รหัสโครงการ: **13p33c002**

การจำแนกข้อความประเภทนวนิยายตามตัวละครแบบกึ่งอัตโนมัติ

Semi-automatic Novel Text Classification based on Character

BEST 2011: การแข่งขันสุดยอดซอฟต์แวร์ประมวลผลภาษาไทย

(Thai Language Processing Software Contest)

รายงานฉบับสมบูรณ์

เสนอต่อ

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ

กระทรวงวิทยาศาสตร์และเทคโนโลยี

ได้รับทุนอุดหนุนโรงการวิจัย พัฒนาและวิศวกรรม

โครงการแข่งขันและพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย ครั้งที่ 13

ประจำปีงบประมาณ 2553

โดย

1. **นางสาว ณัฐธิดา เตชะนภารักษ์ นิสิตคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย**
2. **นางสาว สุภรณี กัลยาณกุล นิสิตคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย**
3. **นางสาว อรณี นิลศรีไพรวัลย์ นิสิตคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย**

อาจารย์ที่ปรึกษาโครงการ

ผศ. ดร. อติวงศ์ สุชาโต

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

กิตติกรรมประกาศ

โครงการเรื่องการจำแนกข้อความประเภทนวนิยายตามตัวละครแบบกึ่งอัตโนมัติ (Semi-automatic Novel Text Classification based on Character) คณะผู้พัฒนาขอขอบคุณ ผศ. ดร. อติวงศ์ สุชาโต ที่ให้คำปรึกษาและแนะนำอย่างดีตลอดการพัฒนาโครงการ

ขอขอบพระคุณศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติที่ได้ให้การสนับสนุนทุนอุดหนุนโครงการในการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทยครั้งที่ 13

คณะผู้พัฒนา

บทคัดย่อ

การจำแนกข้อความประเภทนวนิยายตามตัวละครแบบกึ่งอัตโนมัติ คือ การสร้างโปรแกรมจำแนกและจัดกลุ่มข้อความในบทความต่างๆแบบกึ่งอัตโนมัติ (Semi-automatic Text Classification)โดเมนของบทความที่ใช้โครงการนี้ คือ บทความประเภทนิทานหรือนวนิยาย BEST (Benchmark for Enhancing the Standard for Thai language Processing) โปรแกรมจำแนกและจัดกลุ่มตามบุคคลซึ่งเป็นเจ้าของกลุ่มข้อความนั้นๆ เช่น บทบรรยาย หรือ บทพูดของตัวละครแต่ละตัว ผู้ใช้งานจำเป็นต้องระบุกลุ่มของการจำแนก ในที่นี้คือรายชื่อตัวละครที่เป็นเจ้าของบทพูดต่างๆทั้งหมดในนวนิยายเรื่องนั้น จากนั้นโปรแกรมจะวิเคราะห์และประมวลผลตามข้อมูลที่มีการเรียนรู้และจดจำลักษณะของกลุ่มข้อความ รูปแบบของโครงสร้างและบริบทที่เป็นไปได้ของกลุ่มข้อความที่มีบทพูด โดยการประมวลผลการจำแนกนี้ใช้เทคโนโลยีแบบจำลองภาษา(Language Model) เพื่อจำแนกกลุ่มข้อความที่ตามตัวละครในนวนิยายเรื่องหนึ่งๆที่ถูกระบุโดยผู้ใช้ ซึ่งผลลัพธ์ของโปรแกรมผู้ใช้งานจะได้รับบทความที่มีการจำแนกและจัดกลุ่มข้อความทั้งหมดตามตัวละครที่ผู้ใช้กำหนดไว้ โดยจะมีการแทรกสัญลักษณ์บ่งบอกรหัสของตัวละครที่เป็นผู้พูด เพื่อให้สามารถนำไปประยุกต์ใช้ในโปรแกรมอื่นๆต่อไปได้ โปรแกรมนี้จะมีส่วนช่วยเพิ่มความสะดวกให้ผู้ใช้งานมากขึ้น และประหยัดเวลาในการจำแนกและจัดกลุ่มข้อความบทความทั้งหมดด้วยแรงงานคน

ผลที่ได้จากการทดลองประมวลผลโปรแกรมในขณะนี้โปรแกรมสามารถจำแนกหาตัวละครที่เป็นผู้พูดได้เพียงพิจารณาจากรูปแบบประโยคอย่างง่าย เช่น ผู้พูดที่สามารถพิจารณาจากตัวละครไม่กี่ตัวที่อยู่โดยรอบในบริบทของคำพูดนั้นๆ สำหรับผู้พูดที่ต้องพิจารณาจากคำพูดก่อนหน้าและสรรพนามที่บ่งบอกถึงตัวผู้พูดยังอยู่ในข้อยกเว้น ทั้งนี้จากการวิเคราะห์ผลการทดสอบโปรแกรมจะเห็นได้ว่า 74.21% ของจำนวนคำพูดที่ถูกจำแนกได้สำหรับการฝึกฝนด้วยประโยคอย่างง่าย สามารถช่วยในการจำแนกคำพูดตามผู้พูดได้อย่างถูกต้อง

สำหรับการพัฒนาโปรแกรมในอนาคตเราได้เล็งเห็นว่าการใช้สรรพนามสามารช่วยในการจำแนกคำพูดได้ดีขึ้น โดยทางกลุ่มผู้พัฒนาวางแผนที่จะใช้แผนผังตัวละครซึ่งบอกลักษณะของทุกตัวละครให้เรื่องเช่น ฉายา เพศ อายุ ความสัมพันธ์กับตัวละครอื่นๆ โดยอาศัยการป้อนข้อมูลบางส่วนจากผู้ใช้ นอกจากนี้เราจะพัฒนาในส่วนของการจำแนกคำพูดที่ต้องพิจารณาจากคำพูดก่อนหน้าอีกด้วย

Abstract

The semi-automatic Novel Text Classification based on Characters a program which is capable to group and classify a text, semi-automatically. The domain of the text used for training and testing in this project is BEST (Benchmark for Enhancing the Standard for Thai language Processing) text set in a novel category. The program has the ability to separate a narration from a conversation and identify the owner of each speech in a conversation. The possible characters and the criteria, which will be considered in classification, are expected to be initialized by the user. The user is required to identify all characters within the input content. From a training set, the program will learn and memorize the structure of the content in which each speech appears, together with the speech owner. After the program has been trained, the testing novel can be classified. The process of text classification in this project is based on the "Language Model" and "n-gram" techniques. As a consequence, the user will get the classified text with different tags for various characters initialized by the user.

Resultantly, at this moment we can just classify the simple structure of the sentences such as the speech which can be identified its owner by a few characters nearby itself. To identify the owners of the speeches that depend on the previous speeches is still excepted. According to the analysis from the result of our program, we can see that 74.21% of the overall simple sentences which could be classified can be identified by the speeches' owners correctly.

Furthermore, the further improvement of our program is to take the pronoun words into consideration because currently we consider only the characters' names. In order to use pronoun for identifying the speech's owner, we plan to build the relationship chart of each character in each novel by getting some information from the user. Besides, the analysis on the owner of the speeches that rely on the previous speeches is going to be done as well.

คำสำคัญ

แบบจำลองภาษา (Language model)

ซอฟต์แวร์โอเพนซอร์ส (Open Source Software)

โปรแกรมตัดคำภาษาไทย Swath (Smart Word Analysis for THai)

สัญลักษณ์ (Tag)

ชนิดของคำ (Part of Speech)

คลังข้อมูล (database)

1. บทนำ

ในปัจจุบันด้วยความเจริญของอินเทอร์เน็ตและเทคโนโลยี ทำให้ข้อมูลต่างๆมีจำนวนมาก และมีการเติบโตของข้อมูลอยู่ตลอดเวลาเพื่อให้สะดวกในการใช้งานข้อมูล ค้นหา หรือสืบค้นข้อมูล จึงจำเป็นต้องมีการจัดหมวดหมู่ หรือจำแนกกลุ่มของข้อมูล เพื่อให้สามารถสืบค้น หรือเรียกใช้งานข้อมูลสะดวกยิ่งขึ้นแต่การจัดการเพื่อจำแนกกลุ่มต่างๆ ของข้อมูลจำนวนมากนั้น ต้องใช้ทรัพยากรบุคคล เวลา และค่าใช้จ่ายจำนวนมาก เทคโนโลยีทางด้านการประมวลผลข้อความ การจัดการข้อมูล หรือการแบ่งกลุ่มของข้อมูล จึงมีบทบาทสำคัญเพื่อแก้ไขปัญหาการจัดหมวดหมู่ด้วยคน

การจัดกลุ่มของข้อมูลที่ปรากฏในข้อความนั้น ต้องใช้ความรู้ เพื่อตัดสินใจจำแนกกลุ่มต่างๆ ซึ่งในที่นี้การจำแนกกลุ่มที่สนใจได้แก่ การจำแนกกลุ่มของบทสนทนาของแต่ละตัวละครต่างๆ และการจำแนกส่วนการบรรยาย ซึ่งปกติมนุษย์มีความสามารถในการตัดสินใจจำแนกได้จากการอ่านบทความ วิเคราะห์จากบริบทแวดล้อมการสังเกตเครื่องหมายต่างๆ การปรากฏตำแหน่งบทสนทนากับชื่อตัวละครที่อยู่ในเรื่องแล้วจึงใช้ประสบการณ์ เพื่อจำแนกกลุ่มของบริบทนั้น ๆ ดังนั้นโครงการนี้จึงต้องการสร้างระบบเพื่อจำแนกและแบ่งกลุ่มข้อมูลแบบอัตโนมัติ ซึ่งเป็นความท้าทายของโครงงานนี้ที่ต้องสร้างให้คอมพิวเตอร์ให้สามารถตัดสินใจจำแนกกลุ่มต่างๆได้อัตโนมัติ แทนการตัดสินใจด้วยคน ซึ่งชุดข้อความที่ใช้ในการพัฒนาและทดสอบระบบ คือ ชุดข้อความประเภทนิยายทั่วไปและชุดข้อความ BEST[1] ซึ่งจัดทำโดยหน่วยปฏิบัติการวิจัยวิทยาการมนุษยภาษา (Human Language Technology Laboratory) ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ เป็นคลังข้อความภาษาไทยที่มีการกำกับขอบเขตของคำ ขนาด 5 ล้านคำ ประกอบด้วย 4 หมวดคือ บทความวิชาการ (Article), สารานุกรม (Encyclopedia), ข่าว (News), และ นวนิยาย (Novel) แต่ในโครงการนี้เลือกใช้ หมวดนวนิยาย เท่านั้นเพื่อใช้ในการพัฒนาและทดสอบการจำแนกกลุ่มของโปรแกรม

องก์ประกอบของข้อความที่เป็นหมวดนวนิยายนี้ ประกอบด้วย ตัวละคร ซึ่งรวมตั้งแต่ คน สัตว์ หรืออื่น ๆ ที่ถูกสร้างขึ้นในเรื่อง และบทการบรรยายลักษณะต่างๆ เช่น บรรยายลักษณะตัวละคร ความคิด คำพูด การกระทำ ฉาก เวลา และสภาพแวดล้อมเป็นต้น จากลักษณะของข้อความหมวดนวนิยาย ที่มีชื่อเฉพาะของตัวละครปรากฏในการบรรยายเรื่อง ดังนั้นในโครงงานนี้จึงต้องการสกัดค่าลักษณะสำคัญดังกล่าว เพื่อใช้เป็นค่าบ่งบอกความแตกต่างของการแบ่งบทสนทนาของแต่ละตัวละคร เพื่อใช้จำแนกกลุ่มของข้อความต่อไปโครงงานนี้จึงใช้สมบัติที่ของลักษณะข้อมูล BEST นี้มีการกำกับข้อมูล (markup) ใน

ลักษณะต่างๆตามมาตรฐานสากลเพื่อความสะดวกของการสืบค้นทางคอมพิวเตอร์ เช่น ข้อความหมวดนวนิยาย มีการกำกับของชื่อเฉพาะ <NE>เพื่อบอกตำแหน่งของชื่อเฉพาะของตัวละครในข้อความ ซึ่งเป็นค่าลักษณะสำคัญค่าหนึ่งซึ่งนำมาใช้เพื่อบ่งบอกถึง เอกลักษณ์เฉพาะของกลุ่มข้อความที่ปรากฏในเอกสาร

นอกจากนี้เพื่อความสามารถของเทคโนโลยีทางด้านการสังเคราะห์เสียงพูด ซึ่งโปรแกรมสังเคราะห์เสียงพูดที่มีอยู่ในปัจจุบันสามารถแปลงจากข้อความภาษาไทยให้เป็นข้อมูลประเภทเสียงได้ถูกต้องอย่างมีประสิทธิภาพ แต่โปรแกรมดังกล่าวมีข้อจำกัดคือข้อความที่รับเข้ามาจะถูกสังเคราะห์เสียงออกมาเป็นเสียงอ่านรูปแบบของสียงแบบเดียวตลอดทั้งเรื่องเสมือนมีผู้อ่านคนเดียวกำลังอ่านข้อความนั้นอยู่ซึ่งเมื่อนำโปรแกรมนี้มาใช้ในการแปลงเสียงของข้อความบางประเภทเช่นบทสนทนาในนิทานหรือข้อความนวนิยายที่มีการโต้ตอบกันระหว่างตัวละครจะพบปัญหาว่าผู้ฟังจะไม่สามารถแยกแยะความแตกต่างของเสียงที่ถูกสังเคราะห์ออกมาของแต่ละตัวละครได้ซึ่งจะส่งผลให้ผู้ฟังขาดอรรถรสจากการรับฟังเสียงที่ถูกสังเคราะห์ขึ้น ซึ่งข้อจำกัดดังกล่าวเกิดขึ้นเนื่องจากโปรแกรมไม่สามารถจำแนกความแตกต่างของข้อความที่เกิดจากตัวละครที่แตกต่างกันด้วยเหตุนี้ผู้จัดทำโครงงานนี้จึงต้องการเพิ่มความสามารถให้โปรแกรมสังเคราะห์เสียงพูด ทำหน้าที่ได้ลักษณะเหมือนกับการอ่านข้อความของคนเล่าเรื่องได้นั้น และเพื่อเพิ่มความบันเทิงให้แก่ผู้ฟังโปรแกรมสังเคราะห์เสียงพูด ให้มีการออกเสียงของตัวละครในเรื่อง หรือบรรยายต่างๆแตกต่างกันได้ จึงเสนอการพัฒนาส่วนโปรแกรมที่จำแนกกลุ่มข้อความตามประเภทนั่นคือการแยกส่วนที่เป็นบทสนทนาออกจากส่วนที่เป็นคำบรรยายหรือคำพรรณนารวมทั้งสามารถแยกบทสนทนาของแต่ละตัวละครออกจากกันอีกด้วยเพื่อให้การสังเคราะห์เสียงออกมาในรูปแบบที่แตกต่างกันตามประเภทของกลุ่มข้อความที่ได้แบ่งไว้ซึ่งโปรแกรมที่ถูกต่อยอดดังกล่าวจะช่วยให้เสียงที่สังเคราะห์ออกมามีความเป็นธรรมชาติมากยิ่งขึ้นทั้งนี้การวิเคราะห์เพื่อจัดกลุ่มข้อความตามประเภทต่างๆสามารถนำไปประยุกต์ใช้กับโปรแกรมประมวลผลข้อความภาษาไทยอื่นๆได้อีกด้วย

สารบัญ

2. วัตถุประสงค์และเป้าหมาย

1.1. เพื่อพัฒนาซอฟต์แวร์ต้นแบบที่สามารถใช้จำแนกหมวดหมู่ของข้อความหมวดนวนิยาย

1.2. เพื่อพัฒนาโปรแกรมการจัดกลุ่มของข้อมูล ส่วนบทสนทนาของแต่ละตัวละครต่างๆในเรื่อง ส่วนคำบรรยายหรือคำพรรณนาในบทความประเภทนวนิยายได้

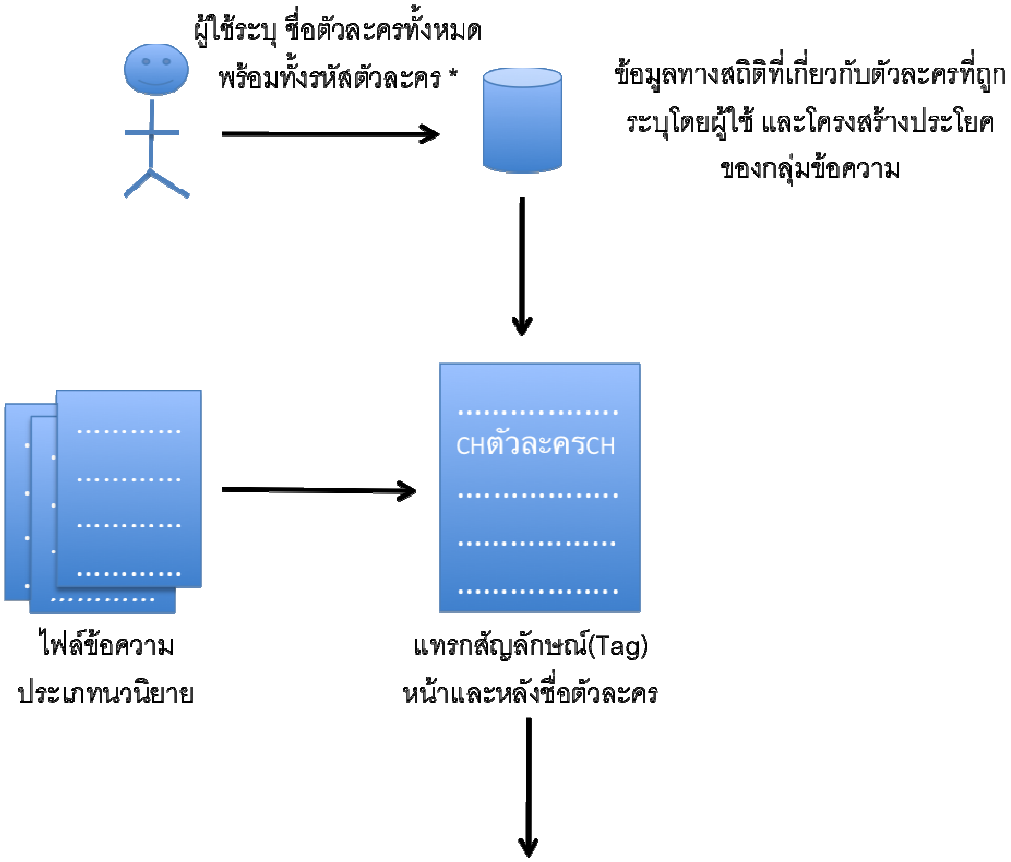
1.3. เพื่อพัฒนาโปรแกรม ให้วิเคราะห์รูปแบบของกลุ่มข้อความ และสามารถจำแนกกลุ่มข้อความออกเป็นประเภทตามความแตกต่างของรูปแบบที่กำหนดได้

1.4. เพื่อวิเคราะห์ การจัดกลุ่มข้อความตามประเภทต่างๆ ให้สามารถนำไปประยุกต์ใช้กับโปรแกรมประมวลผลข้อความภาษาไทยอื่นๆได้

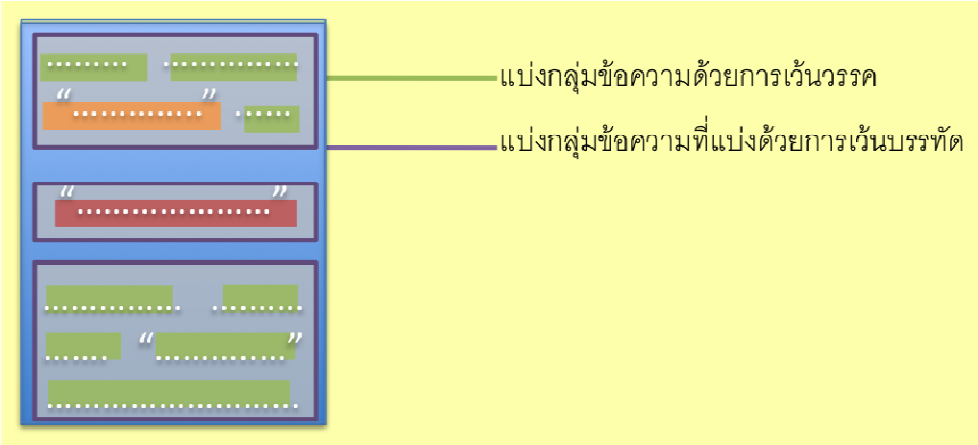
1.5. เพื่อส่งเสริมการพัฒนาและการใช้ซอฟต์แวร์เกี่ยวกับเทคโนโลยีการประมวลผลข้อความสำหรับข้อความภาษาไทย

3. รายละเอียดของการพัฒนา

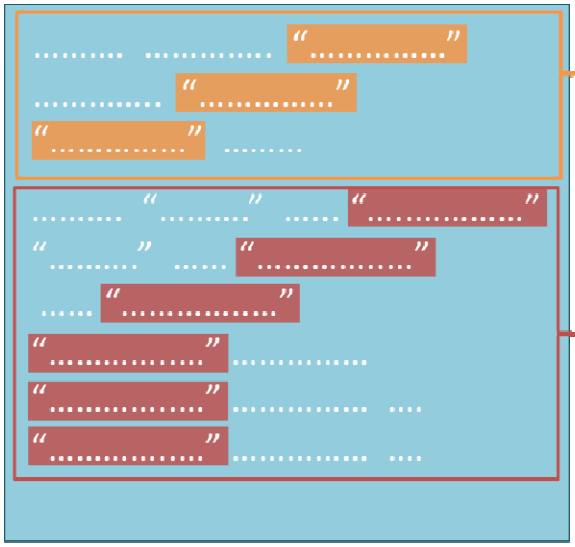
3.1 ขั้นตอนการทำงานของโปรแกรม



จัดแบ่งข้อความออกเป็นกลุ่ม



สร้างชุดของลำดับกลุ่มข้อความที่อยู่ก่อนคำพูด และหลังคำพูด*



ชุดของลำดับกลุ่มข้อความ ชุดที่ 1
(สำหรับคำพูดที่ 1)

ชุดของลำดับกลุ่มข้อความ ชุดที่ 2
(สำหรับคำพูดที่ 2)

* คำพูดที่พิจารณาของลำดับกลุ่มข้อความในแต่ละชุด เป็นคำพูดเดียวกัน

แปลงแต่ละชุดของลำดับกลุ่มข้อความเป็นโครงสร้างประโยค

ที่ประกอบด้วย Part of Speech* (ยกเว้น <UCHAR>)

...
<NCMN>+<RPRE>+<CHAR>+<UVERB>+<JSBR>+<CHAR>+<UVERB>+<VSTA>+<space>+<speech>
<speech>+<space>+<NCMN>+<RPRE>+<CHAR>+<UVERB>+<JSBR>
<speech>+<space>+<NCMN>+<RPRE>+<CHAR>+<UVERB>+<JSBR>+<UVERB>+<VSTA>
...
...
<DDBQ>+<VACT>+<NCMN>+<NEG>+<ADVN>+<DCNM>+<para>+<speech>
<CHAR><NCMN>+<VATT>+<RPRE>+<NPRP>+<space>+<DDBQ>+<NEG>+<ADVN>+<DCNM>+<para>
+<speech>
...

หาความน่าจะเป็นของโครงสร้างประโยคของแต่ละชุดลำดับกลุ่มข้อความ

ที่ได้กำหนดให้<CHAR> ตัวหนึ่งเป็น <UCHAR> (ผู้พูด)

1 ลำดับกลุ่มข้อความ จากชุดที่ 1 ⇒Probability = $6.68*10^{-14}$

<NCMN>+<RPRE>+<UCHAR>+<UVERB>+<JSBR>+<CHAR>+<UVERB>+<VSTA>+<space>+<speech>
<NCMN>+<RPRE>+<CHAR>+<UVERB>+<JSBR>+<UCHAR>+<UVERB>+<VSTA>+<space>+<speech>

⇒Probability = $2.29*10^{-14}$

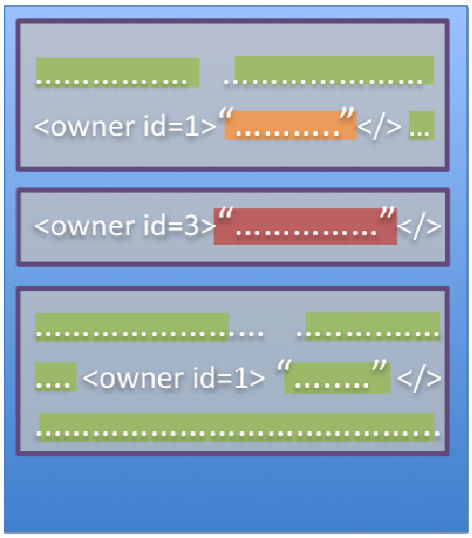
ผลของการเป็นตัวละครที่เป็นผู้พูด(ตำแหน่งที่มี Tag เป็น <UCHAR>)

ของแต่ละชุดลำดับกลุ่มข้อความโดยพิจารณาจากความน่าจะเป็นที่มีค่ามากที่สุด

ผลลัพธ์ คือ ชุดข้อความที่ได้รับการระบุประเภทเรียบร้อยแล้ว

โดยทำการแทรกสัญลักษณ์(Tag) หน้าและหลัง เครื่องหมายคำพูดและ

ระบุรหัสตัวละครที่เป็นผู้พูดประโยคในเครื่องหมายคำพูดนั้น



3.2 ทฤษฎีที่เกี่ยวข้อง

- 3.2.1 แบบจำลองภาษา [2][3]

แบบจำลองภาษาเป็นแบบจำลองที่บ่งบอกความน่าจะเป็นของโครงสร้างประโยคตามหลักหรือกฎเกณฑ์ที่ผู้นำไปใช้สร้างขึ้นหรือข้อมูลทางสถิติที่มีการรวบรวมไว้แบบจำลองภาษาได้ถูกนำมาใช้ในการวิเคราะห์ความเป็นไปได้สำหรับโครงสร้างประโยคแบบหนึ่งๆทั้งนี้การวิเคราะห์ทางด้านภาษาส่วนใหญ่ต้องอาศัยความรู้เกี่ยวกับแบบจำลองภาษาซึ่งแบบจำลองดังกล่าวถูกนำไปใช้ในหลายจุดประสงค์อาทิเช่นการตัดแบ่งข้อความย่อยจากข้อความทั้งหมดความน่าจะเป็นที่ได้มีส่วนช่วยในการตัดสินใจว่าข้อความย่อยแต่ละส่วนควรเริ่มต้นและสิ้นสุดลงที่ไหนโดยยังคงความหมายที่ถูกต้องเมื่อรวมกับข้อความทั้งหมดโปรแกรมประมวลผลภาษาธรรมชาติที่แบบจำลองภาษามาใช้อาทิการแปลภาษาการแบ่งส่วนของข้อความเป็นต้น

ดังที่กล่าวไว้ข้างต้นแบบจำลองภาษาจะวิเคราะห์ความน่าจะเป็นของโครงสร้างประโยคได้สองแบบคือแบบแรก: การตั้งกฎเกณฑ์ตามการใช้งานของผู้ใช้ซึ่งผู้ใช้ต้องกำหนดกฎไว้ว่าคำใดตามด้วยคำใดบ้างอีกแบบคือ N-Gram ซึ่ง N-Gramเป็นวิธีที่ใช้ค่าทางสถิติของคำหรืออักขระที่มาเรียงต่อกันจำนวน N ตัวและใช้หลักการของความน่าจะเป็นในการวิเคราะห์ความเป็นไปได้ดังนี้

กำหนดให้ w_i เป็นคำใดๆ

$w_1,\ldots, w_m$เป็นประโยคที่สร้างจากคำมาเรียงต่อกัน

$$P(w_1,\ldots,w_m) = \prod_{i=1}^{m} P(w_i|w_1,\ldots,w_{i-1}) \approx \prod_{i=1}^{m} P(w_i|w_{i-(n-1)},\ldots,w_{i-1})$$

ดังนั้นจะได้ $P(w_1,\ldots, w_m)$ ออกมาเป็นความน่าจะเป็นที่จะเกิดประโยคจากคำว่า $w_1,\ldots, w_m$เรียงต่อกันทั้งนี้เราสามารถนำหลักการของ N-Gram ไปใช้ในรูปแบบอื่นได้ที่นอกเหนือจากอักขระและคำเช่นรูปแบบประโยคที่เรียงต่อกันของตัวละครต่างๆที่เกิดขึ้นซ้ำๆเพื่อใช้ในการแยกแยะความแตกต่างระหว่างบทพูดของตัวละครแต่ละตัว

สำหรับการนำมาประยุกต์ใช้ในโปรแกรม Semi-automatic novel text classification based on Character นี้ ได้นำหลักการของแบบจำลองภาษาหรือLanguage model ร่วมกับ N-gram โดยในที่นี้เราได้ใช้แบบ 3-gram หลักการดังกล่าวนำมาประยุกต์ใช้ในการหาความน่าจะ

เป็นของโครงสร้างประโยคของแต่ละชุดลำดับกลุ่มข้อความ เพื่อหาความน่าจะเป็นที่มากที่สุดที่เมื่อกำหนดให้ตัวละครแต่ละตัวในลำดับกลุ่มข้อความหนึ่งเป็นผู้พูด ตำแหน่งของตัวละครในลำดับกลุ่มข้อความที่ให้ผลลัพท์เป็นค่าความน่าจะเป็นที่มากที่สุดตามหลักการ N-gram จะถือว่ามีโอกาสมากที่สุดที่จะเป็นผู้พูดของคำพูดที่กำลังพิจารณาอยู่ในชุดลำดับกลุ่มข้อความนั้นๆ

โดยกำหนดให้ w_i เป็นชนิดของคำใดๆ เช่น <CHAR>

$w_1,\ldots, w_m$เป็นโครงสร้างประโยคที่สร้างจากชนิดของคำมาเรียงต่อกัน

ตัวอย่างเช่น

ตัวอย่างโครงสร้างของลำดับชุดข้อความหนึ่ง ปรากฏตัวละครสองตัว (สังเกตได้จากชนิดของคำประเภท <CHAR>)

<NCMN>+<CHAR>+<UVERB>+<JSBR>+<CHAR>+<UVERB>+<VSTA>+<space>+<speech>

เราต้องการหาว่าตัวละคร หรือ <CHAR> ตำแหน่งใดที่มีโอกาสเป็นผู้พูดสำหรับคำพูดหรือ <speech> นั้น โดยหาจากความน่าจะเป็นที่มากที่สุด เมื่อกำหนดให้แต่ละ <CHAR> เป็นผู้พูด หรือ <UCHAR>

- <UCHAR> แทน <CHAR> ตำแหน่งที่ 1

<NCMN>+<UCHAR>+<UVERB>+<JSBR>+<CHAR>+<UVERB>+<VSTA>+<space>+<speech>

มีความน่าจะเป็นเท่ากับ

P(<NCMN>,<UCHAR>,<UVERB>,<JSBR>,<CHAR>,<UVERB>,<VSTA>,<space>,<speech>)

- <UCHAR> แทน <CHAR> ตำแหน่งที่ 2

<NCMN>+<CHAR>+<UVERB>+<JSBR>+<UCHAR>+<UVERB>+<VSTA>+<space>+<speech>

มีความน่าจะเป็นเท่ากับ

P(<NCMN>,<CHAR>,<UVERB>,<JSBR>,<UCHAR>,<UVERB>,<VSTA>,<space>,<speech>)

(ความน่าจะเป็นดังกล่าวหามาจากคลังข้อมูล (database) ที่เกิดจากการฝึกฝน โดยคลังข้อมูลจะเก็บโครงสร้างประโยคที่ปรากฏในชุดบทความประเภทนวนิยายชุดฝึกฝน)

3.3 เครื่องมือที่ใช้ในการพัฒนา

- ภาษาที่ใช้เขียน : Java
- Tools อื่นๆ : Swath , MySQL

3.4 รายละเอียดโปรแกรมที่ได้พัฒนาในเชิงเทคนิค (Software Specification)

- Input/ Output Specification

  Input Specification คือ ชุดข้อความที่ใช้ในการพัฒนาและทดสอบระบบ คือ ชุดข้อความ BEST หมวดนวนิยาย และการระบุตัวละครโดยผู้ใช้

  Output Specification คือ ชุดข้อความที่ถูกจำแนกประเภทของกลุ่มข้อความด้วยสัญลักษณ์(Tag)ปิดหน้าและหลังของแต่ละคำพูดในนวนิยาย ซึ่งแท็กจะบ่งบอกรหัสของตัวละครที่เป็นผู้พูดคำพูดนั้นๆ ทั้งนี้รหัสของตัวละครได้ถูกระบุไว้โดยผู้ใช้ในเบื้องต้นและเก็บไว้ในฐานข้อมูล ตัวอย่างเช่น

  <owner id=…>”คำพูด”</>

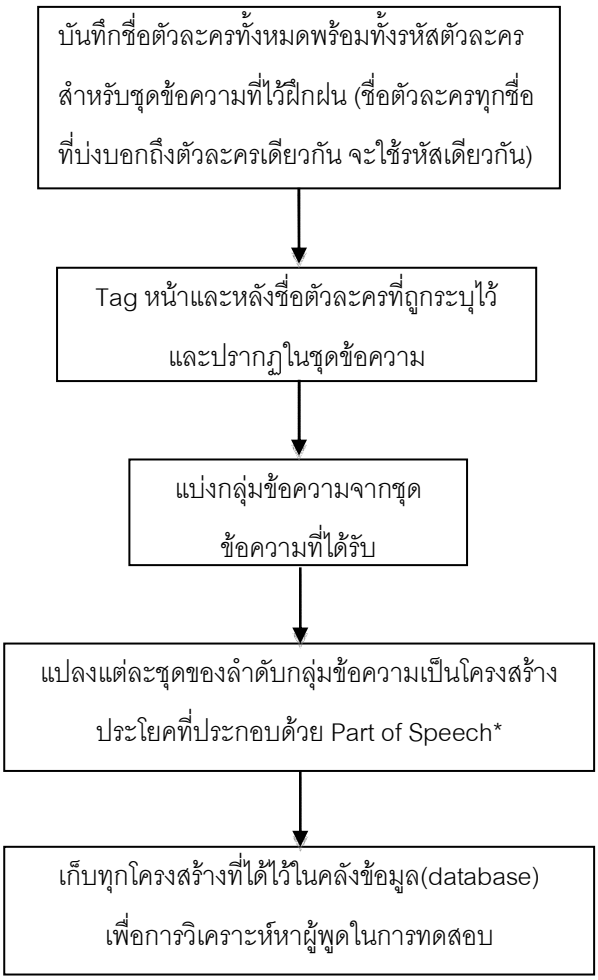
- Functional Specification

1. สามารถวิเคราะห์รูปแบบของกลุ่มข้อความและจำแนกกลุ่มข้อความตามผู้พูดของข้อความในเครื่องหมายคำพูด โดยผู้พูดจะเป็นตัวละครตามที่ผู้ใช้กำหนดไว้เบื้องต้นได้

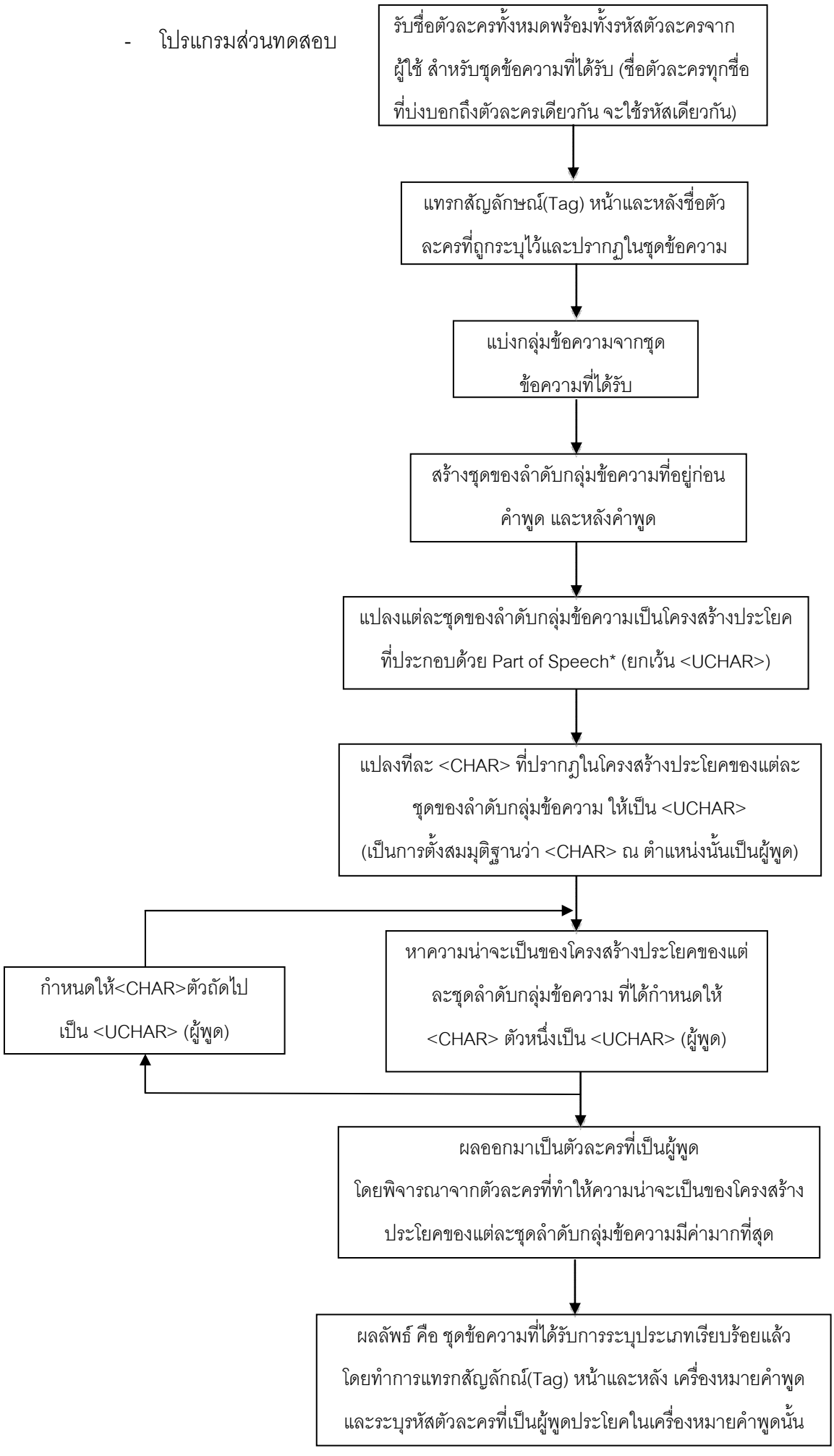
2. สามารถนำโปรแกรมไปประยุกต์ใช้กับโปรแกรมประมวลผลข้อความภาษาไทยอื่นๆได้

- โครงสร้างซอฟต์แวร์ (Design)

  - โปรแกรมส่วนฝึกฝน



- โปรแกรมส่วนทดสอบ



* Part of Speech ที่นำมาใช้ในการสร้างโครงสร้างประโยคของกลุ่มข้อความมีดังนี้

- <speech> สำหรับข้อความที่อยู่ในเครื่องหมายคำพูด
- Part of Speech โดยใช้Swath
- <CHAR> สำหรับตัวละครที่ระบุโดยผู้ใช้
- <UCHAR> สำหรับตัวละครที่เป็นผู้พูด
- <UVERB> สำหรับคำกริยาที่บ่งบอกถึงการแสดงออกทางคำพูด ความคิด หรือ ความรู้สึกที่มักจะถูกเขียนอยู่ในเครื่องหมายคำพูด
- <UNOUN> สำหรับคำนามที่บ่งบอกถึงการแสดงออกทางคำพูด ความคิด หรือ ความรู้สึกที่มักจะถูกเขียนอยู่ในเครื่องหมายคำพูด
- <space> สำหรับการแบ่งระหว่างแต่ละกลุ่มข้อความในย่อหน้าเดียวกัน
- <para> สำหรับการแบ่งระหว่างชุดของกลุ่มข้อความแต่ละย่อหน้า

- ส่วนสำคัญที่พัฒนาขึ้นเอง

โปรแกรมในส่วนฝึกฝนและส่วนทดสอบ ตั้งแต่การแบ่งกลุ่มข้อความจากชุดข้อความ การเก็บโครงสร้างในคลังข้อมูล (database) เพื่อนำไปวิเคราะห์ในส่วนทดสอบ การสร้างชุดของลำดับกลุ่มข้อความที่อยู่ก่อนคำพูดและหลังคำพูด การหาความน่าจะเป็นของโครงสร้างประโยคของแต่ละชุดลำดับกลุ่มข้อความ รวมไปถึงการวิเคราะห์ผลลัพธ์หรือชุดข้อความที่ได้รับการระบุประเภท (ระบุรหัสตัวละครที่เปนผู้พูดประโยคในแต่ละเครื่องหมายคำพูด) เป็นโปรแกรมส่วนที่ทำการพัฒนาขึ้นเองทั้งสิ้น แต่สำหรับส่วนของการแปลงโครงสร้างประโยคที่ประกอบด้วย Part of Speech* มีการประยุกต์ใช้โปรแกรม Swath ประกอบกับการกำหนด Part of speech บางส่วนเอง ดังที่จะกล่าวในหัวข้อถัดไป

- ส่วนที่นำมาประกอบในโปรแกรม (แหล่งที่มา) [4]

ในส่วนของการสร้างโครงสร้างประโยคของกลุ่มข้อความซึ่งประกอบด้วย Part of speech* ส่วนหนึ่งผู้พัฒนาได้นำโปรแกรม Swath มาประกอบในการพัฒนา โปรแกรม Swath (Smart Word Analysis for THai) เป็นซอฟต์แวร์โอเพนซอร์ส (Open Source Software) ของหน่วยปฏิบัติการวิจัยวิทยาการมนุษยภาษา (HLT) เนคเทค Swath เป็นโปรแกรมตัดคำภาษาไทยที่สามารถระบุสัญลักษณ์ (Tag) ตามชนิดของคำ (Part of speech) ซึ่งตรงตามความต้องการในการนำไปประยุกต์ใช้กับโปรแกรมของผู้พัฒนาได้อย่างดี

หลังจากได้ผลลัพท์จากการตัดคำและระบุสัญลักษณ์(Tag) ตามชนิดของคำ (Part of speech) ของกลุ่มข้อความโดยการประมวลผลจาก Swath สัญลักษณ์ของกลุ่มคำหนึ่งๆจะถูกนำมาเรียงต่อกันเป็นโครงสร้างประโยค และเราได้มีการเปลี่ยน Part of speech สำหรับคำหรือกลุ่มบางประเภทและมีการแทรกสัญลักษณ์ (Tag) เพื่อการประมวลผลที่มีประสิทธิภาพมากขึ้น ดังนี้ <speech>, <CHAR>, <UCHAR>, <UVERB>, <UNOUN>, <space>, <para> ตามความหมายที่ได้ระบุไว้ข้างต้น

3.5 ขอบเขตและข้อจำกัดของโปรแกรมที่พัฒนา

1. โปรแกรมที่พัฒนาขึ้นมานั้นจะมีวัตถุประสงค์เพื่อการจำแนกข้อความแบบกึ่งอัตโนมัติจากบทความประเภทนวนิยายเท่านั้น
2. โปรแกรมจะจำแนกข้อความออกเป็น 2 ประเภทคือ
   - คำบรรยายหรือพรรณนา
   - คำพูดของตัวละครใดๆซึ่งถูกจำแนกโดยเครื่องหมายอัญประกาศเท่านั้น (“)
3. ผู้ใช้จำเป็นต้องระบุชื่อตัวละครทั้งหมดในนวนิยายเรื่องนั้นๆโปรแกรมจึงจะสามารถประมวลผลต่อไปได้
4. โปรแกรมพัฒนาเพื่อใช้ประเมินผลลัพธ์จากการจำแนกประเภทของข้อความในบทความ
5. โปรแกรมในขณะนี้สามารถจำแนกหาตัวละครที่เป็นผู้พูดได้เพียงพิจารณาจากรูปแบบประโยคอย่างง่าย เช่น ผู้พูดที่สามารถพิจารณาจากตัวละครไม่กี่ตัวที่อยู่โดยรอบในบริบทของคำพูดนั้นๆ สำหรับผู้พูดที่ต้องพิจารณาจากคำพูดก่อนหน้าและสรรพนามที่บ่งบอกถึงตัวผู้พูดยังอยู่ในข้อยกเว้น

3.6 คำศัพท์เฉพาะ [5]

ชนิดของคำ(Part of speech) ที่ใช้วิเคราะห์ในการจำแนกประเภทประกอบด้วย

<para> : การขึ้นบรรทัดใหม่

<space> : การเว้นวรรค

<speech>: คำพูดใดๆ

<CHAR> : ตัวละครใดๆ

<UCHAR>: ตัวละครซึ่งเป็นเจ้าของคำพูดในประโยคนั้นๆ

<UVERB> : คำกริยาที่บ่งบอกถึงการแสดงออกทางคำพูด ความคิด หรือ ความรู้สึกที่มักจะถูกเขียนอยู่ในเครื่องหมายคำพูด

<UNOUN>: คำนามที่บ่งบอกถึงการแสดงออกทางคำพูด ความคิด หรือ ความรู้สึกที่มักจะถูกเขียนอยู่ในเครื่องหมายคำพูด

<NPRP> : คำนามเฉพาะเช่นวินโดว์โคโรน่า

<NCNM> : จำนวนใดๆเช่น 1 2 หนึ่งสอง

<NONM> : ลำดับที่เช่นที่หนึ่งที่สองที่ 1 ที่ 2

<NLBL> : คำนามใช้บอกสิ่งของเช่น 1 2 a b ก ข

<NCMN> : คำนามทั่วไปเช่นหนังสือทีวี

<NTTL> : คำขึ้นต้นชื่อบุคคลเช่นพลเอก

<PPRS> : คำสรรพนามเรียกบุคคลเช่นเขาเธอฉัน

<PDMN> : คำบ่งชี้เช่นนี่นั่นโน่น

<PNTR> : สรรพนามที่ใช้ในคำถามเช่นใครที่ไหน

<PREL> : ประพันธสรรนามเช่นที่ซึ่งอัน

<VACT> : กริยาการกระทำเช่นเดินร้องเพลงกิน

<VSTA> : กริยาบอกสถานะเช่นเห็นคือรู้

<VATT> : กริยาบอกลักษณะเช่นสวยอ้วน

<XVBM> : กริยานุเคราะห์ก่อนคำว่า “ไม่” เช่นเกิดเกือบกำลัง

<XVAM> : กริยานุเคราะห์หลังคำว่า “ไม่” เช่นค่อยน่าได้

<XVMM> : กริยาก่อนหรือหลังคำว่า “ไม่” เช่นควรเคยต้อง

<XVBB> : กริยาเชิงคำสั่งก่อนกริยานุเคราะห์เช่นกรุณาจงห้าม

<XVAE> : กริยาหลังกริยานุเคราะห์เช่นไปมาขึ้น

<DDAN> : คำบ่งชี้เฉพาะใช้หลังนามเช่นนี่นั่นทั้งหมด

<DDAC> : คำบ่งชี้เฉพาะใช้หลังนามและคำบ่งชี้อื่นๆเช่นนี้นั้นโน้น

<DDBC> : คำบ่งชี้เฉพาะระหว่างนามและคำบ่งชี้เช่นทั้งอีกเพียง

<DDAQ> : คำบ่งชี้เฉพาะหลังคำบอกปริมาณเช่นถ้วนพอดี

<DIAC> : คำบ่งชี้ใช้หลังคำนามและอาจจะมีคำบ่งชี้อื่นๆคั่นเช่นไหนอื่นต่างๆ

<DIBQ> : คำบ่งชี้ระหว่างคำนามและคำบ่งชี้อื่นๆหรือคำบอกปริมาณใดๆเช่นบาง

<DIAQ> : คำบ่งชี้หลังคำบอกปริมาณเช่นกว่าเศษ

<DCNM> : คำบ่งชี้บอกจำนวนเช่นหนึ่งคนเสือ 2 ตัว

<DONM> : คำบ่งชี้บอกลำดับที่เช่นที่หนึ่งที่สองที่สุดท้าย

<ADVN> : คำวิเศษณ์ทั่วไปเช่นเก่งเร็วช้าสม่ำเสมอ

<ADVI> : คำวิเศษณ์ที่ประกอบด้วยคำซ้ำเช่นเร็วๆเสมอๆช้าๆ

<ADVP> : คำวิเศษณ์ที่ประกอบด้วยคำนำหน้าเช่นโดยเร็ว

<ADVS> : คำวิเศษณ์บอกทัศนคติของผู้พูดเช่นโดยปกติธรรมดา

<CNIT> : ลักษณนามบอกจำนวนเช่นตัวคนเล่ม

<CLTV> : ลักษณนามบอกจำนวนกลุ่มเช่นคู่ฝูงกลุ่ม

<CMTR> : หน่วยของการวัดเช่นกิโลเมตรแก้วชั่วโมง

<CFQC> : ลักษณนามบอกความถี่เช่นครั้งเที่ยว

<CVBL> : ลักษณนามในคำพูดเช่นม้วนมัด

<JCRG> : คำสันธานเชื่อมประโยคเช่นและหรือแต่

<JCMP> : คำสันธานในการเปรียบเทียบเช่นกว่าเหมือนกับเท่ากับ

<JSBR> : คำสันธานบอกส่วนขยายเช่นเพราะว่าเนื่องจากแม้ว่า

<RPRE> : คำบุพบทเช่นจากละของใต้

<INT> : คำอุทานเช่นโอ๊ยโอ้อ๋อ

<FIXN> : คำนำหน้าทั่วไปเช่นความสนุกสนานการทำงาน

<FIXV> : คำนำหน้าที่เกี่ยวกับกริยาวิเศษณ์เช่นอย่างเร็ว

<EAFF> : คำตอบรับเช่นจ๊ะค่ะครับ

<EITT> : คำลงท้ายคำถามเช่นหรือเหรอไหมมั้ย

<NEG> : คำบ่งความปฏิเสธเช่นไม่มิได้

<PUNC> : เครื่องหมายต่างๆเช่น , “

3.7 คุณลักษณะของอุปกรณ์ที่ใช้กับโปรแกรม

เนื่องจากโปรแกรมได้นำโปรแกรม Swath ซึ่งพัฒนาจากบุคคลภายนอกมาประยุกต์ใช้ ซึ่งโปรแกรมดังกล่างจำเป็นต้องประมวลผลบนระบบปฏิบัติการวินโดว์เท่านั้น ทำให้โปรแกรมที่ทางกลุ่มพัฒนามีความจำเป็นต้องประมวลผลบนวินโดว์เท่านั้น

4. กลุ่มผู้ใช้โปรแกรม

- กลุ่มผู้ใช้ที่มีความสนใจในโปรแกรมประมวลผล จัดกลุ่มและจำแนกกลุ่มข้อความภาษาไทย เพื่อนำไปศึกษา และประยุกต์ใช้กับโปรแกรมประมวลผลข้อความภาษาไทยอื่นๆ โดยเฉพาะกลุ่มข้อความประเภทนวนิยาย หรือประเภทอื่นๆที่มีรูปแบบเฉพาะตัว
- กลุ่มผู้ใช้ที่ต้องการใช้ประโยชน์จากโปรแกรมนี้โดยตรง เพื่อจัดกลุ่มบทบรรยาย และบทสนทนาของแต่ละตัวละครในนวนิยาย และนำผลลัพธ์ที่ได้ไปประยุกต์ใช้ต่อไป

5. ผลของการทดสอบโปรแกรม

ส่วนของการฝึกฝน

นวนิยายเรื่องที่	จำนวนคำพูดทั้งหมด	จำนวนคำพูดที่นำไปฝึกฝน (เฉพาะรูปแบบประโยคอย่างง่าย)	%ของจำนวนคำพูดที่นำไปฝึกฝนจากทั้งหมด
1	174	71	40.80%
2	148	81	54.73%
3	130	28	21.54%
4	193	69	35.75%
รวมทั้งหมด	645	249	38.60%

% ของจำนวนคำพูดที่นำไปฝึกฝนจากทั้งหมด โดยรวม = 38.60%

โครงสร้างประโยคที่ได้จากการฝึกฝน

จากนวนิยายจำนวน 7 ชุด ซึ่งประกอบด้วยประโยคที่เปนคำพูดภายใต้เครื่องหมาย ”…”

ทั้งหมด 71+134+174+93+81+28+69 = 650 คำพูด

ถูกแปลงเป็นโครงสร้างประโยคได้ทั้งสิ้น 563 โครงสร้าง คิดเป็น 86.62%

จะเห็นได้ว่า มีคำพูดหลายประโยคที่มีโครงสร้างที่เหมือนหรือใกล้เคียงกัน

ส่วนของการทดสอบ

นวนิยายเรื่องที่	จำนวนคำพูดทั้งหมด	จำนวนคำพูดที่ถูกจำแนก<u>จากทั้งหมด</u>	%ของจำนวนคำพูดที่ถูกจำแนก	ผลในการทดสอบ (จำนวนคำพูด)	%ผลในการทดสอบ<u>จากจำนวนคำพูดที่ถูกจำแนก</u>
1	174	30	17.24%	ถูก : 21	ถูก : 70%
				ผิด : 9	ผิด : 30%
2	148	83	56.08%	ถูก : 70	ถูก : 84.34%
				ผิด : 13	ผิด : 15.66%
3	130	25	19.23%	ถูก : 17	ถูก : 68%
				ผิด : 8	ผิด : 32%
4	193	83	43.01%	ถูก : 56	ถูก : 67.47%
				ผิด : 27	ผิด : 32.53%
รวมทั้งหมด	645	221	34.26%	ถูก : 164	ถูก : 74.21%
				ผิด : 57	ผิด : 25.79%

% ของผลในการทดสอบ<u>จากจำนวนคำพูดที่ถูกจำแนก</u> และให้<u>ผลที่ถูกต้อง</u> โดยรวม = 74.21%

จะเห็นได้ว่าจากการทดสอบสามารถจำแนกคำพูดได้ 34.26% จากคำพูดทั้งหมดโดยรวม สำหรับการฝึกฝนด้วยคำพูดทั้งหมด 38.60% ซึ่งเป็นค่าที่ไม่ต่างกันมากนัก ถึงแม้ว่าเปอร์เซนต์ในการจำแนกคำพูด

ในผลดารทดสอบควรมีค่าเท่ากับเปอร์เซนต์ของคำพูดที่นำมาใช้ในการฝึกฝน เนื่องจากมาจากนวนิยายเรื่องเดียวกัน

นอกจากนี้จะเห็นได้ว่าจากคำพูดที่สามารถจำแนกได้และมีผลลัพท์ที่ถูกต้อง คือ สามารถบ่งบอกตัวละครที่เป็นผู้พูดของคำพูดที่ถูกจำแนกได้ตรงตามชุดนวนิยายเริ่มต้นที่มีการระบุตัวละครที่ถูกต้องไว้ คิดเป็นเปอร์เซนต์ได้ 74.21%

6. ปัญหาและอุปสรรค

เนื่องจากโครงงานนี้เป็นการพัฒนาโปรแกรมประมวลภาษาไทย ซึ่งมักเกิดปัญหาด้านการเข้ารหัสตัวอักษร (Encoding) ระหว่างการเขียนโปรแกรมแล้ว ในกลุ่มข้อความที่นำมาประมวลผลนั้นยังมีอักขระสัญลักษณ์พิเศษต่างๆ ซึ่งก็มีปัญหาในการเข้ารหัสตัวอักษรเช่นกัน กลุ่มผู้พัฒนาจึงต้องคอยระวังไม่ให้มีการเข้ารหัสผิดพลาด ระหว่างการส่งข้อมูลไปมา ระหว่างโปรแกรมและคลังข้อมูล (Database) รวมทั้งการอ่านและเขียนไฟล์ตัวอักษร เพื่อไม่ให้อักษรภาษาไทยและสัญลักษณ์พิเศษต่างๆ ถูกส่งไปผิดพลาด และก่อให้เกิด Error ในโปรแกรม

7. แนวทางในการพัฒนาและประยุกต์ใช้ร่วมกับงานอื่นๆ ในขั้นต่อไป

- การพัฒนาโปรแกรมในขั้นต่อไป

ในการประมวลผลเพื่อหาผู้พูดในแต่ละประโยคคำพูดนั้นๆนอกจากการวิเคราะห์จากโครงสร้างของประโยคแล้วโปรแกรมสามารถพัฒนาเพื่อเพิ่มความถูกต้องของการระบุผู้พูดมากขึ้นได้ โดยการวิเคราะห์การเรียงกันของแต่ละย่อหน้าหรือประโยคได้ซึ่งสามารถใชทฤษฎีแบบจำลองภาษาและหลักการของN-gramได้ จากการวิเคราะห์ในหลายๆองค์ประกอบสามารถหาค่าความน่าจะเป็นของผู้พูดในแต่ละประโยคซึ่งสามารถทำให้โปรแกรมสามารถตัดสินใจได้อย่างมีประสิทธิภาพและถูกต้องยิ่งขึ้น

การใช้สรรพนามเพื่อสื่อถึงผู้พูดนั้นโปรแกรมจะต้องมีความเข้าใจในความหมายของสรรพนามนั้นๆและสามารถระบุค่าความน่าจะเป็นของสรรพนามนั้นๆกับตัวละครได้ โปรแกรมจึงจะสามารถระบุผู้พูดใน

ประโยคคำพูดที่ใช้คำสรรพนามสื่อถึงผู้พูดได้ ทางกลุ่มมีความคิดที่จะให้ผู้ใช้สร้างแผนผังตัวละครซึ่งบอกลักษณะของทุกตัวละครให้เรื่องเช่นฉายาเพศอายุความสัมพันธ์กับตัวละครอื่นๆซึ่งช่วยในการระบุตัวละครจากสรรพนามได้

นอกจากนี้ในบทสนทนาหรือคำพูดที่มีความซับซ้อนในการระบุผู้พูดซึ่งไม่สามารถระบุผู้พูดของประโยคคำพูดได้จากบริบทข้างเคียงได้และการเรียงลำดับของผู้พูดนั้นไม่มีความแน่นอน ไม่สามารถประมวลผลจากการเรียงดับของผู้พูดของกลุ่มประโยคคำพูดก่อนหน้าได้ ดังนั้นการประเมินเพื่อหาผู้พูดของบทสนทนานั้นอาจมีความจำเป็นต้องวิเคราะห์จากข้อความในคำพูดนั้นๆซึ่งแต่ละตัวละครจะมีลักษณะการพูดที่แตกต่างกัน โปรแกรมจึงสามารถหาค่าความน่าจะเป็นของผู้พูดได้จากข้อความในคำพูดเหล่านั้น

- การประยุกต์ใช้งานร่วมกับงานอื่นๆ

จุดประสงค์หลักของโปรแกรมเพื่อจำแนกประเภทของข้อความในบทความตามตัวละครนั้นสามารถนำไปประยุกต์ใช้กับโปรแกรมอื่นๆที่เกี่ยวข้องกับการประมวลผลทางภาษาไทยได้ เช่นโปรแกรมแปลงข้อความเป็นเสียงในการแปลงข้อความเป็นเสียงเสียงที่ได้ออกมาสามารถเลือกแบบเสียงต่างๆได้และในบทความประเภทนวนิยายนั้นผู้ใช้จำเป็นต้องระบุเสียงในแต่ละประโรคคำพูดเพื่อให้เกิดเสียงที่แตกต่างกันในการบรรยายคำบรรยายและคำพูดได้ ดังนั้นโปรแกรมที่ทางกลุ่มพัฒนาจิงสามารถประยุกต์รวมกับโปรแกรมแปลงเสียงเป็นข้อความได้ เพื่อผู้ใช้จะไม่มีความจำเป็นต้องอ่านทุกข้อความในบทความและระบุเสียงในแต่ละข้อความโปรแกรมสามารถตัดสินใจระบุเสียงในแต่ละคำพูดได้ทันทีหลังจากผู้ใช้ระบุทุกตัวละครในบทความเรื่องนั้นได้แล้ว

8. ข้อสรุปและข้อเสนอแนะ

จากผลการทดลองจะเห็นได้ว่าถ้านำนวนิยายไปฝึกฝนให้กับโปรแกรมแล้ว เมื่อนำนวนิยายเรื่องเดิมมาทดสอบ จะสามารถให้ผลการทดสอบใกล้เคียงกับจำนวนคำพูดที่นำไปให้ระบบเรียนรู้โครงสร้างประโยคในส่วนการฝึกฝนและสามารถระบุตัวละครที่เป็นผู้พูดของคำพูดบางส่วนได้อย่างถูกต้อง และจากการศึกษาโครงสร้างประโยคของคำพูดในนวนิยายทั้งหมด 7 เรื่องสำหรับส่วนฝึกฝน โครงสร้างบางส่วนมีรูปแบบที่เหมือนหรือใกล้เคียงกัน ซึ่งจะเป็นส่วนช่วยในการจำแนกผู้พูดจากโครงสร้างประโยคได้มีประสิทธิภาพขึ้น ทั้งนี้ประสิทธิภาพจะมากขึ้นเมื่อได้รับการฝึกฝนด้วยนวนิยายจำนวนมากขึ้น สรุปได้ว่า

การจำแนกคำพูดโดยตัวละครที่เป็นผู้พูดส่วนหนึ่งสามารถจำแนกได้โดยอาศัยการเรียนรู้และจดจำโครงสร้างของประโยคได้ และจะทำได้ดียิ่งขึ้นถ้ามีการฝึกฝนมากขึ้น

นอกจากนี้การทดสอบที่ได้ทำไปแล้วนั้นยังไม่ได้ทดสอบกับนวนิยายชุดอื่นๆที่ไม่ได้ใช้ในการฝึกฝน ซึ่งส่วนนี้จะถูกนำไปทดสอบในการพัฒนาโปรแกรมในอนาคต เพื่อปรับปรุงให้โปรแกรมสามารถใช้งานได้กับนวนิยายในหลายๆลักษณะ

9. เอกสารอ้างอิง (Reference)

[1] Human Language Technology Laboratory, National Electronics and Computer Technology Center, “BEST text sets,” retrieved October 31, 2010, from:http://thailang.nectec.or.th/downloadcenter/index.php?option=com_docman&task=cat_view&gid=34&Itemid=61

[2] Human Language Technology Laboratory, National Electronics and Computer Technology Center, “Language model,” retrieved November1, 2010, from:http://www.hlt.nectec.or.th/speech/index.php?option=com_content&view=article&id=63&Itemid=86

[3] Wikipedia, “Language model,” retrieved November1, 2010, from:http://en.wikipedia.org/wiki/Language_model

[4] Paisarn Charoenpornsawat, “Smart Word Analysis for Thai (SWATH)”, retrieved December22,2010, from:http://www.cs.cmu.edu/~paisarn/software.html

[5] Human Language Technology (HLT) Laboratory, NECTEC, “Orchid Corpus”, retrieved December22,2010, from:http://www.hlt.nectec.or.th/orchid/

10. สถานที่ติดต่อของผู้พัฒนา โทรศัพท์ มือถือ โทรสาร อีเมล์

1. นางสาว ณัฐธิดา เตชะนภารักษ์

โทรศัพท์มือถือ : 0866105889

โทรสาร : -

อีเมล์ : focus_nan@hotmail.com

2. นางสาว สุภรณี กัลยาณกุล

โทรศัพท์มือถือ : 0840092019

โทรสาร : -

อีเมล์ : m_dogcat@hotmail.com

3. นางสาว อรณี นิลศรีไพรวัลย์

โทรศัพท์มือถือ : 0890128272

โทรสาร : -

อีเมล์ : little.sun.shine@live.com

11. ภาคผนวก (Appendix)

- คู่มือการติดตั้ง

1. สร้าง Database ชื่อ textclassification
2. Import “textclassification.sql” ลงไปใน database ซึ่งจะสร้าง table ที่จำเป็นสำหรับโปรแกรมขึ้นมาให้
3. Import “textclassification.zip” เป็น java project ชื่อ textclassification ใน Eclipse (หรือ IDE อื่นๆ)
4. Copy ไฟล์ใน folder “moveToTextClassificationFolder” ลงไปไว้ใน folder ชื่อ textclassification ใน workspace (ซึ่งก้คือ folder หลักของ project)
5. ไฟล์ใน ข้อ 4 คือไฟล์ต่างๆที่จำเป็นสำหรับการใช้โปรแกรม รวมทั้ง Swath2.0 ที่เป็น open source
6. ใน IDE คลิกขวา เลือก properties ของ project “textclassification”
7. เลือก Java Build Path -> Add Eternal JARs…
8. เลือก ไฟล์ “mysql-connector-java-5.1.10-bin.jar” ใน folder “eternalLibrary” แล้วกด Ok
9. หากมีปัญหา Console ไม่สามารถแสดงผลภาษาไทย ให้ไปที่ properties ของ project “textclassification” เลือก resource และในหัวข้อ Text File Encoding ให้เลือกเป็น UTF-8

- คู่มือการใช้งาน

การนำนวนิยายมาหาตัวละครที่เป็นเจ้าของบทพูดต่างๆ

1. กำหนดรหัส (ID) ของนวนิยายเรื่องนั้น เพื่อนำมาใช้อ้างอิงในขั้นตอนต่อไป

2. สร้าง Table ใหม่ ใน Database ชื่อ owner ตามด้วย รหัสนวนิยาย เช่น owner00003

3. ใส่รายชื่อของตัวละครทุกตัวที่ปรากฏในนวนิยายเรื่องนั้น ถึงแม้ว่าจะเป็นตัวละครเดียวกันแต่มีการกล่าวถึงชื่ออื่นก็ให้ใส่ลงไปด้วยเช่นกัน แต่จะกำหนดให้ชื่อเหล่านั้นมี รหัสตัวละคร (owner_id) เดียวกัน เช่น

owner_id	raw_owner
1	นางลออ
2	เข็มขาว
4	นางเฉลา
1	แม่ของเข็มขาว
1	แม่
3	พ่อ

4. กำหนด Tag <NE> </NE> ครอบ ชื่อตัวละครทุกตัวใน ไฟล์ตัวอักษรนวนิยายเรื่องนั้น เช่น

"ได้งานอะไรก็เอาทั้งนั้นแหละพ่อ <NE>เข็ม</NE>ไม่เลือกหรอก เสียแต่ว่ามันหาไม่ได้เท่านั้น" หล่อนลุกไปหยิบขวดน้ำและแก้วน้ำมา รินน้ำดื่มอย่างกระหาย
พ่อกับแม่ใช่ว่ามีเงินพอส่งเสียลูกได้ตามสบายหรือก็เปล่าทั้งสิ้น เคราะห์ดีที่บ้านน้าเขาเป็นแหล่งผลิตดอกไม้ประดิษฐ์ส่งขายตามร้าน ต้องการลูกมือมาช่วย <NE>เข็มขาว</NE>ไปพักอยู่ด้วยจึงได้ทั้งที่พักและค่าจ้างแรงงานพอช่วยเหลือตัวเองได้ตลอดเวลาเรียนหนังสืออาศัยเงินช่วยจากพ่อแม่อีก
หล่อนเรียนจบแล้วน้าก็ยังเสียดายความ ขยันของหลานสาวชวนให้อยู่ต่อ <NE>เข็มขาว</NE>สองจิตสองใจ
"น้าเขาก็ชวน<NE>เข็ม</NE>นะจ๊ะให้อยู่กับเขา แต่<NE>เข็ม</NE>คิดถึงพ่อแม่เลยกลับมาก่อน ว่าเอางานเขามาทำด้วยแล้วเอาไปส่ง" หญิงสาวบุ้ยใบ้ไปทางกระเป๋าใบใหญ่ ซึ่งสะพายมาอีกใบหนึ่ง "น้าเขารับงานปักเลื่อมเสื้อส่าเร็จรูปด้วย เดี๋ยวนี้เขาปักกันใหญ่เลย เสื้อยืนหนาๆ เขาก็ปักกัน"
เห็นลูกสาวพูดถึงแต่งาน <NE>นายกิ่ง</NE>กับ<NE>นางลออ</NE>ก็ชักสงสาร เลยเปลี่ยนเรื่องเป็น

5. เซฟชื่อไฟล์นวนิยายจากข้อ 4. ชื่อ “edited_novel_รหัสนวนิยาย.txt” เช่น edited_novel_00003.txt

6. นำไฟล์นวนิยาย จากข้อ 5. ใส่ลงไปใน folder ของโปรแกรม -> workspace / textclassification (ใน folder มีไฟล์ตัวอย่างอยู่แล้ว)

7. เปิด Eclipse (หรือ IDE อื่นๆ) เรียกไฟล์ java ทุกไฟล์ขึ้นมาเพื่อใส่ password ของ database(ทั้งนี้ ไฟล์ไหนมีความจำเป็นต้องเรียกใช้ database จะมี ค่า password ให้แก้ไขโดยง่ายอยู่ด้านบนของโค้ด)

8. จากนั้น เรียกไฟล์ WriteResult.java แล้วแก้ไข รหัสนวนิยายในโค้ดด้านบน แล้ว run

9. โปรแกรมจะสร้าง ไฟล์ใหม่ชื่อ “result_novelรหัสนวนิยาย.txt” ใน folder เดียวกับ ข้อ 6. (ในข้อนี้อาจจะใช้เวลาในการประมวลผลนานหลายชั่วโมง)

10. หลังจากโปรแกรมสร้างไฟล์ result ซึ่งเป็นผลการหาเจ้าของบทพูดโดยโปรแกรมแล้วนั้น เรียกไฟล์ ReadResultToDB.java ขึ้นมา แก้ไข รหัสรหัสนวนิยายในโค้ดด้านบน แล้ว run

11. จาก ข้อ 10. โปรแกรมจะสร้าง table ใหม่ขึ้นมาใน database เพื่อเก็บข้อมูลของที่โปรแกรมประมวลผลและจำแนกกลุ่มของยทพูดต่างๆออกมา

12. ถ้าต้องการไฟล์ ที่มีการ Tag <owner id=… > </owner> คร่อมบทพูดต่างๆนั้น (ตามที่โปรแกรมประมวลผล) ให้เรียกไฟล์ TagOwnerID.java ขึ้นมา แล้ว run

13. ถ้าต้องการเปรียบเทียบผลจากจำแนกของโปรแกรม ให้เรียกไฟล์ CheckResultAsProb.java ขึ้นมาแล้ว run โปรแกรมจะแสดงผลขึ้นมาใน Console รวมทั้งสร้างไฟล์สรุปผลขึ้นมาใน folder ตามข้อ 6.