้างขึ้นได้ด้วยการประมวลผลภาพเพื่อให้การเจาะผ่านทำได้ยากขึ้น
6. ระบบสร้าง CAPTCHA ภาษาไทยสามารถจัดเก็บภาพชุดตัวอักษร CAPTCHA ที่สร้างขึ้นในรูปแบบไฟล์ภาพมาตราฐานได้
7. ระบบสร้าง CAPTCHA ภาษาไทยสามารถให้บริการสร้างภาพชุดตัวอักษร CAPTCHA แบบออนไลน์ผ่านเว็บหรือโปรโตตอล HTTP ได้
8. ระบบสร้าง CAPTCHA ภาษาไทยสามารถทำการประเมินผลลัพธ์ที่สร้างขึ้น ด้วยการเรียกใช้โปรแกรม OCR ภายนอกเพื่อทดลองแปลงภาพชุดตัวอักษร CAPTCHA ที่สร้างขึ้นได้
9. ผู้ใช้สามารถตั้งค่าต่างๆ ของระบบ CAPTCHA ภาษาไทยผ่านทาง GUI บนระบบปฏิบัติการไมโครซอฟต์วินโดว์ และสามารถบันทึกการตั้งค่าเก็บไว้ในไฟล์ได้

1.4 ขั้นตอนและวิธีการพัฒนาระบบ

1. ศึกษาหลักการพื้นฐานในการสร้าง CAPTCHA
2. ศึกษาหลักการทำงานของแบบจำลองไตรแกรมเพื่อใช้ในการสุ่มชุดตัวอักษรตามความน่าจะเป็นทางภาษา

3. ศึกษาวิธีการใช้งานไลบรารี่ WTT สำหรับการตรวจสอบการป้อนอินพุต (ISC)
4. ศึกษาวิธีการใช้งานไลบรารี่ CxImage สำหรับการประมวลผลภาพ
5. ศึกษาวิธีการพัฒนาเว็บเซอร์เวอร์จากหนังสือ Inside Visual C++
6. ศึกษาวิธีการเรียกใช้งานโปรแกรม OCR จากระบบสร้าง CAPTCHA ภาษาไทย
7. ออกแบบและพัฒนาระบบสร้าง CAPTCHA ภาษาไทย
8. ทดสอบและตรวจสอบข้อผิดพลาดต่างๆ ของระบบสร้าง CAPTCHA ภาษาไทย เพื่อทำการปรับปรุงแก้ไขให้สมบูรณ์
9. สรุปผลการทำงานของระบบสร้าง CAPTCHA ภาษาไทย

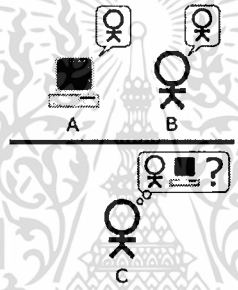
1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเข้าใจหลักการสร้าง CAPTCHA ภาษาไทย
2. สามารถนำหลักการทำงานและวิธีการต่างๆ ที่ได้ศึกษามาประยุกต์ใช้ในการออกแบบพัฒนาระบบสร้าง CAPTCHA ภาษาไทยได้
3. การใช้ภาษาไทยใน CAPTCHA จะช่วยส่งเสริมภาษาไทยซึ่งเป็นเอกลักษณ์ของชาติให้ปรากฏเด่นชัดยิ่งขึ้น
4. การใช้ภาษาไทยใน CAPTCHA ซึ่งมีความซับซ้อนกว่าภาษาที่ใช้กันโดยทั่วไปใน CAPTCHA ได้แก่ ภาษาอังกฤษ จะช่วยเพิ่มประสิทธิภาพในการป้องกันให้ดีขึ้น
5. การใช้ไตรแกรมในการสร้างชุดตัวอักษร CAPTCHA ภาษาไทยจะช่วยลดความยากในการใช้งานได้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 CAPTCHA

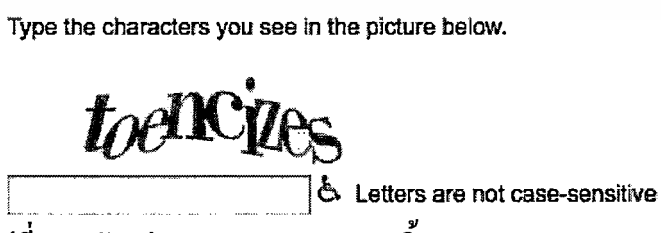
ประวัติโดยสังเขปของ CAPTCHA เริ่มในปี 2493 เมื่อ อลัน ทัวริง ได้เสนอวิธีการในการทดสอบปัญญาประดิษฐ์ (AI) ขึ้น เรียกว่าการทดสอบทัวริง (Turing, 1950) โดยทำการแยกแยะระหว่างเครื่องคอมพิวเตอร์กับคนด้วยการคุยโต้ตอบ ถ้าหากคนเราไม่สามารถแยกระหว่างเครื่องคอมพิวเตอร์กับคนที่คุยด้วยได้ ก็ถือว่าคอมพิวเตอร์เครื่องนั้นเป็นเครื่องที่ชาญฉลาด ซึ่งต่อมาได้มีการประยุกต์แนวคิดนี้ไปใช้ในการป้องกันปัญหาบ็อทหรือสแปม



รูปที่ 2.1 การทดสอบทัวริง

2.1.1 การทดสอบทัวริงแบบอัตโนมัติหรือ CAPTCHA

การทดสอบทัวริงถูกพัฒนาให้สามารถทำการทดสอบได้โดยอัตโนมัติเพื่อแยกแยะคอมพิวเตอร์และคน หรือเรียกอีกอย่างว่า CAPTCHA (**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part) เพื่อใช้คัดกรองคอมพิวเตอร์ที่มีระบบปัญญาประดิษฐ์หรือบ็อทที่ไม่พึงประสงค์ไม่ให้สามารถเข้าไปใช้บริการออนไลน์ต่างๆ อย่างไม่ถูกต้อง



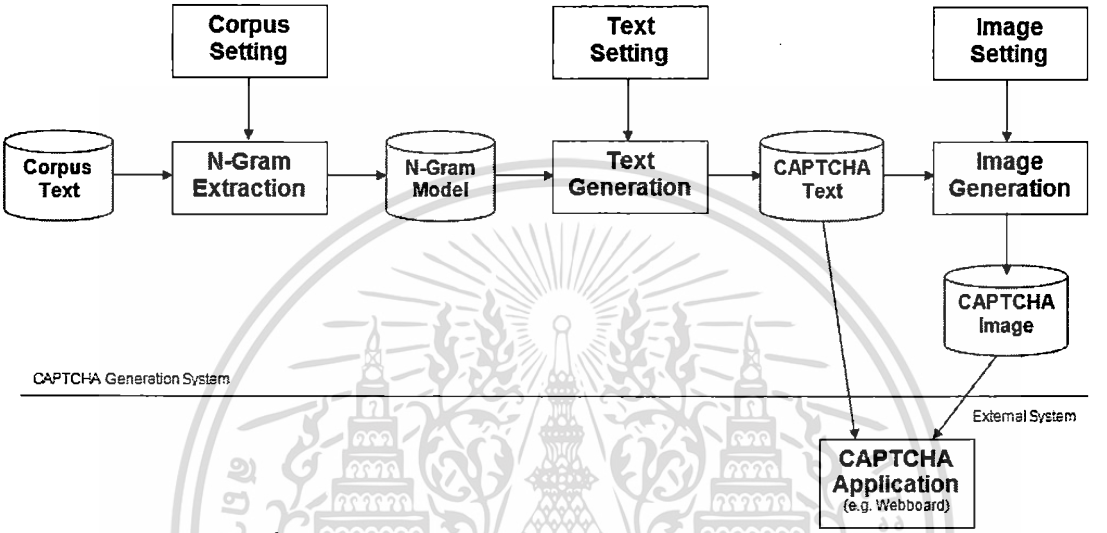
รูปที่ 2.2 ตัวอย่าง CAPTCHA ของกูเกิ้ล (Google CAPTCHA)

จากรูปที่ 2.2 เป็นตัวอย่าง CAPTCHA จากเว็บของกูเกิ้ลซึ่งสร้างโดยการนำชุดตัวอักษรที่สุ่มขึ้นให้ดูคล้ายกับภาษาอังกฤษมาทำการลดความชัดเจนด้วยการบิดงอภาพ ในปัจจุบันมี CAPTCHA ในลักษณะอื่นอีกหลายรูปแบบทั้งที่ใช้ตัวอักษรและไม่ใช้ตัวอักษร

สำหรับในโครงการนี้จะใช้ CAPTCHA ในรูปแบบที่ใช้ตัวอักษรที่สุ่มสร้างด้วยไตรแกรม โดยใช้หลักการเดียวกับ Baffle Text ซึ่งจะทำให้ได้ชุดตัวอักษรที่คล้ายคลึงกับภาษาที่ใช้สกัดสร้างแบบจำลองเอ็นแกรมขึ้นมา (Chew & Baird, 2003)

2.1.2 การสร้าง CAPTCHA

โดยทั่วไปแล้วการสร้าง CAPTCHA ด้วยตัวอักษรที่สุ่มด้วยเอ็นแกรมนั้นมีกระบวนการทำงานเป็นระบบดังรูปที่ 2.3



รูปที่ 2.3 กระบวนการทำงานของระบบสร้าง CAPTCHA

การทำงานหลักของระบบสร้าง CAPTCHA โดยอาศัยเอ็นแกรมในการสุ่มสร้างตัวอักษรนั้นสามารถแบ่งออกได้เป็นสามส่วนคือ

1. การสกัดสร้างแบบจำลองเอ็นแกรม (N-Gram Extraction)
2. การสุ่มสร้างชุดตัวอักษรด้วยเอ็นแกรม (Text Generation)
3. การสร้างและลดความชัดของภาพชุดตัวอักษร (Image Generation)

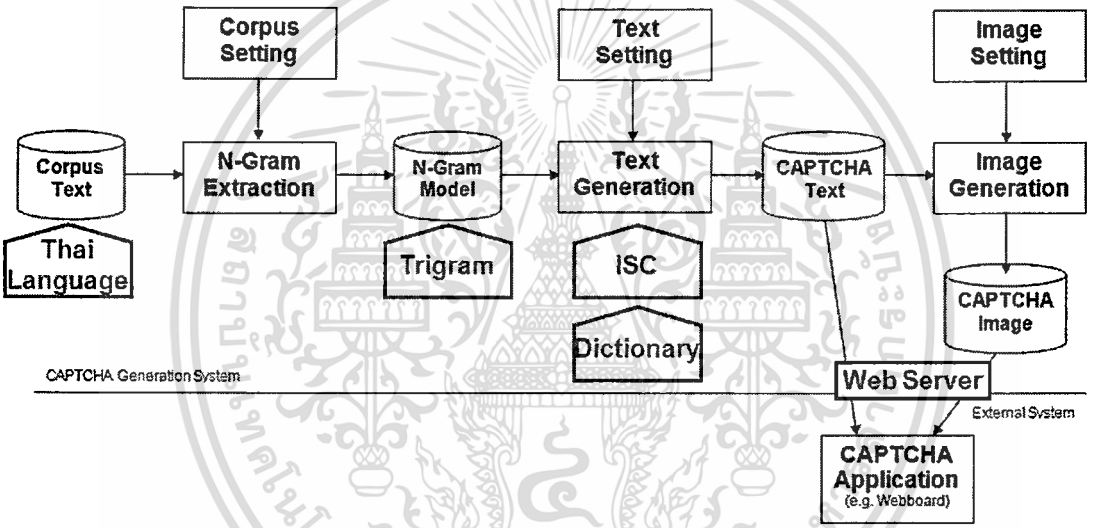
แต่ละส่วนมีกระบวนการทำงานโดยเริ่มจากการนำคลังข้อความ (Corpus Text) ของภาษาต้นแบบ ซึ่งรายละเอียดต่างๆ ของคลังข้อความดังกล่าวเช่นไดเร็กทอรี่ที่ใช้เก็บและรูปแบบไฟล์ จะถูกกำหนดด้วยการตั้งค่าคลังข้อความ (Corpus Setting) โดยคลังข้อความนี้จะถูกนำไปใช้ในส่วนสกัดสร้างเอ็นแกรมเพื่อให้ได้แบบจำลองความน่าจะเป็นทางภาษาเอ็นแกรม (N-Gram Model) ออกมา

จากนั้นจึงนำแบบจำลองที่ได้เข้าสู่ส่วนการสุ่มสร้างชุดตัวอักษรด้วยเอ็นแกรม ซึ่งจะนำค่าที่ตั้งไว้ในการตั้งค่าชุดตัวอักษร (Text Setting) เช่น ความยาวของชุดตัวอักษร มาใช้ในการสุ่มสร้างชุดตัวอักษรตามค่าความน่าจะเป็นด้วยแบบจำลองเอ็นแกรม ทำให้สามารถสุ่มสร้างชุดตัวอักษร (CAPTCHA Text) ที่ดูคล้ายกับคำในภาษาต้นแบบได้ ในขั้นตอนนี้อาจมีการคัดกรองผลลัพธ์ที่ตรงกับคำในพจนานุกรมทิ้งเพื่อช่วยป้องกันการโจมตีด้วยพจนานุกรม (Dictionary Attack)

สุดท้ายจึงนำชุดตัวอักษรที่สร้างขึ้นเข้าสู่การสร้างและลดความชัดของภาพชุดตัวอักษร เพื่อให้ได้ภาพชุดตัวอักษร (CAPTCHA Image) ซึ่งได้ถูกลดความชัดเจนด้วยการประมวลผลภาพ (Image Processing) แบบต่างๆ ตามค่าที่ตั้งไว้ในการตั้งค่าภาพ (Image Setting) เช่น องศาในการหมุนภาพ เป็นต้น เมื่อสร้าง CAPTCHA เสร็จแล้วระบบอื่นที่อยู่นอกเช่น เว็บบอร์ด ก็สามารถนำภาพ CAPTCHA และชุดตัวอักษรเฉลยนั้นไปใช้งานโดยสร้างแบบสอบถามให้ผู้ใช้ป้อนตัวอักษรกลับมาเพื่อพิสูจน์ว่าเป็นคนที่ต้องการใช้ระบบนั้นจริงๆ หรือเป็นบ็อทที่อาจเข้ามารบกวนกันแน่

2.1.3 การสร้าง CAPTCHA ภาษาไทย

ในโครงการนี้ได้นำหลักการสร้าง CAPTCHA ด้วยตัวอักษรที่สุ่มด้วยเอ็นแกรมดังที่กล่าวไว้ในหัวข้อก่อนหน้านี้มาพัฒนาให้สามารถใช้ได้กับภาษาไทย โดยได้มีการปรับปรุงกระบวนการทำงานของระบบดังรูปต่อไปนี้



รูปที่ 2.4 กระบวนการทำงานของระบบสร้าง CAPTCHA ภาษาไทย

จากรูปที่ 2.4 จะเห็นว่าได้มีการพัฒนาเพิ่มเติมกระบวนการย่อยๆ ในส่วนที่เกี่ยวข้องกับภาษาไทย 4 ส่วน โดยมีรายละเอียดดังนี้

1. ใช้คลังข้อความซึ่งเป็นภาษาไทยในการสร้างแบบจำลองเอ็นแกรม โดยได้นำคลังข้อความจากทางศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติหรือเนคเทค ซึ่งได้เผยแพร่ไว้ให้บุคคลทั่วไปสามารถดาวน์โหลดได้บนเว็บไซต์ของโครงการ BEST2010 สำหรับคลังข้อความของเนคเทคนี้ถูกจัดเก็บไว้ในเท็กซ์ไฟล์ด้วยรูปแบบ UTF-8 ทำให้ระบบสร้าง CAPTCHA ภาษาไทยจะต้องสามารถอ่านและแปลงไฟล์ในรูปแบบดังกล่าวเพื่อให้สามารถเข้ากันได้กับรูปแบบของโครงสร้างข้อมูลที่ระบบใช้ในการจัดเก็บเอ็นแกรมได้
2. แบบจำลองเอ็นแกรมที่ระบบสร้าง CAPTCHA ภาษาไทยนำมาประยุกต์ใช้งานคือไตรแกรม โดยยูนิแกรมและไบแกรมจะใช้ในการสร้างตัวอักษรตัวแรกและตัวที่สอง

ส่วนไตรแกรมจะใช้สร้างตัวอักษรตั้งแต่ตัวที่สามเป็นต้นไป ซึ่งต่างจากการสร้าง CAPTCHA ภาษาอังกฤษที่ใช้เฉพาะไบแกรมในการสร้างชุดตัวอักษร (Chew & Baird, 2003) เท่านั้น ทั้งนี้เนื่องจากภาษาไทยประกอบด้วยชุดตัวอักษรเป็นของตัวเองทั้งหมดถึง 67 ตัว ซึ่งมากกว่าจำนวนตัวอักษรในภาษาอังกฤษ ทำให้หากใช้เฉพาะไบแกรมในการสร้างชุดตัวอักษรอาจทำให้ได้ชุดตัวอักษรที่ดูแล้วไม่ค่อยจะคล้ายคลึงกับภาษาไทยอย่างที่ต้องการ

3. ใช้คำศัพท์จากพจนานุกรมภาษาไทยในการคัดกรองผลลัพธ์ เพื่อลดโอกาสในการถูกเจาะผ่านระบบด้วยการการสุ่มไล่คำศัพท์ ในส่วนของรายการคำศัพท์ภาษาไทยที่ใช้ในระบบสร้าง CAPTCHA ภาษาไทยนี้ได้นำมาจากไฟล์ th18057.txt (IBM, 2000)
4. เพิ่มกระบวนการตรวจสอบลำดับการป้อนอินพุต หรือ ISC เพื่อให้มั่นใจได้ว่าผู้ใช้จะสามารถป้อนชุดตัวอักษรที่ระบบสร้าง CAPTCHA ภาษาไทยสร้างขึ้นเข้าทางส่วนอินพุตหรือคีย์บอร์ดของคอมพิวเตอร์ได้ รายละเอียดจะกล่าวถึงในหัวข้อที่ 2.3

นอกจากส่วนต่างๆ ที่ใช้รองรับกับภาษาไทยดังกล่าวแล้ว ในโครงการนี้ได้เพิ่มส่วนที่รองรับกับการให้บริการผ่านเว็บด้วยการพัฒนาเว็บเซิร์ฟเวอร์ที่ประกอบรวมไว้ภายในระบบเพื่อให้ระบบอื่นภายนอกสามารถใช้งานระบบสร้าง CAPTCHA ภาษาไทยในแบบออนไลน์ได้ผ่านทางโปรโตคอล HTTP โดยผู้เขียนได้ศึกษาและนำหลักการพัฒนาเว็บเซิร์ฟเวอร์จากหนังสือ Inside Visual C++ มาประยุกต์ใช้งาน (Kruglinski, 1997)

2.1.4 การเจาะผ่านการทดสอบ CAPTCHA

ปัจจุบันมีความพยายามหาวิธีการที่จะเจาะผ่านการทดสอบ CAPTCHA โดยสามารถแบ่งออกได้เป็นสองแนวทางคือ อาศัยแรงงานคนราคาถูกมาทำการทดสอบแทน และการรู้จำตัวอักษรด้วยคอมพิวเตอร์หรือ OCR

หลักการทำงานของ OCR ที่ใช้ในการเจาะผ่านการทดสอบ CAPTCHA แบบตัวอักษรมักจะประกอบไปด้วยขั้นตอนหลักสามขั้นตอน ดังต่อไปนี้

1. *การประมวลผลก่อนเริ่ม* ในขั้นตอนนี้จะทำการตัดภาพพื้นหลังออกเพื่อกรองให้เหลือเพียงตัวอักษรในลักษณะที่เป็นภาพขาวดำหรือลายเส้นเท่านั้น
2. *การแบ่งตัวอักษร* ขั้นตอนนี้เป็นขั้นตอนที่ยากที่สุดเพราะหากผิดพลาดเพียงเล็กน้อยก็อาจทำให้ได้ผลลัพธ์ที่ผิดพลาดมากจนไม่สามารถแก้ไขในขั้นถัดไปได้ ดังนั้นการทดสอบ CAPTCHA ในปัจจุบันจึงมักใช้เทคนิคต่างๆ เพื่อรบกวนการทำงานในขั้นตอนการแบ่งตัวอักษรนี้อาทิเช่น ทำให้ตัวอักษรบิดงอเป็นรูปงูเลื้อย หรือทำการเชื่อมตัวอักษรหลายตัวด้วยการลากเส้นที่สุ่มขึ้นมา
3. *การรู้จำตัวอักษร* ขั้นตอนสุดท้ายคือทำการวิเคราะห์เปรียบเทียบตัวอักษรแต่ละตัวกับแบบรูปตัวอักษรที่เรียนรู้มาจากฐานข้อมูลตัวอักษรที่ใช้เป็นชุดสอนซึ่งมักได้มาจาก

ตัวอย่างตัวอักษรจากสิ่งพิมพ์จำนวนมากหรือจากไฟล์ฟอนต์หลายๆ ไฟล์ แล้วใช้อัลกอริทึมประมวลผลภาษาธรรมชาติตัดแต่งผลลัพธ์ให้มีความถูกต้องมากยิ่งขึ้น

สำหรับในโครงการนี้ได้พัฒนาให้สามารถทำงานร่วมกับโปรแกรม OCR ที่ระบบรองรับ (ABBY FineReader) เพื่อให้ผู้ใช้สามารถทำการประเมินประสิทธิภาพในการป้องกันบ็อทเบื้องต้นได้ ด้วยการทดลองใช้โปรแกรม OCR แปลงภาพ CAPTCHA ที่สร้างขึ้นว่าสามารถป้องกันการเจาะผ่านด้วย OCR ได้มากน้อยเพียงใด

2.2 แบบจำลองเอ็นแกรม (N-Gram)

ชุดตัวอักษรที่ใช้ในการสร้าง CAPTCHA มีทั้งแบบที่เกิดจากการสุ่มโดยเสมอกัน (uniform) และแบบที่สุ่มตามค่าความน่าจะเป็นทางภาษาเพื่อให้ได้ผลลัพธ์ที่ดูแล้วคุ้นเคยกว่า ซึ่งทำให้ผู้ใช้งานสามารถระบุชุดตัวอักษรได้แม่นยำและรวดเร็วขึ้น ดังนั้นจึงเกิดการประยุกต์เอาแบบจำลองเอ็นแกรม (N-Gram) มาใช้ในการสร้างชุดตัวอักษร CAPTCHA ขึ้น

เอ็นแกรม (อัครพล, 2548) คือแบบจำลองที่ใช้คำนวณค่าความน่าจะเป็นของชุดตัวอักษรที่ประกอบเรียงกัน (Character Sequence) เป็นคำ หรือค่าความน่าจะเป็นของคำที่ประกอบเรียงกัน (Word Sequence) เป็นประโยค ซึ่งค่าความน่าจะเป็นนั้นสามารถประมาณได้ด้วยการนับและการคำนวณข้อมูลทางสถิติจากคลังข้อความ แบบจำลองเอ็นแกรมใช้หลักการทางสถิติในหลาย ๆ ด้านมาประยุกต์ใช้ โดยมีรายละเอียดดังต่อไปนี้

2.2.1 แกรม (Gram)

คือ หน่วยที่ใช้ในการสร้างแบบจำลอง อาจจะเป็นเสียง คำ หรือ ตัวอักษรก็ได้และแกรม (Gram) มีได้หลายขนาดแล้วแต่จะกำหนด ตั้งแต่ 1 จนถึง N ในแบบจำลองเอ็นแกรมนี้ใช้ความยาวของชุดตัวอักษรและคำที่เขียนเรียงกัน ได้แก่ 2-Gram , 3-Gram , 4-Gram ฯลฯ ถ้าจะประมาณค่าความน่าจะเป็นของชุดคำหรือชุดตัวอักษรจากคลังข้อความโดยการใช้วิธีเอ็นแกรมผลที่ได้มีดังนี้

การประมาณค่าด้วย 1-Gram (ยูนิแกรม) คือ การประมาณค่าความน่าจะเป็นของชุดตัวอักษรที่เกิดขึ้นร่วมกันว่ามีค่าเท่ากับผลคูณของความน่าจะเป็นที่จะพบตัวอักษรแต่ละตัวในชุดตัวอักษรนั้น

การประมาณค่าด้วย 2-Gram (ไบแกรม) คือ การประมาณค่าความน่าจะเป็นของชุดตัวอักษรที่เกิดขึ้นร่วมกันว่ามีค่าเท่ากับผลคูณของความน่าจะเป็นที่จะพบตัวอักษร(คำ) ทีละ 2 ตัว (คำ) ติดกันในชุดตัวอักษรนั้น

การประมาณค่าด้วย 3-Gram (ไตรแกรม) คือ การประมาณค่าความน่าจะเป็นของชุดตัวอักษรที่เกิดขึ้นร่วมกันว่ามีค่าเท่ากับผลคูณของความน่าจะเป็นที่จะพบตัวอักษร(คำ) ทีละ 3 ตัว (คำ) ติดกันในชุดตัวอักษรนั้น

การประมาณค่าด้วย 4-Gram (ควอดิแกรม) คือ การประมาณค่าความน่าจะเป็นของชุดตัวอักษรที่เกิดขึ้นร่วมกันว่ามีค่าเท่ากับผลคูณของความน่าจะเป็นที่จะพบตัวอักษร (คำ) ทีละ 4 ตัว (คำ) ติดกันในชุดตัวอักษรนั้น

หรืออาจประมาณค่าความน่าจะเป็นจากความยาวของเอ็นแกรมมากกว่า 4-แกรม ก็ได้ ขึ้นอยู่กับความจำเป็น แต่ระบบของเอ็นแกรมก็ยิ่งซับซ้อนมากขึ้นตามลำดับ

2.2.2 หลักการทำงานของเอ็นแกรม

การประมาณค่าความน่าจะเป็นของชุดอักษร โดยการใช้เอ็นแกรมดังที่กล่าวมา คือ การใช้สมมติฐานของมาร์คอฟ (Markov Assumption) ว่า การปรากฏของตัวอักษรตัวหนึ่งขึ้นกับตัวอักษรก่อนหน้าเพียง n-1 ตัว โดยสามารถประมาณค่าความน่าจะเป็นด้วยสมการดังนี้

ยูนิแกรม $P(c_1 \dots c_i) = P(c_1)P(c_1) \dots P(c_i)$ (2.1)

ไบแกรม $P(c_1 \dots c_i) = P(c_1)P(c_2|c_1) \dots P(c_i|c_{i-1})$ (2.2)

ไตรแกรม $P(c_1 \dots c_i) = P(c_1)P(c_2|c_1)P(c_3|c_1c_2) \dots P(c_i|c_{i-2}c_{i-1})$ (2.3)

เอ็นแกรม $P(c_1 \dots c_i) = P(c_1)P(c_2|c_1)P(c_3|c_1c_2) \dots P(c_i|c_{i-n-1} \dots c_{i-1})$ (2.4)

โดยที่

P คือค่าความน่าจะเป็น

c_x คือตัวอักษรตัวที่ x

$(c_x \dots c_y)$ คือชุดตัวอักษรตั้งแต่ตัวที่ x ถึงตัวที่ y

ส่วนความน่าจะเป็นของชุดคำที่รวมกันเป็นประโยค $w_1 \dots w_i$ หากประมาณค่าด้วย เอ็นแกรมต่าง ๆ โดยที่

P คือค่าความน่าจะเป็น

w_x คือคำลำดับที่ x

$(w_x \dots w_y)$ คือชุดคำที่ประกอบเป็นประโยคตั้งแต่คำลำดับที่ x ถึงตัวที่ y

จะทำให้ได้สมการความน่าจะเป็นเอ็นแกรมของประโยคดังนี้

$$P(w_1 \dots w_i) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \dots P(w_i|w_{i-n-1} \dots w_{i-1}) \quad (2.5)$$

ความน่าจะเป็นของประโยค $P(w_1 \dots w_i)$ สามารถประมาณได้ โดยถือว่าการปรากฏของคำ w_i นั้นขึ้นอยู่กับจำนวนคำข้างหน้า $n-1$ ตัวเท่านั้นหรือขึ้นอยู่กับขนาดของเอ็นแกรม ดังนั้นถ้าหากประมาณค่าความน่าจะเป็นของประโยคนี้ โดยใช้ไบแกรมจะปรับเปลี่ยนสมการดังนี้

$$P(w_1 \dots w_i) = P(w_1| < s >)P(w_2|w_1) \dots P(w_i|w_{i-1}) \quad (2.6)$$

$P(w_1 | < s >)$ คือความน่าจะเป็นของคำที่หนึ่งเมื่อเกิดเป็นคำแรกของประโยค ซึ่ง ในที่นี้คือช่องว่าง

$P(w_2 | w_1)$ คือความน่าจะเป็นของคำ w_2 หลังจากเกิดคำ w_1

$P(w_i | w_{i-1})$ คือความน่าจะเป็นของคำ w_i หลังจากเกิดคำ w_{i-1}

จากสมการประมาณค่าความน่าจะเป็นด้วยไบแกรม (2.6) หากต้องการหาความน่าจะเป็นของประโยค เช่น “He like to eat banana” โดยใช้ไบแกรมของคำ จะได้ผลออกมาคือ

$$P(He| < s >)P(like|He)P(to|like)P(eat|to)P(banana|eat) \quad (2.7)$$

ส่วนค่าความน่าจะเป็นสามารถหาได้จากคลังข้อความโดยใช้สมการดังนี้

$$P(w_i | w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})} \quad (2.8)$$

โดยที่ c คือ จำนวนนับ ตัวอย่างเช่น $P(like|He)$ หมายถึงค่าความน่าจะเป็นของ like เมื่อเกิดร่วมกับคำว่า He คำนวณได้จากนำจำนวนนับของ He ที่เกิดร่วมกับคำว่า like หารด้วยจำนวนนับของการเกิด He เดี่ยว ๆ วิธีการใช้แบบจำลองเอ็นแกรมนี้เป็นวิธีทางสถิติที่นิยมใช้กันมากที่สุด เพราะเป็นวิธีที่เรียบง่าย มีประสิทธิภาพสูง

เมื่อมีประโยค $W = (w_1 \dots w_i)$ มาให้ แบบจำลองเอ็นแกรมจะคำนวณค่าความน่าจะเป็นของการเกิดคำใดๆ โดยพิจารณาจาก $n-1$ คำก่อนหน้า อาทิเช่น $n = 2$ ซึ่งเรียกว่าไบแกรม จะให้ค่าความน่าจะเป็นของ คำใดๆ โดยดูจาก คำก่อนหน้ามันเพียงคำเดียว ตัวอย่างเช่น

P(ไป | จะ) = 0.8	P(จะ | ไป) = 0.01	P(จะ | ผม) = 0.7	(2.9)
P(ตลาด | ไป) = 0.5	P(โรงเรียน | ไป) = 0.6	P(ผม | ไป) = 0.02	

ถ้ามีประโยคยาวๆ W = (w1 ... wM) มาหนึ่งประโยค เช่น "ผม จะ ไป โรงเรียน" ค่าความน่าจะเป็นที่จะเกิดประโยคดังกล่าวก็สามารถ คำนวณได้โดยการคูณต่อๆ กันไปดังนี้

P(ผม จะ ไป โรงเรียน) = P(จะ | ผม) P(ไป | จะ) P(โรงเรียน | ไป) (2.10)

= (0.7) (0.8) (0.6)

= 0.336

P(จะ ไป ผม จะ) = P(ไป | จะ) P(ผม | ไป) P(จะ | ผม) (2.11)

= (0.8) (0.02) (0.7)

= 0.0112

จะเห็นได้ว่า "จะ ไป ผม จะ" มีโอกาสเกิดต่ำมากเมื่อเทียบกับ "ผม จะ ไป โรงเรียน" ในทำนองเดียวกัน เราสามารถสร้างแบบจำลองไตรแกรมโดยที่กำหนดค่าความน่าจะเป็นของแต่ละคำโดยพิจารณาจากคำก่อนหน้า 2 คำ อาทิเช่น

$$P(\text{ผม จะ ไป โรงเรียน}) = P(\text{จะ} \mid \text{ผม})\, P(\text{ไป} \mid \text{ผม จะ})\, P(\text{โรงเรียน} \mid \text{จะ ไป}) \quad (2.12)$$

เพราะฉะนั้นเพื่อสร้างแบบจำลองเอ็นแกรมสิ่งที่จะต้องคำนวณเตรียมเอาไว้ก็คือค่าความน่าจะเป็นซึ่งก็คือ P(w2|w1) สำหรับไบแกรม หรือ P(w3|w1,w2) สำหรับไตรแกรมนั้นเอง

2.2.3 การสุ่มสร้างชุดตัวอักษรด้วยเอ็นแกรม

ในบทความของแชนนอล (Shannon, 1984) ได้อธิบายการสุ่มสร้างชุดตัวอักษรโดยใช้โครงสร้างทางสถิติของข้อความต่างๆ ซึ่งก็คือเอ็นแกรม เพื่อให้ได้ข้อความที่มีโครงสร้างทางสถิติที่เหมือนกัน ดังนั้นหากเราสกัดโครงสร้างทางสถิติของภาษาที่เราต้องการเอาไว้ในแบบจำลองเอ็นแกรมได้ เราก็สามารถจะสุ่มสร้างชุดตัวอักษรที่มีความคล้ายคลึงกับภาษาในแบบจำลองได้เช่นกัน

การสุ่มตัวอักษรหรือการสุ่มตัวเลขตามความน่าจะเป็นนั้นสามารถทำได้หลายวิธี หนึ่งในนั้นคือการใช้วิธีแปลงผกผัน (Inversion Method) สมมุติว่า X คือตัวแปรสุ่มชนิดต่อเนื่อง ซึ่งมี F(X) เป็นฟังชันการแจกแจงความน่าจะเป็นสะสม CDF (Cumulative Distribution Function) ของตัวแปรดังกล่าว นั้นคือ F(x) คือผลรวมของค่าความน่าจะเป็นของตัวแปร X ที่มีค่าน้อยกว่าหรือเท่ากับ x ทำให้สามารถสร้างตัวเลขสุ่มแบบไม่เสมอกันด้วยวิธีผกผันได้จากสมการดังนี้ (Devroye, 1986)

$$x = F^{-1}(u) \quad (2.13)$$

โดยที่ u คือตัวแปรสุ่มแจกแจงสม่ำเสมอแบบต่อเนื่องในช่วง (0, 1) ผลลัพธ์ที่ได้ของสมการก็คือตัวแปรสุ่ม x ที่มีค่าตัวเลขแจกแจงได้ตามฟังชันการแจกแจงความน่าจะเป็น PDF (Probability Density Function) หรือ $f(x)$ ของมัน ซึ่งสมการดังกล่าวมีฟังชันผกผันของฟังชันการแจกแจงความน่าจะเป็นสะสม CDF หรือ $F(x)$ ที่สัมพันธ์กัน

ในกรณีที่ตัวแปรสุ่ม x เป็นแบบไม่ต่อเนื่อง (Discrete Random Variable) คือมีค่ากระจายตัวตามรูปแบบ $\{x_1 < x_2 < \ldots < x_k\}$ ด้วยค่าความน่าจะเป็น $\{p_1, p_2, \ldots p_k\}$ ซึ่ง p_i แต่ละค่าจะมีค่ามากกว่าหรือเท่ากับ 0 และผลรวมของค่าความน่าจะเป็นทุกค่าตั้งแต่ p_1 ถึง p_k จะเท่ากับ 1

ภายใต้ข้อกำหนดของตัวแปรสุ่มแบบไม่ต่อเนื่องดังกล่าวเราสามารถนิยามฟังชันการแจกแจงความน่าจะเป็นสะสม $F(x)$ ที่สัมพันธ์กันได้จากสมการดังนี้

$$F(\mathrm{x}) = \begin{cases} 0 & , \text{if } x < x_1 \\ \sum_{i=1}^{j} p_i & , \text{if } x_j \le x < x_{j+1}, j = 1, \ldots, k-1 \\ 1 & , \text{if } x \ge x_k \end{cases} \quad (2.14)$$

และสามารถนิยามฟังชันผกผันเทียมที่สัมพันธ์กันได้จากสมการดังนี้

$$F^{-1}(u) = \begin{cases} x_1, & \text{if } u \leq p_1 \\ x_j, & \text{if } \sum_{i=1}^{j-1} p_i < u \leq \sum_{i=1}^{j} p_i \,, j = 2, \ldots, k \end{cases} \quad (2.15)$$

2.3 การตรวจสอบลำดับการป้อนอินพุต (ISC)

เพื่อให้ชุดตัวอักษรที่ถูกสุ่มสร้างขึ้นมาเรียงตัวถูกต้องและสามารถป้อนกลับเข้าทางส่วนอินพุตของระบบปฏิบัติการโดยไม่เกิดข้อผิดพลาดขึ้นได้ จะต้องอาศัยการตรวจสอบลำดับการป้อนอินพุตหรือ ISC (Input Sequence Check) สำหรับในระบบสร้าง CAPTCHA ภาษาไทยนี้ชุดตัวอักษรจะถูกสุ่มสร้างขึ้นด้วยภาษาไทย จึงได้ประยุกต์ใช้กระบวนการตรวจสอบลำดับการป้อนอินพุตตามมาตรฐาน วทท. ซึ่งในปัจจุบันมีการนำมาตรฐานนี้ไปใช้งานในระบบปฏิบัติการหลายระบบ ตั้งแต่ระบบไมโครซอฟต์ DOS 6.0 ไปจนถึงวินโดว์ทุกเวอร์ชัน (Koanantakool et al, 2009)

มารตราฐาน วทท. ได้กำหนดให้แบ่งประเภทของตัวอักษรออกเป็นหกประเภทคือ ตัวควบคุม (CTRL), พยัญชนะ (CONS), สระ (V), วรรณยุกต์ (TONE), ตัวกำกับเสียง (D), ตัวอักษรที่ใช้ประสมคำไม่ได้ (NON) ซึ่งทั้งตัวอักษรทั้งหกประเภทสามารถแบ่งออกเป็นประเภทย่อยๆ รวมแล้วได้ทั้งหมด 17 ประเภทย่อยดังตารางที่ 2.2

ตารางที่ 2.1 การแบ่งประเภทตัวอักษรไทยตามมาตรฐาน วทท.

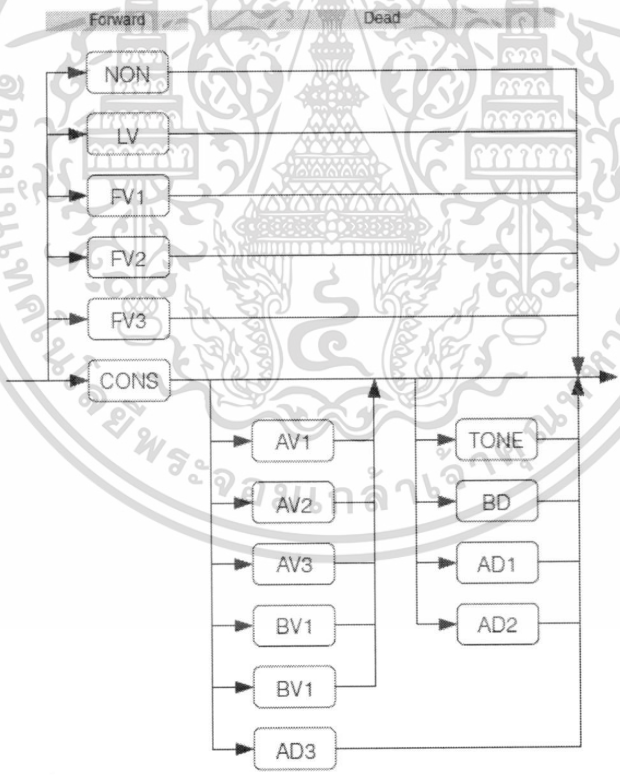
1. CTRL	ตัวอักษรที่ใช้ควบคุมตามมาตรฐานแอสกี
2. NON	ตัวอักษรที่ใช้ประสมคำไม่ได้ (ตัวอักษรภาษาอังกฤษทั้งหมด และตัวเครื่องหมายวรรคตอนได้แก่ ไปยาลน้อย (ฯ), เครื่องหมายเงินบาท (฿), ไม้ยมก (ๆ), โคมูตร (ฺ), ฟองมัน (๎), อังคั่นคู่ (ฺ), และตัวเลขไทย (๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙)
3. CONS	พยัญชนะไทยซึ่งมีทั้งหมด 44 ตัว
4. LV	สระนำหน้าซึ่งมี 5 ตัวได้แก่ เ แ โ ใ ไ
5. FV1	สระตามชนิดที่ 1 ได้แก่ อะ อา อำ
6. FV2	สระตามชนิดที่ 2 ได้แก่ ลากข้างยาว (ๅ)
7. FV3	สระตามชนิดที่ 2 ได้แก่ ฤ ฦ
8. BV1	สระใต้ชนิดที่ 1 ได้แก่ สระอุ (อุ)
9. BV2	สระใต้ชนิดที่ 2 ได้แก่ สระอู (อู)
10. BD	ตัวกำกับเสียงใต้ได้แก่ พินทุ (ฺ)
11. TONE	วรรณยุกต์ 4 ตัวได้แก่ อ่ อ้ อ๊ อ๋
12. AD1	ตัวกำกับเสียงบนชนิดที่ 1 ได้แก่ นิคหิต (อำ), ทัณฑฆาต (อ์)
13. AD2	ตัวกำกับเสียงบนชนิดที่ 2 ได้แก่ ไม้ไต่คู้ (อ็)

ตารางที่ 2.1 (ต่อ)

14. AD3	ตัวกำกับเสียงบนชนิดที่ 3 ได้แก่ ยามักการ (อ์)
15. AV1	สระบนชนิดที่ 1 ได้แก่ สระอิ (อิ)
16. AV2	สระบนชนิดที่ 2 ได้แก่ ไม้หันอากาศ (อั), สระอึ (อึ)
17. AV3	สระบนชนิดที่ 3 ได้แก่ สระอี (อี), สระอื (อื)

การแบ่งประเภทตัวอักษรไทยตามตารางที่ 2.2 ถูกนำไปกำหนดการป้อนอินพุตในแต่ละเซลล์ของภาษาไทย ซึ่งเซลล์หนึ่งๆ ในภาษาไทยอาจประกอบด้วย หนึ่งตัวอักษร (ในระดับกลาง) หรือ สองตัวอักษร (หนึ่งตัวในระดับกลาง และอีกหนึ่งตัวในระดับบนหรือล่าง) หรือ สามตัวอักษร (หนึ่งตัวในระดับกลาง และอีกสองตัวในระดับบนและล่าง หรืออีกสองตัวในระดับบนทั้งคู่)

ข้อกำหนดตามมาตราฐาน วทท. มีสองแบบคือ แบบขั้นพื้นฐาน เช่น “เกาะ” จะต้องประสมด้วย เ-ก-า-ะ ตามลำดับหรือ “ที่” จะต้องประสมด้วย ท - (สระอี) - (ไม้เอก) ไม่ใช่ ท - (ไม้เอก) - (สระอี) ดังที่แสดงไว้ในรูปที่ 2.5



รูปที่ 2.5 การผสมเซลล์ภาษาไทยตามมาตราฐาน วทท.

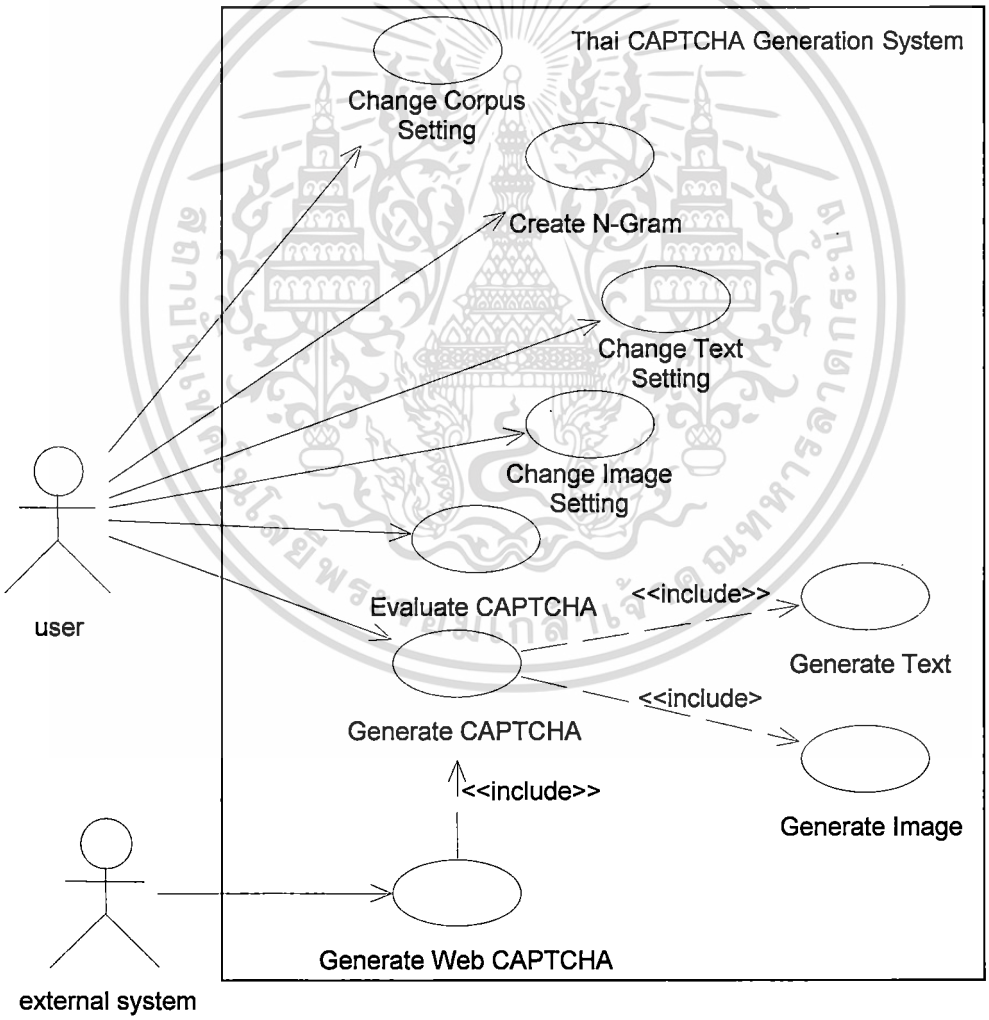
อีกแบบคือการตรวจสอบที่เคร่งครัดขึ้น โดยใช้เงื่อนไขทางตรรกของภาษาไทยซึ่งมีที่มาจากการวิเคราะห์การประสมเซล์ที่เป็นไปได้ทั้งหมดซึ่งช่วยขจัดลำดับการเรียงตัวของตัวอักษรในเซลล์ที่อาจทำให้เกิดปัญหาได้มากยิ่งขึ้น

บทที่ 3
การวิเคราะห์และออกแบบระบบ

ในการวิเคราะห์และออกแบบการทำงานของระบบสร้าง CAPTCHA ภาษาไทยนั้น จะใช้ยูเอ็มแอล (UML) เป็นหลัก โดยมีรายละเอียดดังต่อไปนี้

3.1 การจำลองการทำงานด้วยยูสเคสไดอะแกรม

ระบบสร้าง CAPTCHA ภาษาไทยนั้น มีผู้กระทำ (Actor) ที่ใช้งานระบบอยู่สองส่วน และมียูสเคสหลักทั้งหมดเจ็ดกรณี รวมถึงยูสเคสย่อยอีกสองกรณีดังต่อไปนี้



รูปที่ 3.1 ยูสเคสไดอะแกรมของระบบสร้าง CAPTHA ภาษาไทย

จากรูปที่ 3.1 แสดงให้เห็นยูสเคสของระบบสร้าง CAPTHA ภาษาไทยซึ่งประกอบไปด้วยการใช้งานระบบโดยผู้ใช้ (User) ทั้งหมดหกกรณีคือ

1. Change Corpus Setting (ปรับเปลี่ยนการตั้งค่าคลังข้อความ)
2. Create N-Gram (สร้างเอ็นแกรม)
3. Change Text Setting (ปรับเปลี่ยนการตั้งค่าชุดตัวอักษร CAPTCHA)
4. Change Image Setting (ปรับเปลี่ยนการตั้งค่าชุดภาพ CAPTCHA)
5. Generate CAPTCHA (สร้าง CAPTCHA)
6. Evaluate CAPTCHA (ประเมิน CAPTCHA)

รวมถึงการใช้งานโดยระบบอื่นภายนอกอีกหนึ่งกรณีคือ Generate Web CAPTCHA (สร้าง CAPTCHA ผ่านเว็บ) ในกรณีของยูสเคส Generate CAPTCHA ยังได้รวมยูสเคสย่อยไว้สองยูสเคส คือ Generate Text และ Generate Image ซึ่งรายละเอียดของแต่ละยูสเคสจะถูกอธิบายไว้ในตารางที่ 3.1 ถึง 3.9 ดังนี้

ตารางที่ 3.1 คำอธิบายยูสเคสไดอะแกรมของ Change Corpus Setting

Use case Name :	Change Corpus Setting	
Brief Description :	ปรับเปลี่ยนไดเร็กทอรี่และรูปแบบของไฟล์ที่ใช้เก็บคลังข้อความ	
Actors :	User	
Precondition :	รูปแบบไฟล์ที่ใช้เก็บคลังข้อความต้องอยู่ในรูปแบบ UTF-8 หรือเท็กไฟล์ทั่วไป (Windows-874)	
Postcondition :	ไดเร็กทอรี่และรูปแบบของคลังข้อความถูกบันทึกและจดจำไว้ในระบบ	
Flow of Activities :	Actor	System
	1. ผู้ใช้เปลี่ยนไดเร็กทอรี่ของคลังข้อความที่ต้องการ 2. ผู้ใช้ปรับรูปแบบของคลังข้อความว่าเป็นแบบ UTF-8 หรือแบบธรรมดา 3. ผู้ใช้สั่งบันทึกข้อมูล	1.1 ระบบแสดงไฟล์ที่มีอยู่ในไดเร็กทอรี่ 3.1 ระบบบันทึกไดเร็กทอรี่และรูปแบบของคลังข้อความไว้ในไฟล์

ตารางที่ 3.2 คำอธิบายยูสเคสไดอะแกรมของ Create N-Gram

Use case Name :	Create N-Gram
Brief Description :	สร้างแบบจำลองทางภาษาเอ็นแกรมเพื่อใช้ในการสุ่มสร้างชุดตัวอักษร

ตารางที่ 3.2 (ต่อ)

Actors :	user	
Precondition :	จะต้องมีไฟล์คลังข้อความในรูปแบบและไดเร็กทอรี่ที่ตั้งค่าไว้	
Postcondition :	แบบจำลองทางภาษาเอ็นแกรมถูกสร้างขึ้นและพร้อมนำไปใช้ในการสุ่มสร้างชุดตัวอักษร	
Flow of Activities :	Actor	System
	1. ผู้ใช้สั่งให้สร้างเอ็นแกรม	1.1 ระบบค้นหาไฟล์คลังข้อความทั้งหมดที่มีอยู่ในไดเร็กทอรี่ 1.2 ระบบเปิดไฟล์และนับความถี่เอ็นแกรมของตัวอักษรทีละไฟล์ 1.3 คำนวณค่าความน่าจะเป็นเก็บไว้ในหน่วยความจำ
Exceptional Condition :	1.1 ถ้าระบบไม่พบไฟล์คลังข้อความในไดเร็กทอรี่ ระบบจะแจ้งเตือนผู้ใช้ทราบและถือว่ายังไม่ได้ทำการสร้างเอ็นแกรม	

ตารางที่ 3.3 คำอธิบายยูสเคสไดอะแกรมของ Change Text Setting

Use case Name :	Change Text Setting	
Brief Description :	ตั้งค่าต่างๆ ที่ใช้ในการสุ่มสร้างตัวอักษร CAPTCHA	
Actors :	user	
Precondition :	-	
Postcondition :	ค่าต่างๆ ที่ใช้ในการสุ่มสร้างตัวอักษร CAPTCHA ถูกบันทึกและจดจำไว้ในระบบ	
Flow of Activities :	Actor	System
	1. ผู้ใช้ปรับขนาดความยาวของชุดตัวอักษร 2. ผู้ใช้ปรับการตรวจสอบชุดตัวอักษรกับคำในพจนานุกรมว่าทำหรือไม่	

ตารางที่ 3.3 (ต่อ)

	3. ผู้ใช้ปรับการตรวจสอบการป้อนอินพุตกับชุดตัวอักษรว่าทำหรือไม่ 4. ผู้ใช้ปรับการตรวจสอบตัวอักษรตัวแรกว่าทำหรือไม่ 5. ผู้ใช้ปรับการตรวจสอบตัวอักษรตัวสุดท้ายว่าทำหรือไม่ 6. ผู้ใช้สั่งให้สร้างชุดตัวอักษรให้ดูเป็นตัวอย่าง 7. ผู้ใช้สั่งบันทึกข้อมูล	6.1 ระบบสร้างชุดตัวอักษรตามค่าต่างๆ ที่ผู้ใช้เลือกและแสดงให้ผู้ใช้ดูเป็นตัวอย่าง 7.1 ระบบบันทึกค่าต่างๆ ที่ใช้ในการสร้างชุดตัวอักษรไว้ในไฟล์
Exceptional Condition :	6.1. ถ้ายังไม่ได้สร้างเอ็นแกรมระบบจะแจ้งเตือนผู้ใช้ว่า “ไม่สามารถสร้างตัวอย่างแสดงให้ดูได้เนื่องจากยังไม่ได้สร้างเอ็นแกรม”	

ตารางที่ 3.4 คำอธิบายยูสเคสไดอะแกรมของ Change Image Setting

Use case Name :	Change Image Setting	
Brief Description :	ปรับค่าต่างๆ ที่ใช้ในการลดความชัดของภาพชุดตัวอักษร CAPTCHA	
Actors :	User	
Precondition :	-	
Postcondition :	ค่าต่างๆ ที่ใช้ในการลดความชัดของภาพชุดตัวอักษร CAPTCHA ถูกบันทึกและจดจำไว้ในระบบ	
Flow of Activities :	Actor	System
	1. ผู้ใช้ปรับขนาดความกว้างและความสูงของภาพ 2. ผู้ใช้ปรับระดับสัญญาณรบกวน (noise) ในระหว่าง 0-255	1.1 ระบบสร้างภาพชุดตัวอักษรเมื่อค่าต่างๆ มีการเปลี่ยนแปลงเพื่อแสดงให้ผู้ใช้ดูเป็นตัวอย่าง

ตารางที่ 3.4 (ต่อ)

	3. ผู้ใช้ปรับระดับการสั่นของภาพ (jitter) ในระหว่าง 0-10 4. ผู้ใช้ปรับจำนวนรอบในการทำให้ภาพมัวลง (blur) ในระหว่าง 0-10 รอบ 5. ผู้ใช้ปรับระดับการบิดภาพ (skew) ในระหว่าง 0-10 6. ผู้ใช้เลือกองศาที่ต้องการหมุนภาพ (rotate) ในระหว่าง 0-45 องศา 7. ผู้ใช้เลือกความหนาของการเน้นขอบ (edge) ในระหว่าง 0-10 8. ผู้ใช้สั่งบันทึกข้อมูล	8.1 ระบบบันทึกค่าต่างๆ ที่ใช้ในการลดความชัดเจนของภาพชุดตัวอักษรไว้ในไฟล์

ตารางที่ 3.5 คำอธิบายยูสเคสไดอะแกรมของ Generate CAPTCHA

Use case Name :	Generate CAPTCHA
Brief Description :	สร้างภาพชุดตัวอักษร CAPTCHA
Actors :	User
Precondition :	จะต้องทำการสร้างเอ็นแกรมเสร็จสิ้นแล้ว, และผู้ใช้ต้องทำการตั้งค่าต่างๆ ที่ใช้ในการสร้างชุดตัวอักษรและการลดความชัดเจนของภาพชุดตัวอักษร CAPTCHA ตามที่ต้องการเสร็จสิ้นแล้ว
Postcondition :	ไฟล์ภาพชุดตัวอักษร CAPTCHA ที่สุ่มขึ้นตามความน่าจะเป็นทางภาษาในแบบจำลองเอ็นแกรม

ตารางที่ 3.5 (ต่อ)

Flow of Activities :	Actor	System
	1. ผู้ใช้กำหนดชื่อและรูปแบบของไฟล์ภาพชุดตัวอักษร CAPTCHA 2. ผู้ใช้กำหนดชื่อเท็กซ์ไฟล์ที่ใช้ในการเก็บชุดตัวอักษรเฉลยของ CAPTCHA ที่จะถูกสร้างขึ้น 3. ผู้ใช้เลือกไดเร็กทอรี่ที่ใช้เก็บไฟล์ผลลัพธ์ 4. ผู้ใช้ปรับปริมาณในการสร้าง CAPTCHA 5. ผู้ใช้สั่งให้สร้าง CAPTCHA	5.1 ระบบเรียกใช้ยูสเคส Generate Text เพื่อสุมสร้างชุดตัวอักษรตามค่าต่างๆ ที่ผู้ใช้ตั้งไว้ 5.2 ระบบเรียกใช้ยูสเคส Generate Image เพื่อสร้างภาพชุดตัวอักษรที่ถูกลดความชัดเจนตามค่าต่างๆ ที่ผู้ใช้ตั้งไว้ 5.3 ระบบบันทึกภาพชุดตัวอักษรเก็บไว้ในไฟล์รูปภาพตามรูปแบบที่ผู้ใช้ตั้งไว้ 5.4 ระบบบันทึกชุดตัวอักษรเฉลยไว้ในเท็กไฟล์ตามที่ผู้ใช้ตั้งไว้
Exceptional Condition :	6.1. ถ้ายังไม่ได้สร้างเอ็นแกรมระบบจะแจ้งเตือนผู้ใช้ว่า “ไม่สามารถสร้าง CAPTCHA ได้เนื่องจากยังไม่ได้สร้างเอ็นแกรม”	

ตารางที่ 3.6 คำอธิบายยูสเคสไดอะแกรมของ Generate Text

Use case Name :	Generate Text
Brief Description :	สุ่มสร้างชุดตัวอักษรตามค่าความน่าจะเป็นทางภาษา
Actors :	-
Precondition :	จะต้องทำการสร้างเอ็นแกรมเสร็จสิ้นแล้ว, และผู้ใช้ต้องทำการตั้งค่าต่างๆ ที่ใช้ในการสร้างชุดตัวอักษรเสร็จสิ้นแล้ว
Postcondition :	ชุดตัวอักษรที่สุ่มสร้างขึ้นตามความน่าจะเป็นทางภาษาในแบบจำลองเอ็นแกรม

ตารางที่ 3.7 คำอธิบายยูสเคสไดอะแกรมของ Generate Image

Use case Name :	Generate Image
Brief Description :	สร้างภาพชุดตัวอักษรและลดความชัดเจนของภาพ
Actors :	-

ตารางที่ 3.7 (ต่อ)

Precondition :	จะต้องทำการสร้างเอ็นแกรมเสร็จสิ้นแล้ว, และผู้ใช้ต้องทำการตั้งค่าต่างๆ ที่ใช้ในการลดความชัดเจนของภาพชุดตัวอักษร CAPTCHA ตามที่ต้องการเสร็จสิ้นแล้ว
Postcondition :	ชุดตัวอักษรที่สุ่มสร้างขึ้นตามความน่าจะเป็นทางภาษาในแบบจำลองเอ็นแกรม

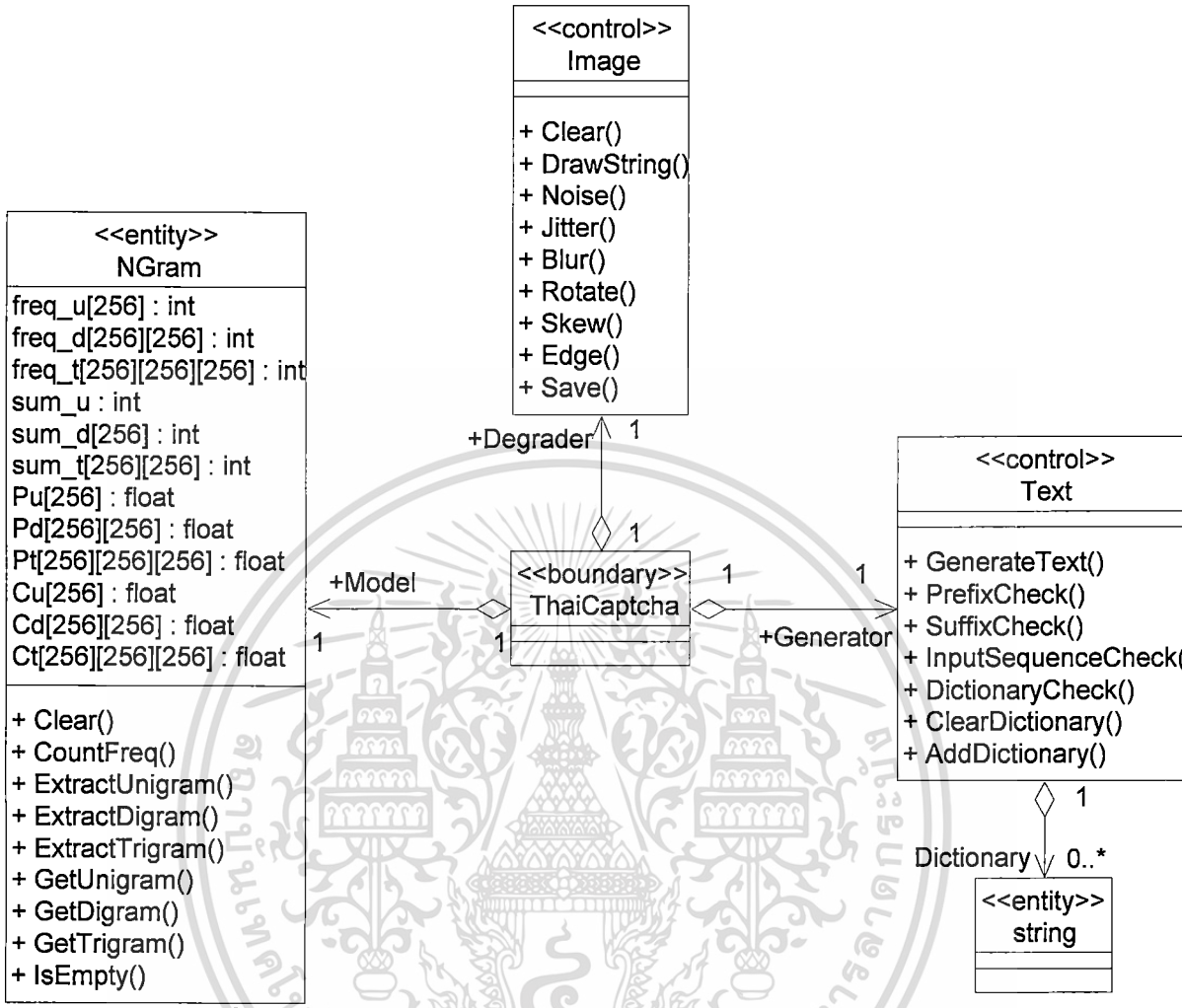
ตารางที่ 3.8 คำอธิบายยูสเคสไดอะแกรมของ Generate Web CAPTCHA

Use case Name :	Generate Web CAPTCHA	
Brief Description :	สร้าง CAPTCHA แบบออนไลน์ผ่านทางโปรโตคอล HTTP (เว็บ)	
Actors :	external system	
Precondition :	จะต้องทำการสร้างเอ็นแกรมเสร็จสิ้นแล้ว, และผู้ใช้ต้องทำการตั้งค่าต่างๆ ที่ใช้ในการสร้างชุดตัวอักษรและการลดความชัดเจนของภาพชุดตัวอักษร CAPTCHA ตามที่ต้องการเสร็จสิ้นแล้ว	
Postcondition :	ไฟล์ภาพชุดตัวอักษร CAPTCHA ที่สุ่มขึ้นตามความน่าจะเป็นทางภาษาในแบบจำลองเอ็นแกรมที่สามารถเข้าถึงแบบออนไลน์ได้ผ่านทางโปรโตคอล HTTP	
Flow of Activities :	Actor	System
	1. external system สั่งให้สร้าง CAPTCHA แบบออนไลน์	1.1 ระบบเรียกใช้ยูสเคสGenerate CAPTCHA เพื่อสร้างภาพชุดตัวอักษรตามค่าที่ตั้งไว้ 1.2 ระบบทำการบันทึกไฟล์ภาพเก็บไว้ในเว็บไดเร็กทอรี่ที่สามารถเข้าถึงแบบออนไลน์ด้วยโปรโตคอล HTTP ได้ 1.3 ระบบตอบ external system ด้วยชื่อไฟล์ภาพและชุดตัวอักษรเฉลยผ่านโปรโตคอล HTTP
Exceptional Condition :	1.1. ถ้ายังไม่ได้สร้างเอ็นแกรมระบบจะแจ้งเตือนไปยัง external system ว่า “ไม่สามารถสร้าง CAPTCHA ได้เนื่องจากยังไม่ได้สร้างเอ็นแกรม”	

ตารางที่ 3.9 คำอธิบายยูสเคสไดอะแกรมของ Evaluate CAPTCHA

Use case Name :	Evaluate CAPTCHA	
Brief Description :	ทำการประเมินประสิทธิภาพในการป้องกันของภาพชุดตัวอักษร CAPTCHA ที่สร้างขึ้น	
Actors :	User	
Precondition :	จะต้องติดตั้งโปรแกรม OCR ที่ระบบสร้าง CAPTCHA ภาษาไทยรองรับ (ABBY FineReader) ไว้บนคอมพิวเตอร์แล้ว, และต้องทำการสร้างภาพชุดตัวอักษร CAPTCHA ภาษาไทยไว้แล้ว	
Postcondition :	แสดงผลลัพธ์ของการประเมินว่า CAPTCHA ที่สร้างขึ้นสามารถถูกแปลงกลับด้วยโปรแกรม OCR ได้ถูกต้องเท่าไหร่	
Flow of Activities :	Actor	System
	1. ผู้ใช้เลือกไดเร็กทอรี่ของโปรแกรม OCR 2. ผู้ใช้สั่งให้ทำการประเมิน CAPTCHA ที่สร้างขึ้น	2.1 ระบบทำการสั่งให้โปรแกรม OCR ทำการแปลงไฟล์ภาพทั้งหมดที่ถูกสร้างขึ้นเป็นตัวอักษร 2.2 ระบบทำการตรวจสอบความถูกต้องของตัวอักษรที่ได้จากผลลัพธ์ของโปรแกรม OCR กับชุดตัวอักษร CAPTCHA ในไฟล์เฉลย 2.3 ระบบแสดงผลลัพธ์ของการประเมินว่า CAPTCHA ที่สร้างขึ้นสามารถถูกแปลงกลับด้วยโปรแกรม OCR ได้ถูกต้องเท่าไหร่
Exceptional Condition :	2.1. ถ้ายังไม่ได้สร้าง CAPTCHA ระบบจะแจ้งเตือนผู้ใช้ว่า “ไม่สามารถประเมิน CAPTCHA ได้เนื่องจากยังไม่ได้สร้าง CAPTCHA” 2.2 ถ้าไม่พบโปรแกรม OCR ในไดเร็กทอรี่ที่ผู้ใช้ตั้งไว้ระบบจะแจ้งเตือนผู้ใช้ว่า “ไม่สามารถประเมินได้เนื่องจากไม่พบโปรแกรม OCR”	

3.2 การออกแบบระบบด้วยคลาสไดอะแกรม



รูปที่ 3.2 คลาสไดอะแกรมของระบบสร้าง CAPTCHA ภาษาไทย

จากรูปที่ 3.2 เป็นการออกแบบระบบด้วยคลาสไดอะแกรมซึ่งแสดงอ็อบเจ็กของคลาสต่างๆ ที่มีอยู่ในระบบสร้าง CAPTCHA ภาษาไทย โดยแต่ละคลาสมีความหมายดังนี้

3.2.1 ThaiCaptcha

คลาส ThaiCaptcha คือคลาสหลักในการติดต่อกับผู้ใช้หรือระบบอื่นภายนอก ซึ่งประกอบรวมอ็อบเจ็กไว้ 3 อ็อบเจ็กคือ แบบจำลองทางภาษา (Model), ตัวสร้างชุดตัวอักษร (Generator), และตัวลดความชัดเจนของภาพ (Degrader) ซึ่งเป็นอ็อบเจ็กของคลาสดังนี้ คลาส NGram, คลาส Text, และคลาส Image ตามลำดับ

3.2.2 NGram

คลาส NGram คือคลาสที่ใช้สร้างและเก็บแบบจำลองทางภาษาเอ็นแกรมโดยจัดเก็บแบบจำลองความน่าจะเป็น ยูนิแกรม, ไบแกรม, และไตรแกรม ซึ่งประกอบด้วย ค่าความถี่, ความ

น่าจะเป็น, และความน่าจะเป็นสะสม ไว้ในตัวแปรโครงสร้างข้อมูลแบบอาร์เรย์หลายมิติที่ใช้รหัสของตัวอักษรเป็นอินเด็กซ์ของอาร์เรย์

ในการสร้างแบบจำลองเอ็นแกรมจะเริ่มจากการเรียกเมท็อด Clear() เพื่อล้างข้อมูลก่อน แล้วจึงเรียกเมท็อด CountFreq() เพื่อนับความถี่ของตัวอักษรจากข้อมูลที่ป้อนเป็นอินพุต ซึ่งได้มาจากไฟล์คลังข้อความแล้วจึงสั่งให้อ็อบเจ็กทำการคำนวณค่าความน่าจะเป็นให้กับแบบจำลองเอ็นแกรมด้วยเมท็อด ExtractUnigram(), ExtractDigram() และ ExtractTrigram()

สถานะของอ็อบเจ็ก NGram สามารถตรวจสอบได้ด้วยการเรียกเมท็อด IsEmpty() เพื่อดูว่าได้สร้างแบบจำลองเอ็นแกรมเสร็จแล้วหรือยัง ถ้ายังไม่มีข้อมูลก็หมายถึงแบบจำลองเอ็นแกรมยังไม่ได้ถูกสร้างขึ้นมา

3.2.3 Text

คลาส Text คือคลาสที่ใช้ในการสุ่มสร้างชุดตัวอักษรโดยอาศัยการสุ่มตามค่าความน่าจะเป็นที่ได้จากอ็อบเจ็ก NGram โดยใช้วิธีแปลงผกผันด้วยเมท็อด GenerateText() โดยมีเมท็อด InputSequenceCheck() เพื่อทำหน้าที่ตรวจสอบการป้อนอินพุตว่าชุดตัวอักษรที่สร้างขึ้นสามารถถูกป้อนกลับผ่านระบบปฏิบัติการได้

ส่วนเมท็อด PrefixCheck() และ SuffixCheck() มีไว้เพื่อการตรวจสอบตัวอักษรตัวแรกและตัวสุดท้ายว่ามีความคล้ายคลึงกับคำในภาษาไทยโดยทั่วไปหรือไม่

คลาส Text ยังประกอบไปด้วยคำในพจนานุกรม (Dictionary) เพื่อใช้ในการตรวจสอบว่าชุดตัวอักษรที่สร้างขึ้นตรงกับคำในพจนานุกรมหรือไม่ด้วยเมท็อด DictionaryCheck() และสามารถเพิ่มคำหรือลบคำในพจนานุกรมได้ด้วยเมท็อด AddDictionary() และ ClearDictionary()

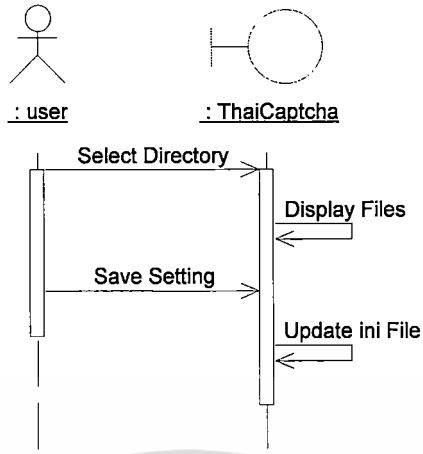
3.2.3 Image

คลาส Image คือคลาสที่ใช้ในการสร้างภาพชุดตัวอักษรด้วยเมท็อด DrawString() และสามารถทำการลดความชัดเจนของภาพโดยอาศัยการประมวลผลภาพด้วยเมท็อด 6 เมท็อดคือ Noise(), Jitter(), Blur(), Rotate(), Skew(), และ Edge() คลาส Image สามารถบันทึกภาพเก็บไว้ในไฟล์รูปภาพด้วยเมท็อด Save() ในรูปแบบต่างๆ อาทิเช่น .bmp, .jpg, .png, และ .gif เป็นต้น

3.3 การออกแบบลำดับการทำงานของระบบด้วยซีเควนซ์ไดอะแกรม

ซีเควนซ์ไดอะแกรมจะแสดงลำดับขั้นตอนการทำงานของระบบสร้าง CAPTCHA ภาษาไทยตามลำดับเหตุการณ์ก่อนหลัง เพื่อให้สามารถเข้าใจขั้นตอนการทำงานของระบบในแง่การปฏิสัมพันธ์ระหว่างอ็อบเจ็กต่างๆ ในระบบได้ชัดเจนยิ่งขึ้น ซึ่งจากการออกแบบยูสเคสไดอะแกรมจะสามารถแสดงถึงลำดับกิจกรรมต่างๆ ที่เกิดขึ้นในแต่ละอ็อบเจ็กได้ด้วยซีเควนซ์ไดอะแกรมดังนี้

3.3.1 ซีเควนซ์ไดอะแกรมของยูสเคส Change Corpus Setting

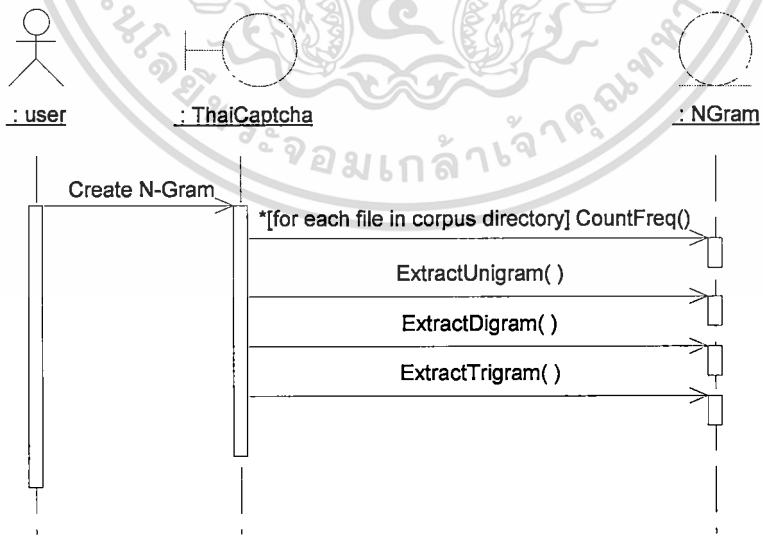


รูปที่ 3.3 ซีเควนซ์ไดอะแกรมของยูสเคส Change Corpus Setting

จากรูปที่ 3.3 แสดงลำดับขั้นตอนการทำงานของยูสเคส Change Corpus Setting ระหว่างผู้ใช้ (user) และอ็อบเจ็ก ThaiCaptcha โดยมีรายละเอียดดังนี้

1. ผู้ใช้เลือกไดเร็กทอรี่ที่ใช้เก็บคลังข้อความผ่าน GUI ของอ็อบเจ็ก ThaiCaptcha
2. อ็อบเจ็ก ThaiCaptcha แสดงไฟล์ที่มีอยู่ในไดเร็กทอรีที่ผู้ใช้เลือก
3. ผู้ใช้สั่งบันทึกการตั้งค่า
4. อ็อบเจ็ก ThaiCaptcha บันทึกไดเร็กทอรี่และชนิดของคลังข้อความเก็บในไฟล์คอนฟิก

3.3.2 ซีเควนซ์ไดอะแกรมของยูสเคส Create NGram

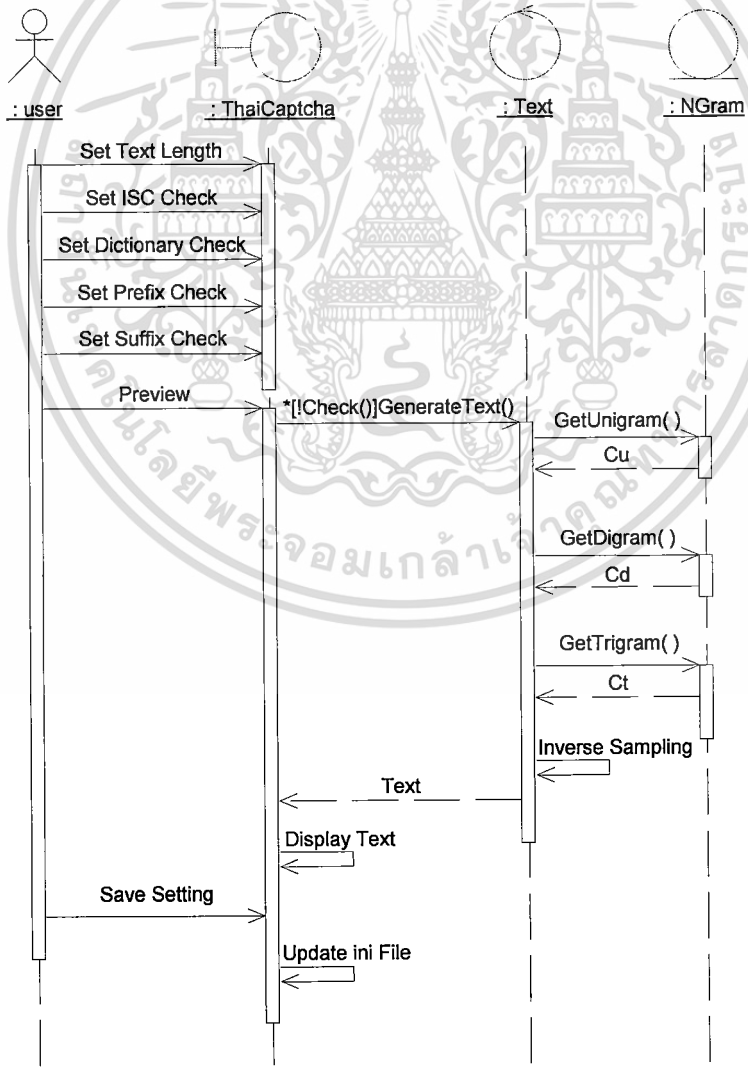


รูปที่ 3.4 ซีเควนซ์ไดอะแกรมของยูสเคส Create N-Gram

จากรูปที่ 3.4 แสดงลำดับขั้นตอนการทำงานของยูสเคส Create N-Gram ระหว่างผู้ใช้ (user), อ็อบเจ็ก ThaiCaptcha, และอ็อบเจ็ก NGram โดยมีรายละเอียดดังนี้

1. ผู้ใช้สั่งสร้างแบบจำลองเอ็นแกรมผ่าน GUI ของอ็อบเจ็ก ThaiCaptcha
2. อ็อบเจ็ก ThaiCaptcha ทำการทำซ้ำสำหรับทุกๆ ไฟล์ในไดเร็กทอรี่ที่เก็บคลังข้อความ โดยเรียกเมธอด CountFreq() ของอ็อบเจ็ก NGram เพื่อนับความถี่เอ็นแกรมของไฟล์คลังข้อความแต่ละไฟล์จนครบทุกไฟล์
3. อ็อบเจ็ก ThaiCaptcha เรียกเมธอด ExtractUnigram() ของอ็อบเจ็ก NGram เพื่อสกัดแบบจำลองยูนิแกรม
4. อ็อบเจ็ก ThaiCaptcha เรียกเมธอด ExtractBigram() ของอ็อบเจ็ก NGram เพื่อสกัดแบบจำลองไบแกรม
5. อ็อบเจ็ก ThaiCaptcha เรียกเมธอด ExtractTrigram() ของอ็อบเจ็ก NGram เพื่อสกัดแบบจำลองไตรแกรม

3.3.3 ซีเควนซ์ไดอะแกรมของยูสเคส Change Text Setting



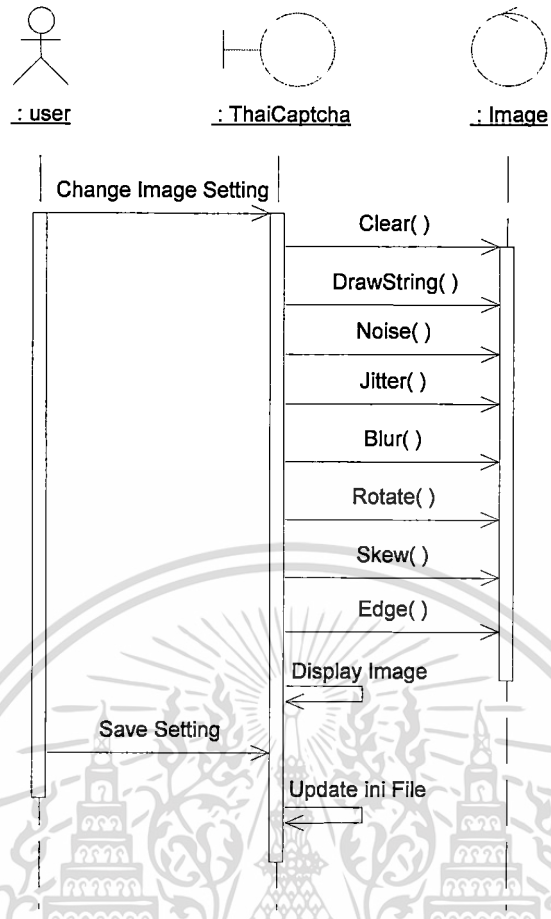
รูปที่ 3.5 ซีเควนซ์ไดอะแกรมของยูสเคส Change Text Setting

จากรูปที่ 3.5 แสดงลำดับขั้นตอนการทำงานของยูสเคส Change Text Setting ระหว่างผู้ใช้ (user), อ็อบเจ็ก ThaiCaptcha, อ็อบเจ็ก Text, และอ็อบเจ็ก NGram โดยมีรายละเอียดดังนี้

1. ผู้ใช้ปรับแต่งการตั้งค่าเกี่ยวกับการสร้างชุดตัวอักษร CAPTCHA ได้แก่ ปรับขนาดความยาวชุดตัวอักษร, ปรับการตรวจการป้อนอินพุต, ปรับการตรวจพจนานุกรม, ปรับการตรวจตัวอักษรตัวแรก, และปรับการตรวจตัวอักษรตัวสุดท้าย ผ่านทาง GUI ของอ็อบเจ็ก ThaiCaptcha
2. ผู้ใช้สั่งให้อ็อบเจ็ก ThaiCaptcha แสดงตัวอย่างชุดตัวอักษร CAPTCHA ตามการตั้งค่าที่ผู้ใช้ปรับไว้
3. อ็อบเจ็ก ThaiCaptcha ทำการทำซ้ำเรียกเมธอด GenerateText() ของอ็อบเจ็ก Text เพื่อสุ่มสร้างชุดตัวอักษร CAPTCHA จนกว่าจะได้ชุดตัวอักษรที่ผ่านการตรวจสอบตามที่ผู้ใช้ตั้งไว้ โดยขั้นตอนการทำงานของเมธอด GenerateText() ในแต่ละรอบมีรายละเอียดดังนี้
   3.1 อ็อบเจ็ก Text เรียกเมธอด GetUnigram() จากอ็อบเจ็ก NGram เพื่อขอเข้าถึงค่าความน่าจะเป็นสะสมของแบบจำลองยูนิแกรม (Cu)
   3.2 อ็อบเจ็ก Text เรียกเมธอด GetDigram() จากอ็อบเจ็ก NGram เพื่อขอเข้าถึงค่าความน่าจะเป็นสะสมของแบบจำลองไบแกรม (Cd)
   3.3 อ็อบเจ็ก Text เรียกเมธอด GetDigram() จากอ็อบเจ็ก NGram เพื่อขอเข้าถึงค่าความน่าจะเป็นสะสมของแบบจำลองไบแกรม (Ct)
   3.4 อ็อบเจ็ก Text ทำการสุ่มสร้างชุดตัวอักษรด้วยวิธีการ Inverse Transform จากค่าความน่าจะเป็นสะสมที่ได้จากขั้นตอนที่ 3.1 ถึง 3.3 แล้วส่งชุดตัวอักษรผลลัพธ์กลับให้อ็อบเจ็ก ThaiCaptcha
4. อ็อบเจ็ก ThaiCaptcha แสดงชุดตัวอักษรที่ผ่านการตรวจสอบแล้ว (ถ้าไม่ผ่านจะทำซ้ำข้อ 3 จนกว่าจะผ่าน)
5. ผู้ใช้สั่งบันทึกการตั้งค่า
6. อ็อบเจ็ก ThaiCaptcha บันทึกการตั้งค่าเกี่ยวกับการสร้างชุดตัวอักษร CAPTCHA เก็บไว้ในไฟล์คอนฟิก

3.3.4 ซีเควนซ์ไดอะแกรมของยูสเคส Change Image Setting

ในรูปถัดไปคือรูปที่ 3.6 ได้แสดงภาพแบบจำลองลำดับขั้นตอนการทำงานของยูสเคส Change Image Setting ดังนี้



รูปที่ 3.6 ซีเควนซ์ไดอะแกรมของยูสเคส Change Image Setting

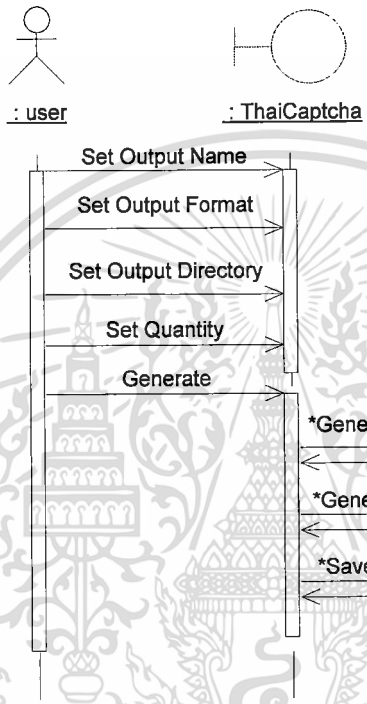
โดยมีรายละเอียดในแต่ละลำดับขั้นตอนการทำงานระหว่างผู้ใช้ (user), อ็อบเจ็ก ThaiCaptcha, อ็อบเจ็ก Text, และอ็อบเจ็ก NGram ดังนี้

1. ผู้ใช้ปรับแต่งการตั้งค่าเกี่ยวกับการลดทอนความชัดเจนภาพชุดตัวอักษร CAPTCHA ได้แก่ ปรับการเติมสัญญาณรบกวน (Noise), ปรับการสั่นภาพ (Jitter), ปรับการทำให้ภาพมัว (Blur), ปรับการหมุนภาพ (Rotate), ปรับการบิดภาพ (Skew), และปรับการเน้นขอบ (Edge) ผ่านทาง GUI ของอ็อบเจ็ก ThaiCaptcha
2. อ็อบเจ็ก ThaiCaptcha ทำการสร้างภาพใหม่ขึ้น แล้วเรียกเมธอดของอ็อบเจ็ก Image เพื่อวาดภาพชุดตัวอักษรและลดทอนความชัดเจนของภาพโดยมีรายละเอียดดังนี้
   2.1 เรียกเมธอด Clear() ของอ็อบเจ็ก Image เพื่อวาดพื้นหลังด้วยสีขาวทั้งหมด
   2.2 เรียกเมธอด DrawString() ของอ็อบเจ็ก Image เพื่อวาดชุดตัวอักษร
   2.3 เรียกเมธอด Noise() ของอ็อบเจ็ก Image เพื่อเติมสัญญาณรบกวนตามค่าที่ตั้งไว้
   2.4 เรียกเมธอด Jitter() ของอ็อบเจ็ก Image เพื่อสั่นภาพตามค่าที่ตั้งไว้
   2.5 เรียกเมธอด Blur() ของอ็อบเจ็ก Image เพื่อทำให้ภาพมัวลงตามค่าที่ตั้งไว้
   2.6 เรียกเมธอด Rotate() ของอ็อบเจ็ก Image เพื่อหมุนภาพตามองศาที่ตั้งไว้
   2.7 เรียกเมธอด Skew() ของอ็อบเจ็ก Image เพื่อบิดภาพตามค่าที่ตั้งไว้

2.8 เรียกเมธอด Edge() ของอ็อบเจ็ก Image เพื่อเน้นขอบภาพตามค่าที่ตั้งไว้

3. อ็อบเจ็ก ThaiCaptcha แสดงภาพ CAPTCHA ที่สร้างไว้ในขั้นตอนที่ 2
4. ผู้ใช้สั่งบันทึกการตั้งค่า
5. อ็อบเจ็ก ThaiCaptcha บันทึกการตั้งค่าเกี่ยวกับการสร้างภาพ CAPTCHA เก็บไว้ในไฟล์คอนฟิก

3.3.5 ซีเควนซ์ไดอะแกรมของยูสเคส Generate CAPTCHA

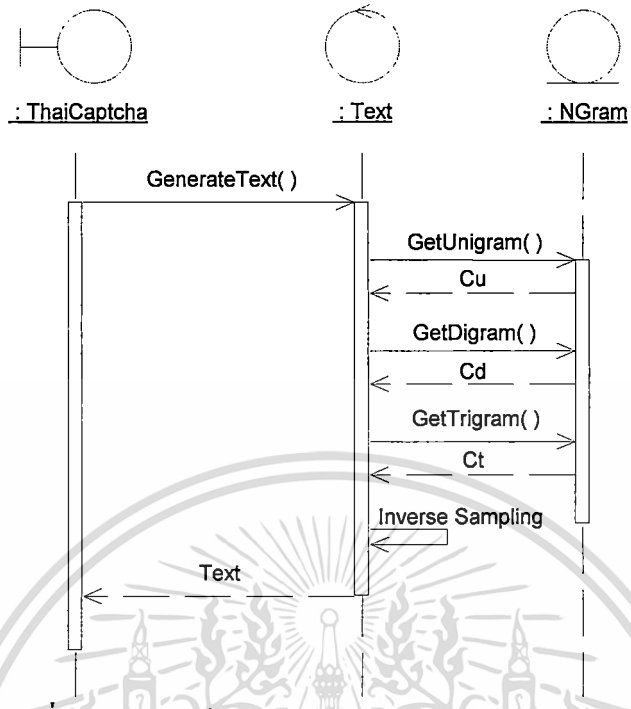


รูปที่ 3.7 ซีเควนซ์ไดอะแกรมของยูสเคส Generate CAPTCHA

จากรูปที่ 3.7 แสดงลำดับขั้นตอนการทำงานของยูสเคส Generate CAPTCHA ระหว่างผู้ใช้ (user), และอ็อบเจ็ก ThaiCaptcha โดยมีรายละเอียดดังนี้

1. ผู้ใช้ปรับแต่งการตั้งค่าเกี่ยวกับเอาพุตของการสร้าง CAPTCHA ได้แก่ชื่อไฟล์, รูปแบบไฟล์, ไดเร็กทอรีที่ใช้เก็บ รวมถึงตั้งจำนวนที่ต้องการให้สร้างผ่านทาง GUI ของอ็อบเจ็ก ThaiCaptcha
2. ผู้ใช้สั่งสร้าง CAPTCHA ผ่านทาง GUI ของอ็อบเจ็ก ThaiCaptcha
3. อ็อบเจ็ก ThaiCaptcha ทำการทำซ้ำเพื่อสุ่มสร้างชุดตัวอักษร, สร้างและลดความชัดเจนภาพชุดตัวอักษร, และบันทึกเก็บไว้ในไฟล์เอาท์พุต ตามจำนวนที่ผู้ใช้ตั้งไว้

3.3.6 ซีเควนซ์ไดอะแกรมของยูสเคส Generate CAPTCHA



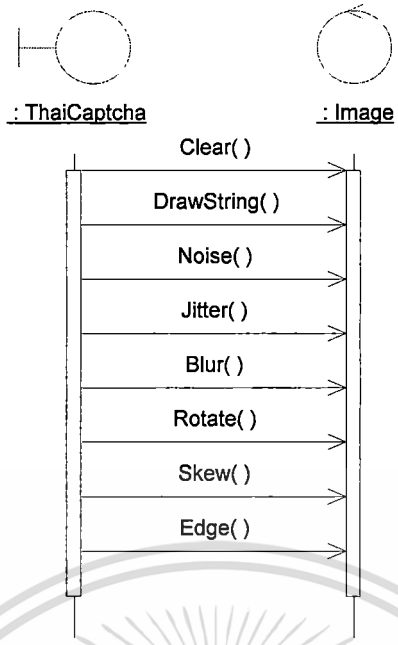
รูปที่ 3.8 ซีเควนซ์ไดอะแกรมของยูสเคส Generate Text

จากรูปที่ 3.8 แสดงลำดับขั้นตอนการทำงานของยูสเคส Generate Text อ็อบเจ็ก ThaiCaptcha, อ็อบเจ็ก Text, และอ็อบเจ็ก NGram โดยมีรายละเอียดดังนี้

1. อ็อบเจ็ก ThaiCaptcha เรียกเมธอด GenerateText() ของอ็อบเจ็ก Text เพื่อสุ่มสร้างชุดตัวอักษร CAPTCHA
2. อ็อบเจ็ก Text เรียกเมธอด GetUnigram() จากอ็อบเจ็ก NGram เพื่อขอเข้าถึงค่าความน่าจะเป็นสะสมของแบบจำลองยูนิแกรม (Cu)
3. อ็อบเจ็ก Text เรียกเมธอด GetDigram() จากอ็อบเจ็ก NGram เพื่อขอเข้าถึงค่าความน่าจะเป็นสะสมของแบบจำลองไบแกรม (Cd)
4. อ็อบเจ็ก Text เรียกเมธอด GetDigram() จากอ็อบเจ็ก NGram เพื่อขอเข้าถึงค่าความน่าจะเป็นสะสมของแบบจำลองไบแกรม (Ct)
5. อ็อบเจ็ก Text ทำการสุ่มสร้างชุดตัวอักษรด้วยวิธีการ Inverse Transform จากค่าความน่าจะเป็นสะสมของแบบจำลองเอ็นแกรม แล้วส่งผลลัพธ์กลับให้อ็อบเจ็ก ThaiCaptcha

3.3.7 ซีเควนซ์ไดอะแกรมของยูสเคส Generate CAPTCHA

ในรูปถัดไปคือรูปที่ 3.9 ได้แสดงภาพแบบจำลองลำดับขั้นตอนการทำงานของยูสเคส Generate Image ดังนี้



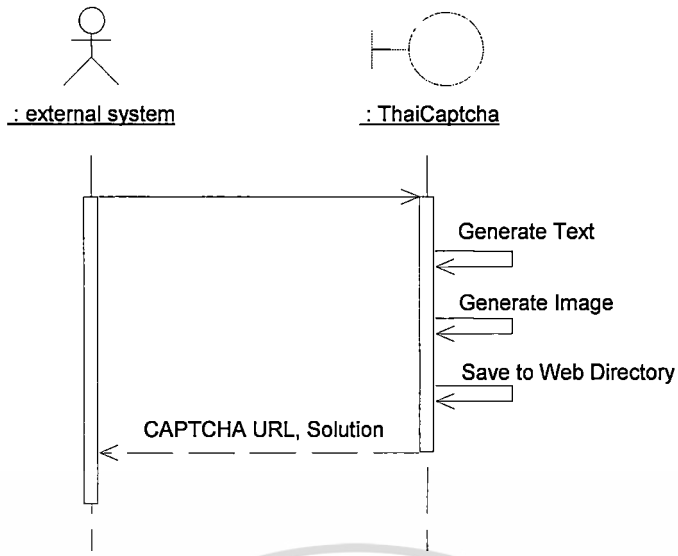
รูปที่ 3.9 ซีเควนซ์ไดอะแกรมของยูสเคส Generate Image

โดยมีรายละเอียดในแต่ละลำดับขั้นตอนการทำงานระหว่างอ็อบเจ็ก ThaiCaptcha และอ็อบเจ็ก Image ดังนี้

1. อ็อบเจ็ก ThaiCaptcha ทำการสร้างภาพใหม่ขึ้น
2. อ็อบเจ็ก ThaiCaptcha เรียกเมธอด Clear() ของอ็อบเจ็ก Image เพื่อวาดพื้นหลังด้วยสีขาวทั้งหมด
3. อ็อบเจ็ก ThaiCaptcha เรียกเมธอด DrawString() ของอ็อบเจ็ก Image เพื่อวาดชุดตัวอักษร CAPTCH
4. อ็อบเจ็ก ThaiCaptcha เรียกเมธอด Noise() ของอ็อบเจ็ก Image เพื่อเติมสัญญาณรบกวน
5. อ็อบเจ็ก ThaiCaptcha เรียกเมธอด Jitter() ของอ็อบเจ็ก Image เพื่อสั่นภาพ
6. อ็อบเจ็ก ThaiCaptcha เรียกเมธอด Blur() ของอ็อบเจ็ก Image เพื่อทำให้ภาพมัวลง
7. อ็อบเจ็ก ThaiCaptcha เรียกเมธอด Rotate() ของอ็อบเจ็ก Image เพื่อหมุนภาพ
8. อ็อบเจ็ก ThaiCaptcha เรียกเมธอด Skew() ของอ็อบเจ็ก Image เพื่อบิดภาพ
9. อ็อบเจ็ก ThaiCaptcha เรียกเมธอด Edge() ของอ็อบเจ็ก Image เพื่อเน้นขอบภาพ

3.3.8 ซีเควนซ์ไดอะแกรมของยูสเคส Generate CAPTCHA

ในรูปถัดไปคือรูปที่ 3.10 ได้แสดงภาพแบบจำลองลำดับขั้นตอนการทำงานของยูสเคส Generate Web CAPTCHA ดังนี้

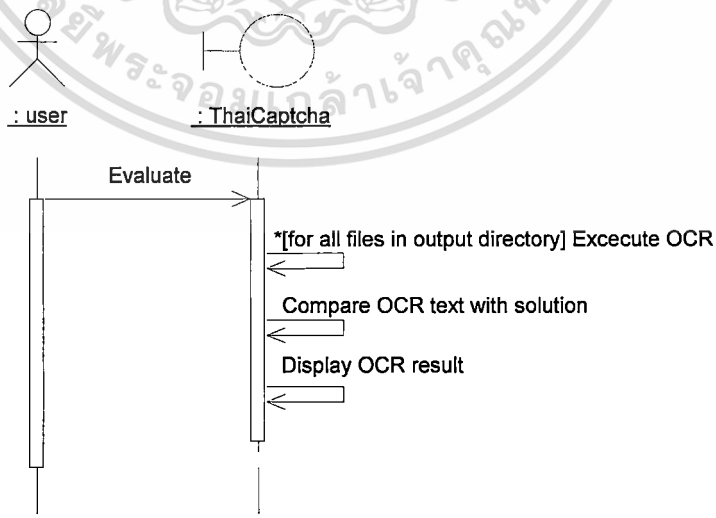


รูปที่ 3.10 ซีเควนซ์ไดอะแกรมของยูสเคส Generate Web CAPTCHA

โดยมีรายละเอียดในแต่ละลำดับขั้นตอนการทำงานระหว่างระบบอื่นภายนอก (external system) และอ็อบเจ็ก ThaiCaptcha ดังนี้

1. ระบบอื่นภายนอกสั่งสร้าง CAPTCHA ผ่านเว็บเซิร์ฟเวอร์ของอ็อบเจ็ก ThaiCaptcha
2. อ็อบเจ็ก ThaiCaptcha ทำการสุ่มสร้างชุดตัวอักษร, สร้างและลดความชัดเจนภาพชุดตัวอักษร, และบันทึกไฟล์เอาท์พุตเก็บไว้ในไดเร็กทอรี่ที่สามารถเข้าถึงผ่านอินเตอร์เฟสแบบเว็บได้
3. อ็อบเจ็ก ThaiCaptcha ตอบกลับด้วย URL ที่ลิงค์ไปยังไฟล์เอาท์พุตพร้อมทั้งเฉลย

3.3.9 ซีเควนซ์ไดอะแกรมของยูสเคส Generate CAPTCHA



รูปที่ 3.11 ซีเควนซ์ไดอะแกรมของยูสเคส Evaluate CAPTCHA

จากรูปที่ 3.11 แสดงลำดับขั้นตอนการทำงานของยูสเคส Evaluate CAPTCHA ระหว่างผู้ใช้ (user), และอ็อบเจ็ก ThaiCaptcha โดยมีรายละเอียดดังนี้

1. ผู้ใช้สั่งทดลองประเมินผล CAPTCHA ผ่านทาง GUI ของอ็อบเจ็ก ThaiCaptcha
2. อ็อบเจ็ก ThaiCaptcha ทำการทำซ้ำกับทุกๆ ไฟล์ในไดเร็กทอรี่เอาท์พุต เพื่อให้โปรแกรม OCR แปลงภาพชุดตัวอักษร CAPTCHA ที่ถูกสร้างขึ้นก่อนหน้านี้
3. อ็อบเจ็ก ThaiCaptcha ตรวจสอบผลการทำ OCR กับเฉลย
4. อ็อบเจ็ก ThaiCaptcha แสดงผลการประเมิน

บทที่ 4
การพัฒนาระบบ

สำหรับการพัฒนาระบบสร้าง CAPTCHA ภาษาไทย สามารถอธิบายโดยแบ่งออกเป็นสามส่วนคือ เครื่องมือที่ใช้ในการพัฒนาระบบ, แหล่งข้อมูลที่ใช้ในการพัฒนาระบบ, และรายละเอียดในการพัฒนาส่วนต่างๆ ของระบบ โดยมีรายละเอียดดังต่อไปนี้

4.1 เครื่องมือที่ใช้ในการพัฒนาระบบ

ในการพัฒนาระบบสร้าง CAPTCHA ภาษาไทยนั้นจำเป็นต้องใช้เครื่องมือในการพัฒนา โดยแบ่งเป็นฮาร์ดแวร์และซอฟต์แวร์ดังนี้

4.1.1 ฮาร์ดแวร์

ในการพัฒนาระบบใช้เครื่องคอมพิวเตอร์ที่มีคุณสมบัติดังนี้

- CPU : Intel Core 2 Duo CPU T6500 @ 2.1 GHz
- RAM : 3.0 GB
- OS : Windows XP
- Hard Disk : 450 GB

4.1.2 ซอฟต์แวร์

ในการพัฒนาระบบใช้ซอฟต์แวร์ดังนี้

- Microsoft Windows XP
- Microsoft Visual C++ 6
- Rational Rose 2003 (สำหรับออกแบบ)
- ABBY FineReader 10 (สำหรับทำการประเมิน)

4.2 แหล่งข้อมูลที่ใช้ในการพัฒนาระบบ

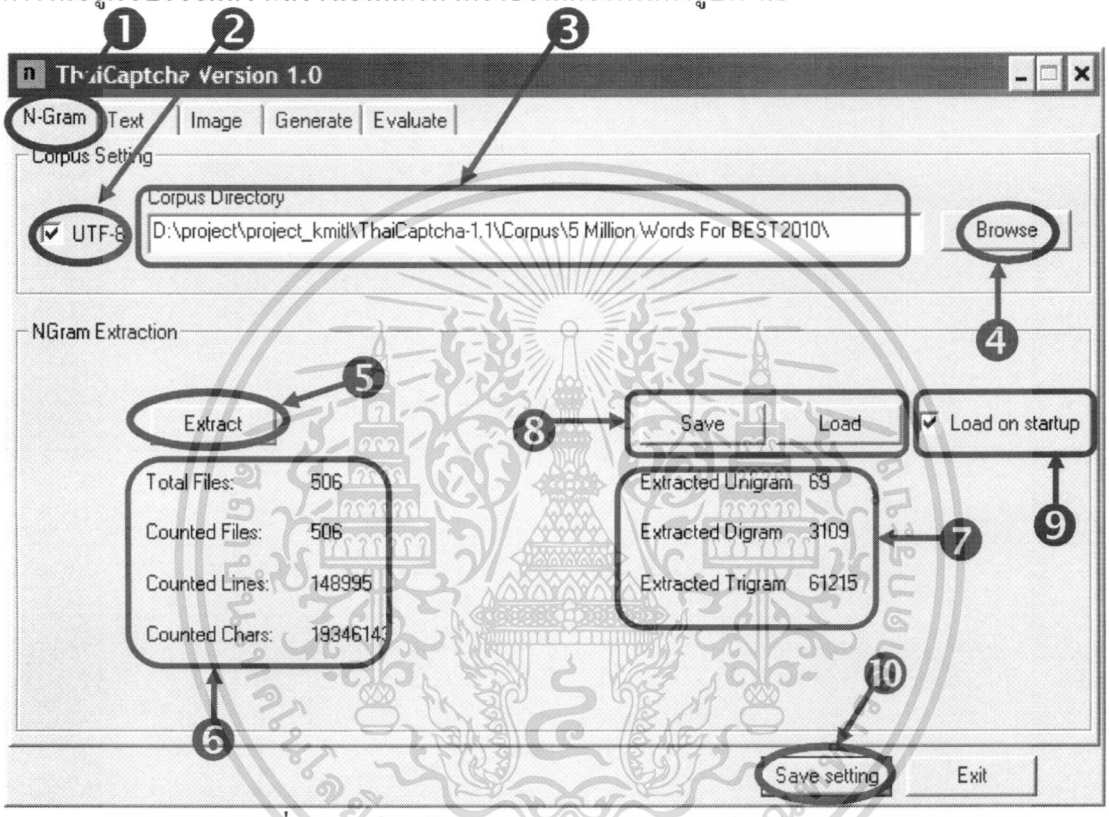
แหล่งข้อมูลที่นำมาใช้ในการสร้างแบบจำลองเอ็นแกรมเพื่อใช้ในระบบสร้าง CAPTCHA ภาษาไทยนั้นสามารถใช้ได้กับคลังข้อความโดยทั่วไปที่จัดเก็บในรูปแบบของเท็กซ์ไฟล์ทั้งในแบบ UTF-8 และแบบปกติ (Windows-874) โดยในการพัฒนานี้ทดสอบใช้งานร่วมกับคลังข้อความจากโครงการ BEST จำนวน 5 ล้านคำ (เนคเทค, 2553) ซึ่งสามารถใช้งานร่วมกับระบบสร้าง CAPTCHA ภาษาไทยได้เป็นอย่างดี

4.3 รายละเอียดในการพัฒนาส่วนต่างๆ ของระบบ

ในการพัฒนาระบบสร้าง CAPTCHA ภาษาไทย ได้แบ่งส่วนประกอบของระบบออกเป็น 5 ส่วน โดยมีรายละเอียดดังต่อไปนี้

4.3.1 ส่วนสร้างเอ็นแกรม

เมื่อผู้ใช้ติดตั้งไฟล์คลังข้อความไว้บนคอมพิวเตอร์ที่ระบบสร้าง CAPTCHA ภาษาไทย ทำงานอยู่เรียบร้อยแล้ว ก็สร้างเอ็นแกรมโดยใช้อินเตอร์เฟสดังรูปที่ 4.1



รูปที่ 4.1 ภาพหน้าจอการทำงานของส่วนสร้างเอ็นแกรม

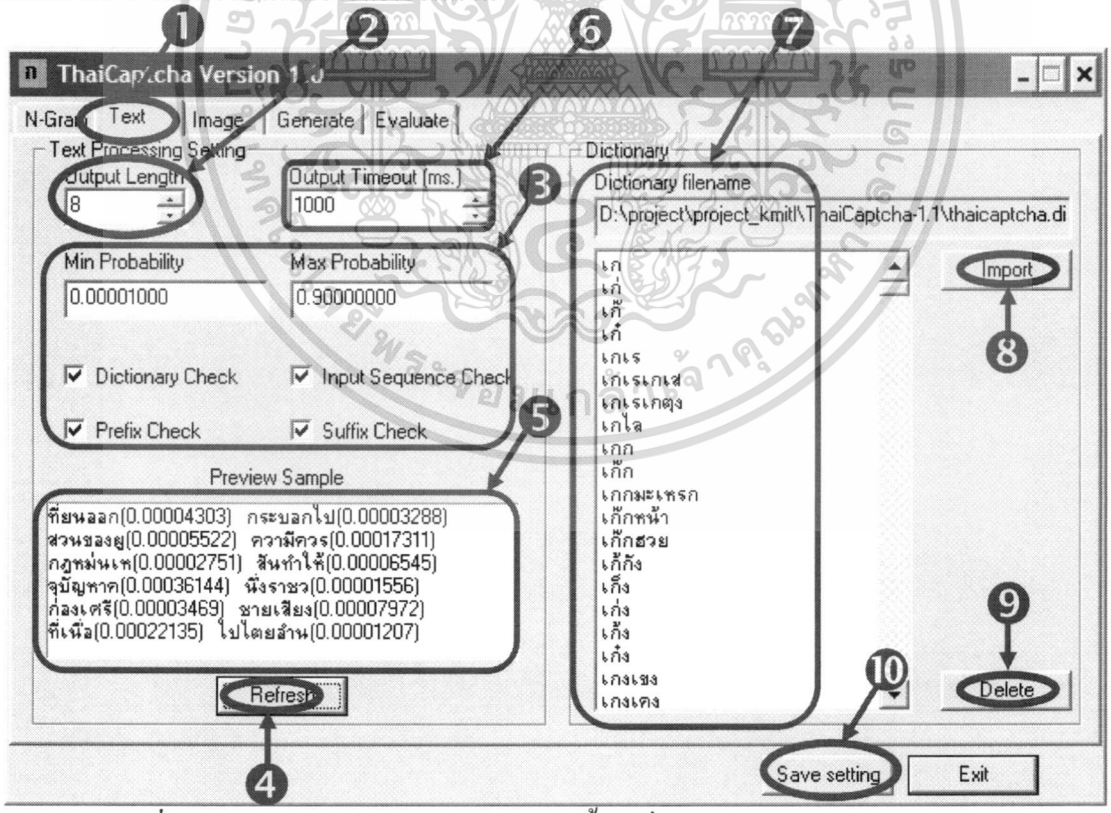
จากรูปที่ 4.1 แสดงภาพหน้าจอการทำงานของส่วนสร้างเอ็นแกรม ซึ่งประกอบด้วยส่วนต่างๆ สิบส่วนดังต่อไปนี้

1. ปุ่มเลือกแท็ปของส่วนสร้างเอ็นแกรม ผู้ใช้สามารถเข้าถึงการทำงานในส่วนนี้ได้ด้วยการคลิ๊กที่แท็ปนี้
2. ปุ่มเลือกรูปแบบของคลังข้อความ ผู้ใช้สามารถเลือกได้ว่าไฟล์ในคลังข้อความที่ใช้ถูกจัดเก็บไว้ในรูปแบบ UTF-8 หรือแบบปกติ (Windows-874)
3. ช่องสำหรับให้ผู้ใช้กรอกที่อยู่ไดเร็กทอรี่ของคลังข้อความ
4. ปุ่มค้นหาไดเร็กทอรี่ของคลังข้อความ
5. ปุ่มสั่งให้ระบบสกัดสร้างแบบจำลองเอ็นแกรมจากไฟล์คลังข้อความทั้งหมดที่อยู่ในไดเร็กทอรี่ที่ผู้ใช้ตั้งไว้

6. แสดงสถานะการนับความถี่ของตัวอักษรจากไฟล์คลังข้อความ โดยแสดงตัวนับจำนวนตัวอักษร, บรรทัด, และไฟล์ที่นับเสร็จสิ้นแล้ว รวมถึงจำนวนไฟล์ที่มีทั้งหมด
7. แสดงสถานะการคำนวณค่าความน่าจะเป็นจากความถี่ที่นับได้ โดยแสดงตัวนับจำนวนแกรมของตัวอักษรที่คำนวณได้ในระดับ ยูนิแกรม, ไบแกรม, และไตรแกรม
8. ปุ่มสั่งให้ระบบบันทึกแบบจำลองเอ็นแกรมที่สกัดได้เก็บไว้ในไฟล์ และปุ่มสั่งให้ระบบโหลดแบบจำลองเอ็นแกรมจากไฟล์
9. ปุ่มเลือกว่าจะให้ระบบโหลดแบบจำลองเอ็นแกรมจากไฟล์ ขึ้นมาทันทีเมื่อเริ่มใช้งานระบบหรือไม่
10. ปุ่มสั่งให้ระบบบันทึกค่าที่ผู้ใช้ตั้งไว้ในขั้นตอนที่ 2 ถึง 4 เก็บไว้ในไฟล์คอนฟิกเพื่อให้สามารถโหลดค่าที่เคยตั้งไว้กลับมาใช้งานได้ทันทีเมื่อเริ่มใช้งานระบบครั้งต่อไปโดยผู้ใช้ไม่ต้องตั้งค่าใหม่

4.3.2 ส่วนตั้งค่าที่ใช้ในการสุ่มสร้างชุดตัวอักษร

เมื่อแบบจำลองเอ็นแกรมถูกสร้างเสร็จแล้ว ระบบก็พร้อมจะสร้าง CAPTCHA ภาษาไทยได้ทันที แต่หากผู้ใช้ต้องการปรับเปลี่ยนค่าต่างๆ ที่ใช้ในการสุ่มสร้างชุดตัวอักษรให้แตกต่างจากค่าเริ่มต้น ก็สามารถทำได้โดยมีรายละเอียดดังนี้



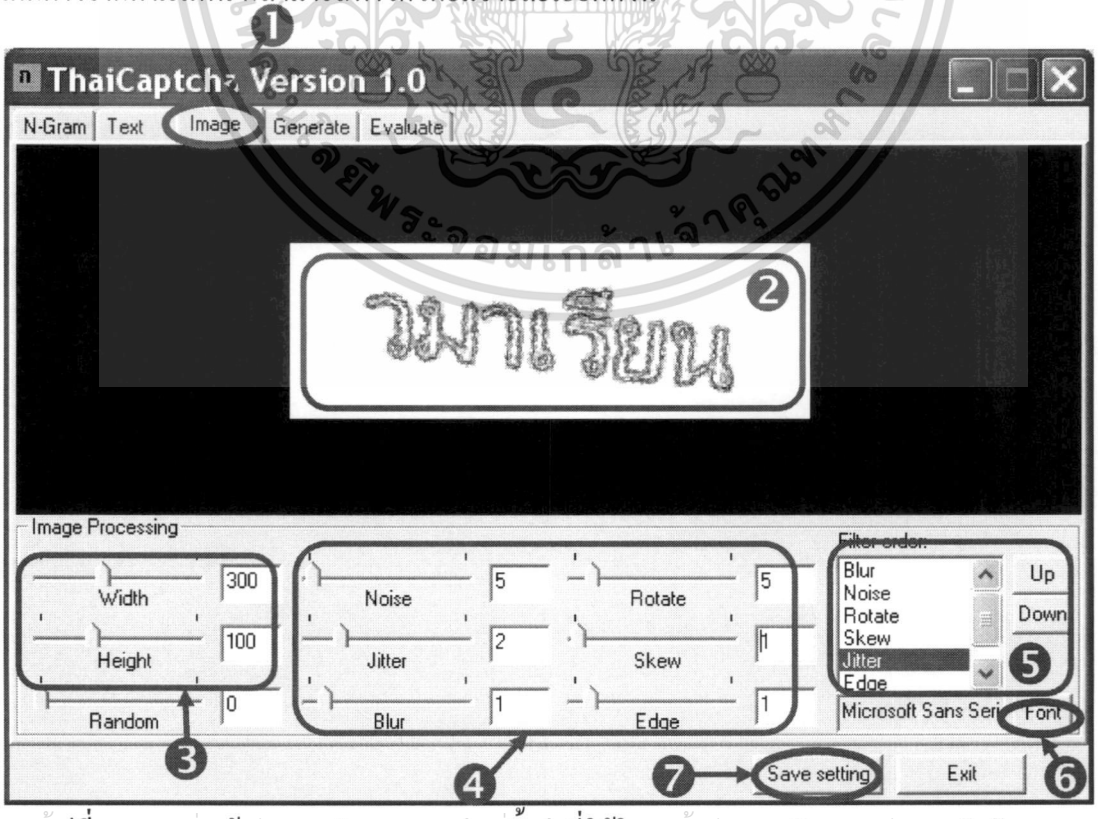
รูปที่ 4.2 ภาพหน้าจอการทำงานของส่วนตั้งค่าที่ใช้ในการสุ่มสร้างชุดตัวอักษร

จากรูปที่ 4.2 แสดงภาพหน้าจอการทำงานของส่วนตั้งค่าที่ใช้ในการสุ่มสร้างชุดตัวอักษร ซึ่งประกอบด้วยส่วนต่างๆ สิบส่วนดังต่อไปนี้

1. ปุ่มเลือกแท็ปของส่วนตั้งค่าที่ใช้ในการสุ่มสร้างชุดตัวอักษร ผู้ใช้สามารถเข้าถึงการทำงานในส่วนนี้ได้ด้วยการคลิ๊กที่แท็ปนี้
2. ช่องให้ผู้ใช้ป้อนความยาวของชุดตัวอักษร CAPTCHA ที่ต้องการให้ระบบสร้าง
3. เลือกการตรวจสอบว่าต้องการตรวจสอบชุดตัวอักษรที่สร้างขึ้นอย่างไรบ้าง ได้แก่ ตรวจค่าความน่าจะเป็นของชุดตัวอักษร (ต่ำสุดและสูงสุด), ตรวจคำในพจนานุกรม, ตรวจการป้อนอินพุต, ตรวจตัวอักษรตัวแรก, และตรวจตัวอักษรตัวสุดท้าย
4. ปุ่มสั่งให้ระบบสุ่มสร้างชุดตัวอักษรตามค่าที่ผู้ตั้งไว้ให้ผู้ใช้ดูเป็นตัวอย่าง
5. ส่วนที่ใช้แสดงตัวอย่างตัวอย่างชุดตัวอักษรพร้อมค่าความน่าจะเป็น
6. ช่องให้ผู้ใช้กำหนดระยะเวลาสูงสุดที่ระบบจะใช้ในการสุ่มสร้างชุดตัวอักษร
7. ส่วนแสดงชื่อไฟล์พจนานุกรมของระบบและคำที่มีอยู่ข้างใน
8. ปุ่มสั่งให้นำเข้าคำจากเท็กซ์ไฟล์อื่นเข้ามาในพจนานุกรมของระบบ
9. ปุ่มสั่งให้ลบคำที่เลือกออกจากพจนานุกรมของระบบ
10. ปุ่มสั่งให้ระบบบันทึกค่าที่ผู้ใช้ตั้งไว้ในขั้นตอนที่ 2 และ 3 เก็บไว้ในไฟล์คอนฟิก

4.3.3 ส่วนตั้งค่าที่ใช้ในการลดความชัดของภาพชุดตัวอักษร

เมื่อแบบจำลองเอ็นแกรมถูกสร้างเสร็จแล้ว ระบบก็พร้อมจะสร้าง CAPTCHA ภาษาไทยได้ทันที แต่หากผู้ใช้ต้องการปรับเปลี่ยนค่าต่างๆ ที่ใช้ในการลดความชัดของภาพชุดตัวอักษรให้แตกต่างจากค่าเริ่มต้น ก็สามารถทำได้โดยมีรายละเอียดดังนี้

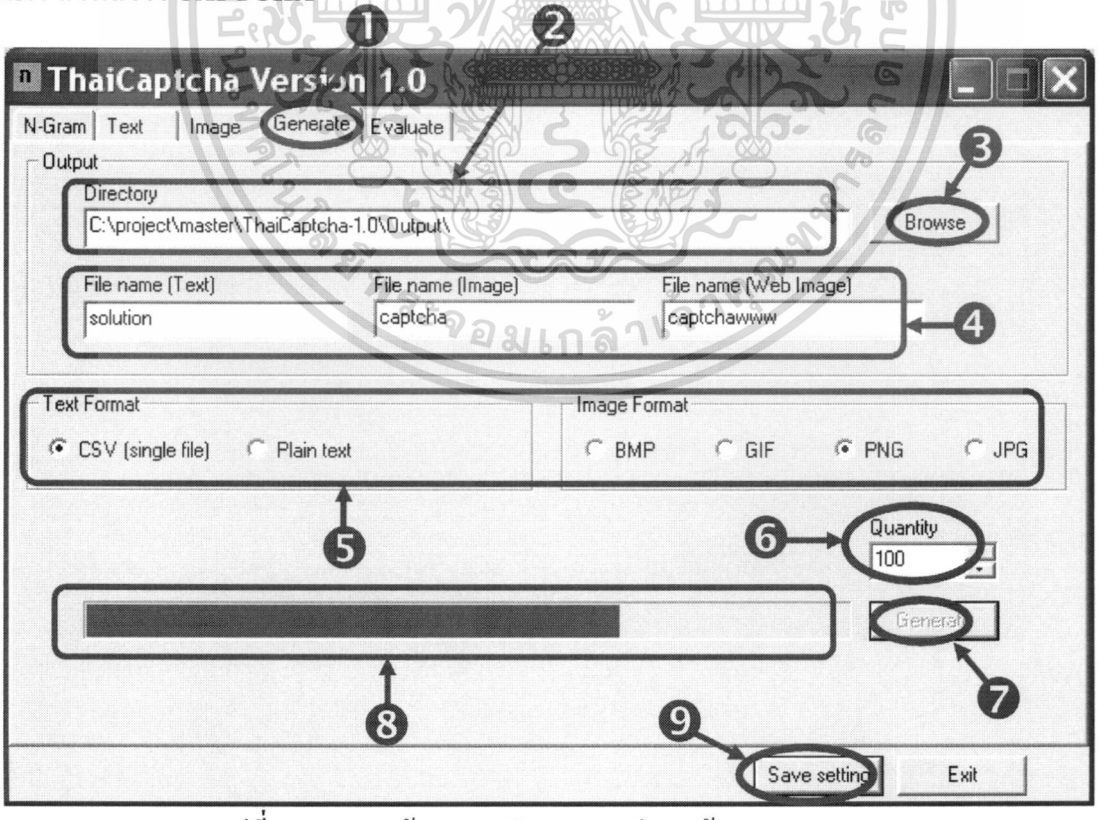


รูปที่ 4.3 ภาพหน้าจอการทำงานของส่วนตั้งค่าที่ใช้ในการลดความชัดของภาพชุดตัวอักษร

จากรูปที่ 4.3 แสดงภาพหน้าจอการทำงานของส่วนตั้งค่าที่ใช้ในการลดความชัดของภาพชุดตัวอักษร ซึ่งประกอบด้วยส่วนต่างๆ เจ็ดส่วนดังต่อไปนี้

1. ปุ่มเลือกแท็ปของส่วนตั้งค่าที่ใช้ในการลดความชัดของภาพชุดตัวอักษร ผู้ใช้สามารถเข้าถึงการทำงานในส่วนนี้ได้ด้วยการคลิ๊กที่แท็ปนี้
2. บริเวณสำหรับแสดงตัวอย่างภาพชุดตัวอักษรตามค่าต่างๆ ที่ผู้ใช้ตั้ง ผู้ใช้สามารถดับเบิ้ลคลิ๊กบริเวณนี้เพื่อสุ่มสร้างชุดตัวอักษรที่ใช้แสดงตัวอย่างใหม่ได้
3. ช่องสำหรับตั้งค่าความกว้างและความสูงของภาพ
4. ช่องสำหรับตั้งค่าของการประมวลผลภาพเพื่อลดความชัดของภาพชุดตัวอักษร ประกอบด้วยการตั้งค่า สัญญาณรบกวน (noise), การสั่นของภาพ (jitter), การทำให้ภาพมัวลง (blur), การหมุนภาพ (rotate), การบิดภาพ (skew), การเน้นขอบ (edge)
5. ส่วนปรับลำดับการทำงานของการลดความชัดของภาพชุดตัวอักษร ระบบจะทำการประมวลผลภาพในแต่ละแบบตามลำดับจากบนลงล่าง
6. ปุ่มเลือกรูปแบบตัวอักษร
7. ปุ่มสั่งให้ระบบบันทึกค่าที่ผู้ใช้ตั้งไว้ในขั้นตอนที่ 3 ถึง 6 เก็บไว้ในไฟล์คอนฟิก เพื่อให้สามารถโหลดค่าที่เคยตั้งไว้กลับมาใช้งานได้ทันทีเมื่อเริ่มใช้งานระบบครั้งต่อไป

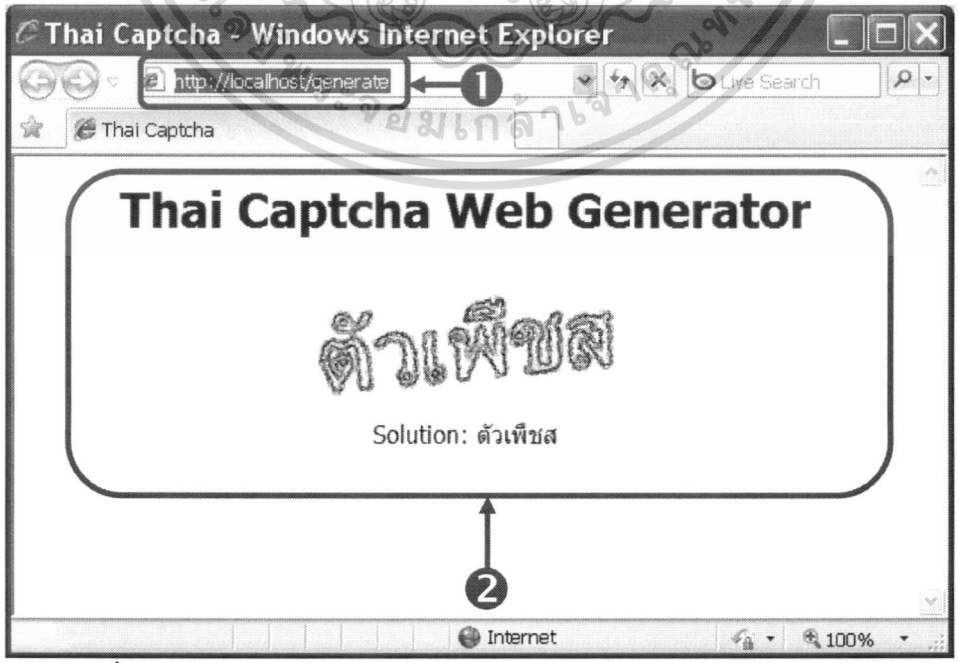
4.3.4 ส่วนสร้าง CAPTCHA



รูปที่ 4.4 ภาพหน้าจอการทำงานของส่วนสร้าง CAPTCHA

เมื่อแบบจำลองเอ็นแกรมถูกสร้างเสร็จแล้ว ระบบจะก็สามารถสร้าง CAPTCHA ภาษาไทยได้ ซึ่งจากรูปที่ 4.4 แสดงภาพหน้าจอการทำงานของส่วนสร้าง CAPTCHA ซึ่งประกอบด้วยส่วนต่างๆ เก้าส่วนดังต่อไปนี้

1. ปุ่มเลือกแท็ปของส่วนสร้าง CAPTCHA ผู้ใช้สามารถเข้าถึงการทำงานในส่วนนี้ได้ด้วยการคลิ๊กที่แท็ปนี้
2. ช่องสำหรับให้ผู้ใช้กรอกที่อยู่ไดเร็กทอรี่เอาท์พุตที่ระบบจะใช้บันทึกไฟล์ CAPTCAHA ที่สร้างขึ้น
3. ปุ่มค้นหาไดเร็กทอรี่เอาท์พุต
4. ช่องสำหรับตั้งชื่อที่ใช้นำหน้าชื่อไฟล์เอาท์พุตโดยมีสามชื่อได้แก่ ชื่อสำหรับเท็กซ์ไฟล์เฉลย, ชื่อสำหรับไฟล์ภาพ CAPTCHA, ชื่อสำหรับไฟล์ภาพ CAPTCHA ที่สร้างแบบออนไลน์ผ่านเว็บ
5. ปุ่มทางด้านซ้ายใช้เลือกรูปแบบของเท็กไฟล์เฉลยจากสองรูปแบบคือ แบบไฟล์เดียวซึ่งเก็บแบบ CSV หรือแบบแยกเก็บเป็นเท็กไฟล์ของแต่ละ CAPTCHA ส่วนปุ่มทางด้านขวาใช้เลือกชนิดของไฟล์ภาพ CAPTCHA ซึ่งเลือกได้จากสี่ชนิด
6. ช่องสำหรับตั้งค่าปริมาณที่ต้องการให้สร้าง CAPTCHA
7. ปุ่มสั่งให้สร้าง CAPTCHA ตามค่าต่างๆ ที่ผู้ใช้ตั้งไว้
8. แถบแสดงความคืบหน้าในการสร้าง CAPTCHA
9. ปุ่มสั่งให้ระบบบันทึกค่าที่ผู้ใช้ตั้งไว้ในขั้นตอนที่ 2 ถึง 6 เก็บไว้ในไฟล์คอนฟิก เพื่อให้สามารถโหลดค่าที่เคยตั้งไว้กลับมาใช้งานได้ทันทีเมื่อเริ่มใช้งานระบบครั้งต่อไป



รูปที่ 4.5 ภาพหน้าจอผลลัพธ์จากการสร้าง CAPTCHA แบบออนไลน์

รูปที่ 4.5 แสดงภาพหน้าจอผลลัพธ์จากการสร้าง CAPTCHA แบบออนไลน์ ซึ่งระบบอื่นภายนอกสามารถสั่งระบบสร้าง CAPTCHA ภาษาไทยให้สร้าง CAPTCHA แบบออนไลน์ผ่านทางโปรโตคอล HTTP (เว็บ) โดยมีรายละเอียดดังนี้

1. ที่อยู่ URL ที่ใช้สั่งสร้าง CAPTCHA แบบออนไลน์ (http://localhost/generate)
2. ผลลัพธ์ที่แสดงผ่านโปรแกรม Internet Explorer

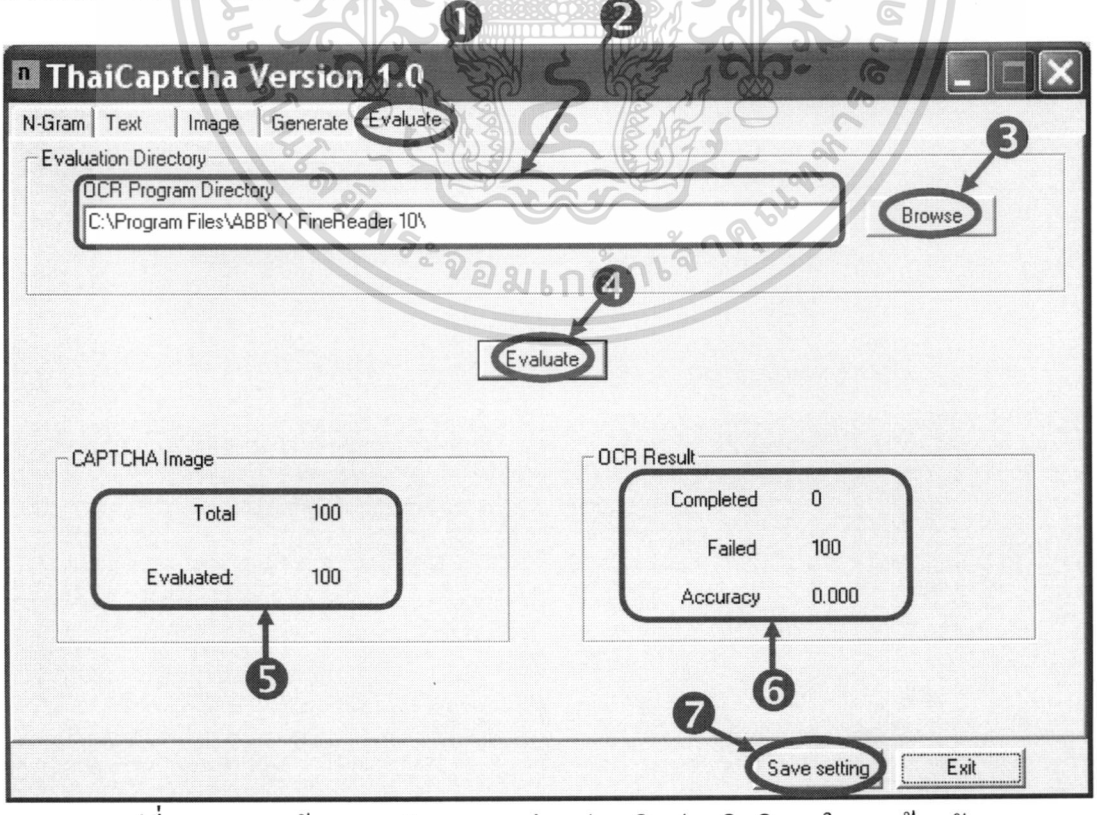
ตัวอย่างซอร์สโค้ด HTML ที่ระบบสร้างขึ้นเพื่อใช้สำหรับส่งรายละเอียดของผลลัพธ์คือ URL ของ CAPTCHA และชุดตัวอักษรเฉลยให้กับผู้ใช้หรือระบบอื่นที่สั่งสร้างผ่านเว็บนั้น มีรายละเอียดแสดงไว้ในรูปที่ 4.6 ดังนี้

```
<html><head><title>Thai Captcha</title></head>
    <body><h1><center>Thai Captcha Web Generator</center></h1>
    <p align=center>
      <img src=captchawww_00001.png><br/>
      Solution: ผ˜วเม˜ผล
    </p>
    </body>
</html>
```

รูปที่ 4.6 ตัวอย่างซอร์สโค้ด HTML จากการสร้าง CAPTCHA ภาษาไทยผ่านเว็บ

4.3.5 ส่วนประเมินประสิทธิภาพในการป้องกัน

ผู้ใช้สามารถทำการประเมินประสิทธิภาพในการป้องกันของ CAPTCHA ที่สร้างขึ้นด้วยโปรแกรม OCR ได้ผ่านทางการทำงานในส่วนนี้

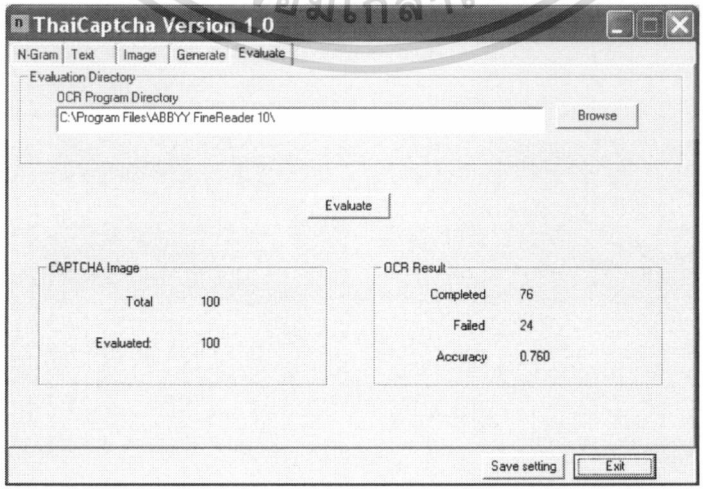


รูปที่ 4.7 ภาพหน้าจอการทำงานของส่วนประเมินประสิทธิภาพในการป้องกัน

จากรูปที่ 4.7 แสดงภาพหน้าจอการทำงานของส่วนประเมินประสิทธิภาพในการป้องกัน ซึ่งประกอบด้วยส่วนต่างๆ 7 ส่วนดังต่อไปนี้

1. ปุ่มเลือกแท็ปของส่วนประเมินประสิทธิภาพในการป้องกัน ผู้ใช้สามารถเข้าถึงการทำงานในส่วนนี้ได้ด้วยการคลิ๊กที่แท็ปนี้
2. ช่องสำหรับให้ผู้ใช้กรอกที่อยู่ไดเร็กทอรี่ของโปรแกรม OCR ที่ระบบรองรับ (ABBY FineReader) เพื่อใช้ทำ OCR ในการประเมิน
3. ปุ่มค้นหาไดเร็กทอรี่ของโปรแกรม OCR
4. ปุ่มสั่งให้ทำการประเมิน (ระบบจะสร้างโปรแซสเพื่อรันโปรแกรม OCR โดยส่งชื่อไฟล์ภาพ CAPTCHA ทั้งหมดที่ระบบสร้างให้โปรแกรม OCR แต่ไม่เกินครั้งละ 200 ไฟล์ให้โปรแกรม OCR แปลงเป็นตัวอักษร แล้วจึงรอรับผลลัพธ์ของการแปลงจากโปรแกรม OCR)
5. ส่วนแสดงจำนวน CAPTCHA ที่ระบบสร้างทั้งหมด และจำนวนที่ทำการประเมินแล้วด้วยโปรแกรม OCR
6. ส่วนแสดงผลลัพธ์จากการแปลงด้วยโปรแกรม OCR เมื่อเทียบกับชุดตัวอักษรเฉลย
7. ปุ่มสั่งให้ระบบบันทึกค่าที่ผู้ใช้ตั้งไว้ในขั้นตอนที่ 2 และ 3 เก็บไว้ในไฟล์คอนฟิก เพื่อให้สามารถโหลดค่าที่เคยตั้งไว้กลับมาใช้งานได้ทันทีเมื่อเริ่มใช้งานระบบครั้งต่อไปโดยผู้ใช้ไม่ต้องตั้งค่าใหม่

ในรูปที่ 4.7 แสดงผลลัพธ์ที่ได้จากค่าที่ตั้งไว้ในการลดความชัดของภาพชุดตัวอักษรตามรูปที่ 4.3 ซึ่งทำให้โปรแกรม OCR ไม่สามารถแปลงได้ถูกต้องเลย นั้นแสดงว่า CAPTCHA ที่สร้างขึ้นด้วยค่าที่ตั้งไว้น่าจะมีประสิทธิภาพในการป้องกันการเจาะผ่านได้ แต่ในกรณีที่ตั้งค่าให้ไม่มีการลดความชัดของภาพเลยก็อาจทำให้โปรแกรม OCR สามารถแปลงได้มากขึ้นดังตัวอย่างในรูปที่ 4.8



รูปที่ 4.8 ภาพหน้าจอการประเมินในกรณีที่ตั้งค่าโดยไม่ทำการลดความชัดของภาพ

บทที่ 5
สรุปผลการพัฒนาระบบ

ในการพัฒนาระบบสร้าง CAPTCHA ภาษาไทยนี้ผู้เขียนได้ศึกษาทฤษฎีต่างๆ เพื่อให้สามารถวิเคราะห์, ออกแบบ, พัฒนา, และทดสอบจนระบบสามารถทำงานได้ตามวัตถุประสงค์ที่ตั้งไว้ กล่าวคือสามารถสร้างภาพชุดตัวอักษร CAPTCHA ภาษาไทยตามที่ผู้ใช้หรือระบบอื่นภายนอกต้องการได้

สำหรับกระบวนการทำงานของระบบจะเริ่มจากการสกัดแบบจำลองไตรแกรมจากคลังข้อความภาษาไทย จากนั้นผู้ใช้ก็จะตั้งค่ารายละเอียดเกี่ยวกับผลลัพธ์ที่ต้องการ แล้วจึงทำการสร้าง CAPTCHA ภาษาไทย ส่วนในการใช้งานนั้นผู้เขียนได้ออกแบบระบบให้สามารถใช้งานได้โดยตรงผ่านทาง GUI ของระบบหรือร้องขอการใช้งานแบบออนไลน์ผ่านโปรโตคอล HTTP ก็ได้ ทั้งนี้ก็เพื่อให้เกิดความสะดวกในการนำไปประยุกต์ใช้งาน

ระบบสร้าง CAPTCHA ภาษาไทยนี้ได้พัฒนาขึ้นให้เหมาะกับผู้ที่มีความคุ้นเคยหรือชำนาญในการใช้งานภาษาไทยบนคอมพิวเตอร์มากกว่าภาษาอื่น โดยได้ประยุกต์ใช้แบบจำลองไตรแกรมในการสร้างชุดตัวอักษรเพื่อให้ระบบสามารถใช้งานได้ง่ายขึ้น และด้วยการใช้ภาษา C++ ในการพัฒนาทำให้ระบบที่พัฒนาขึ้นสามารถทำงานได้อย่างรวดเร็ว

ทั้งหมดนี้ก็เพื่อให้สามารถนำระบบสร้าง CAPTCHA ภาษาไทยไปประยุกต์ใช้ในการป้องกันปัญหาบ็อทหรือสแปมได้จริง และช่วยส่งเสริมการใช้งานภาษาไทยซึ่งเป็นเอกลักษณ์ของชาติให้ปรากฏเด่นชัดยิ่งขึ้น

5.1 อุปสรรค์ในการพัฒนาระบบ

เนื่องจากคลังข้อความของไทยยังมีไม่มากและไม่ค่อยมีการเผยแพร่ต่อสาธารณะ ทำให้ทางเลือกในการเลือกใช้คลังข้อความสำหรับโครงการนี้มีไม่มากเท่าภาษาอังกฤษ ทำให้ไม่สามารถสร้างแบบจำลองเอ็นแกรมสำหรับสร้าง CAPTCHA ภาษาไทยในหมวดหมู่ที่ผู้ใช้ต้องการได้ เช่น ไม่สามารถสร้าง CAPTCHA ภาษาไทยด้วยชุดตัวอักษรที่คล้ายคลึงกับชื่อของคนไทยได้ เนื่องจากไม่มีคลังข้อความชื่อคนไทย

5.2 ข้อเสนอแนะ

1. ระบบสร้าง CAPTCHA ภาษาไทยนั้นสามารถนำไปพัฒนาต่อโดยเพิ่มส่วนแปลงข้อความให้เป็นเสียงพูด (TTS) เพื่ออำนวยความสะดวกให้กับผู้พิการทางสายตา

2. ส่วนที่ใช้ในการสุ่มสร้างชุดตัวอักษรตามความน่าจะเป็นในแบบจำลองเอ็นแกรมนั้น สามารถนำไปพัฒนาเพื่อประยุกต์ใช้ในการสุ่มสร้างชื่อตัวละครต่างๆ ในบทกวี หรือในเกมส์ออนไลน์ ที่ใช้ภาษาไทยได้ โดยอาศัยการสกัดสร้างเอ็นแกรมจากคลังข้อความชื่อคนไทย
3. ใช้วิธีการอื่นในการสุ่มสร้างชุดตัวอักษร CAPTCHA เช่น การทำกระบวนการแก้คำผิดอัตโนมัติแบบย้อนกลับด้วยคำที่สุ่มจากพจนานุกรมเพื่อให้ได้ชุดตัวอักษรที่สุ่มขึ้นโดยการตั้งใจทำให้ผิดเพี้ยนไปจากคำในพจนานุกรม
4. พัฒนาอินเตอร์เฟสในการสร้าง CAPTCHA แบบออนไลน์ให้สามารถทำงานกับโปรโตคอลที่หลากหลายมากขึ้นอาทิเช่น โปรโตคอล SOAP

บรรณานุกรม

เนคเทค. 2553. **คลังข้อความ BEST 2010**. [Online] เข้าถึงได้จาก: http://www.hlt.nectec.or.th/best/.

อัครพล เอกวงศ์อนันต์. 2548. "การระบุคำไทยและคำทับศัพท์ด้วยแบบจำลองเอ็นแกรม". อักษรศาสตรมหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย.

Chew, M., Baird, H. 2003. "Baffle Text: a Human Interactive Proof." pp. 305-316. In **Proceeding of the SPIE/IS&T Document Recognition & Retrieval Conference X.** Santa Clara, CA. Jan. 22-23, 2003.

Devroye, L. 1986. **Non-uniform Random Variate Generation.** [Online]. Available: http://cg.scs.carleton.ca/~luc/rnbookindex.html.

IBM Corp. 2000. **International Components for Unicode Project**. [Online]. Available: http://bugs.icu-project.org/trac/browser/icu/tags/release-2-6/source/test/testdata/th18057.txt.

Koanantakool, H. T., Karoonboonyanan, T., Wutiwiwatchai, C., 2009. "Computers and the Thai Language." **IEEE Annals of the History of Computing.** vol. 31. no. 1. pp. 46-61.

Kruglinski, D. J. 1997. **Inside Visual C++.** Fourth Edition. Redmond, WA. Microsoft Corporation.

Shannon, Claude E. 1984. "A Mathematical Theory of Communication." **Bell System Technical Journal.** Vol. 27. pp. 379-423 and 623-656.

Symantec Corp. 2004. **Symantec Internet Security Threat Report** [Online]. Available: http://www.symantec.com/press/2004/n040920b.html.

Turing, A. 1950. **Computing Machinery and Intelligence**. [Online]. Available: http://mind.oxfordjournals.org/content/LIX/236/433.full.pdf.

ภาคผนวก
อัลกอริทึมที่ใช้ในการพัฒนาระบบ

1. อัลกอริทึมสำหรับนับความถี่ให้กับแบบจำลองเอ็นแกรม

```
declare integer array(256) F1
declare integer array(256, 256) F2
declare integer array(256, 256, 256) F3
Procedure Count(string text)
Begin
  declare integer c0, c1, c2
  for integer i loops 1 to length(text)
    c2 = ascii code of text[i]
    if c2 is not in Thai ascii code then
      next i
    F1[c2] := F1[c2] + 1
    if c1 ≠ 0 then
      F2[c1][c2] := F2[c1][c2] + 1
    if c0 ≠ 0 and c1 ≠ 0 then
      F3[c0][c1][c2] := F3[c0][c1][c2] + 1
    c0 := c1
    c1 := c2
End
```

2. อัลกอริทึมสำหรับสกัดค่าความน่าจะเป็น (P) และความน่าจะเป็นสะสม (C)

```
declare real array(256, 256, 256) P3
declare real array(256, 256, 256) C3
Procedure ExtractTrigram()
Begin
  declare integer array(256, 256) sum

  #sumarize frequency loops
  for integer i loops 1 to 256
    for interger j loops 1 to 256
      sum[i][j] := 0
      for interger k loops 1 to 256
        sum[i][j] := sum[i][j] + F3[i][j][k]

  #calculate proability loops
  for integer i loops 1 to 256
    for interger j loops 1 to 256
      for interger k loops 1 to 256
        P3[i][j][k] := F3[i][j][k] / sum[i][j]

  #calculate cummulative proability loops
  for integer i loops 1 to 256
    for interger j loops 1 to 256
      C3[i][j][0] := P3[i][j][0];
      for interger k loops 1 to 256
        C3[i][j][k] := C3[i][j][k-1] + P3[i][j][k]
end
```

3. อักอริทึมสำหรับสร้างชุดตัวอักษร CAPTCHA

```
procedure string GenerateText(integer length)
begin
  declare string t;
  t[0] := sampling(C1)
  t[1] := sampling(C2[ascii(t[0])])
  for integer i loops 2 to length - 1
    t[i] := sampling(C3[ascii(t[i-2])][ascii(t[i-1])])
  result := t
end
```

```
procedure string sampling(
  real array(256) C: is cdf of ascii code
)
begin
  declare integer x
  declare real u := random(0..1)
  if u = 0 then
    for integer i loops 1 to 256
      if C[i] ≠ 0 then break
    x := i
  elif u < 1 then
    x := absearch(u, C, 256)  # approximate search
  else
    x := 255;
  while x > 0                 # find min x
    if C[x - 1] ≠ C[x] then
      break;
    x := x - 1
  result := char(x)
end
```

4. อักอริทึมสำหรับค้นหาหาอินเด็กซ์ x จากอาร์เรย์ของความน่าจะเป็นสะสม C ที่ตรงกันหรือใกล้เคียง (approximate search)

$$\min\{ x : C[x] \geq u \}$$

ซึ่งสูตรดังกล่าวสามารถนำไปพัฒนาด้วยการนำอัลกอริทึมค้นหาแบบไบนารีมาประยุกต์

ประวัติผู้เขียน

ชื่อ นาย กนกวุธ ถนัดการ
หัวข้อ ระบบสร้าง CAPTCHA ภาษาไทย
สาขาวิชา เทคโนโลยีสารสนเทศ

ประวัติ

ประวัติส่วนตัว เกิดวันที่ 27 ตุลาคม 2519

ประวัติการศึกษา สำเร็จการศึกษาระดับปริญญาตรี วิศวกรรมศาสตร์บัณฑิต (วศ.บ) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ ในปี 2542 และเข้าศึกษาในหลักสูตรวิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง คณะเทคโนโลยีสารสนเทศ ในปี 2551