



รายงานความก้าวหน้า 240-401 โครงการวิศวกรรมคอมพิวเตอร์ 1 ครั้งที่ 1/2564  
การประยุกต์ใช้การประมวลผลภาษาธรรมชาติเพื่อการวิเคราะห์ข่าวตลาด forex gold spot  
An Application of Natural Language Processing on forex gold spot News Analysis

นายชลชัย อภิชาติศิริวัฒน์  
รหัสนักศึกษา 6210110646

อาจารย์ที่ปรึกษาโครงการ

.....

(รศ.ดร.มนตรี กาญจนเดชะ)

รายงานความก้าวหน้าโครงการวิศวกรรมคอมพิวเตอร์นี้เป็นส่วนหนึ่งของการศึกษา  
ตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

## บทที่ 1

### 1.1. ที่มาและความสำคัญ

ในปัจจุบันสังคมมีการลงทุนในส่วนของหุ้น เทรดหุ้น forex หรือ cryptocurrency มากขึ้นและมีอาชีพเกิดขึ้นมากมายในวงการของการเล่นหุ้น หรือ Stock และมีการใช้ AI ต่าง ๆ เพื่อวิเคราะห์หาแนวโน้มหรือคาดการณ์ล่วงหน้าของกราฟหุ้น ซึ่งบางครั้งย่อมเกิดปัญหา AI วิเคราะห์ได้แค่ทฤษฎีของกราฟและเครื่องมือต่าง ๆ ที่ใช้กัน และการวิเคราะห์ข่าวนั้นย่อมคาดการณ์ได้ยากเพราะข่าวสารที่มากมายย่อมส่งผลกระทบต่อนั้น ๆ

ดังนั้นการลงทุนในส่วนของหุ้น เทรดหุ้น หรือ cryptocurrency ควรต้องมีการใช้เครื่องมือวิเคราะห์ข่าวเข้ามาเป็นส่วนหนึ่งของการคาดการณ์จากการเก็งกำไร หรือ ลงทุน ซึ่งเห็นได้ชัดว่า AI ที่มีอยู่แล้วเช่น EA เป็นตัวช่วยให้เราไม่ต้องมาเทรดเองโดยเป็นการเซตค่าจาก เครื่องมือต่าง ๆ ตามเทคนิคที่อยู่ของเราเอง ซึ่งในบางครั้งเรื่องของเทคนิคก็ผิดพลาดเพราะบางหลักทรัพย์มีความผันผวนสูง ทำให้เทคนิคที่เราเซตไว้ว่าจะเกิดการผิดพลาดได้สูง และส่งผลให้เราขาดทุน

ในการทำ AI วิเคราะห์ข่าวหุ้นได้มีการนำ Machine Learning NLP มาวิเคราะห์ ซึ่งเป็นตัวช่วยทำให้คอมพิวเตอร์วิเคราะห์ข่าวสารหรือข้อความได้อย่างง่ายดาย NLP เป็นสาขาหนึ่งในการเรียนรู้ของเครื่องด้วยความสามารถของคอมพิวเตอร์ในการทำความเข้าใจ วิเคราะห์ จัดการ และสร้างภาษามนุษย์ได้ ในปัจจุบันเทคโนโลยี Machine Learning NLP เป็นที่นิยมในการเอามาทำ AI เช่น การดึงข้อมูล การแปลภาษา การทำให้ข้อความง่ายขึ้น การวิเคราะห์ให้ความรู้สึกรู้สึกของผู้ใช้ การสรุปข้อความ ตัวกรองสแปม คาดการณ์ผลการค้นหาของผู้ใช้ แก้ไขข้อผิดพลาดอัตโนมัติ เป็นต้น Natural Language Processing(NLP) for Machine Learning หรือการประมวลผลภาษาธรรมชาติด้วย Python ซึ่งภาษา Python เป็นภาษาที่รวดเร็วและในการทำ NLP จะใช้ Natural Language Toolkit (NLTK) เป็น Library Opensource ยอดนิยมใน Python

### 1.2. วัตถุประสงค์ของโครงการ

1. พัฒนาระบบที่ใช้หลักการของ NLP
2. วิเคราะห์ข่าวต่าง ๆ ที่เกี่ยวกับตลาด forex gold spot ซึ่งเป็นข่าวที่อยู่ในรูปแบบออนไลน์ เพื่อใช้เป็นข้อมูลสำหรับระบบ Robot Trader

### 1.3. ขอบเขตโครงการ

1. สร้าง Machine Learning AI วิเคราะห์ข่าว โดยใช้ Language Processing(NLP)
2. เฉพาะ forex gold spot
3. วิเคราะห์แนวโน้มขึ้นหรือลง แสดงเป็น 1 0 และ -1

4. วิเคราะห์เฉพาะข่าวสำคัญที่อยู่ในตารางปฏิทิน
5. วิเคราะห์เฉพาะตลาดอเมริกา ช่วงเวลา 7.00 pm – 3.00 am ตามเวลาประเทศไทย
6. จัดทำให้แสดงข้อมูลบนเว็บไซต์

#### 1.4. แผนการดำเนินงาน

[illegible]

[illegible]

## บทที่ 2

### ทฤษฎีและความรู้พื้นฐาน

#### 2.1 Forex gold spot

Forex คือ ตลาดแลกเปลี่ยนเงินตราต่างประเทศ (หรือที่เรียกว่า forex หรือ FX) หมายถึง ตลาดที่ซื้อขายกันโดยตรง (OTC) ระดับโลกซึ่งเทรดเดอร์ นักลงทุน สถาบัน และธนาคารจะแลกเปลี่ยน กัน กำไร ซื้อและขายสกุลเงินของโลกการเทรดจะเกิดขึ้นใน ‘ตลาดระหว่างธนาคาร’ ซึ่งเป็นช่องทางทางออนไลน์ที่มีการเทรดสกุลเงิน 24 ชั่วโมงต่อวัน ห้าวันต่อสัปดาห์ Forex เป็นหนึ่งในตลาดการเทรดที่ใหญ่ที่สุดโดยมีเงินหมุนเวียนทั่วโลกในแต่ละวันโดยประมาณมากกว่า 5 ล้านล้านดอลลาร์สหรัฐฯ

Gold Spot คือตลาดสากลในการซื้อขายทองคำทั่วโลก เป็นตลาดที่มี Volume สูงมาก เพราะเป็นการซื้อขายทองคำจากทั่วโลก มักเรียกกันอีกชื่อหนึ่งว่า “การเทรดทองคำในตลาดโลก” โดยจะเป็นการซื้อขายในรูปแบบสัญญาหรือใบรับประกัน ไม่ได้มีการจัดส่งทองคำแท้ให้ผู้ซื้อ

#### 2.1.1 ปัจจัยที่ส่งผลกระทบต่อราคาทองคำ

ในปัจจุบัน ราคาทองคำมีความผันผวนค่อนข้างต่ำในระยะยาวจึงเป็นสินทรัพย์ที่ปลอดภัยถึงแม้ว่าทองคำจะเป็นสินทรัพย์ที่ปลอดภัย แต่ก็มีความเสี่ยงและมีปัจจัยที่ต้องพิจารณาด้วย เช่น ปัจจัยที่จะมีผลต่อทิศทางการเคลื่อนไหวของราคาทองคำและส่งผลต่อกำไรที่นักลงทุนจะได้รับ ในปัจจุบันทองคำยังได้รับความนิยมอยู่ เนื่องจากเป็นสินทรัพย์ที่มีความสามารถในการป้องกันความเสี่ยงในรูปแบบต่างๆ ได้ เช่น ความเสี่ยงจากภาวะเงินเฟ้อ ความผันผวนของอัตราแลกเปลี่ยนเงินตราต่างประเทศ ภาวะเศรษฐกิจหดตัว ไปจนถึงการเปลี่ยนแปลงทางการเมือง เพราะทองคำเป็นสิ่งที่มีความมั่นคงในตัวเองอยู่ตลอดเวลา จึงทำให้การลงทุนในทองคำสามารถกระจายความเสี่ยงของพอร์ตการลงทุน และยังนำไปใช้สร้างผลกำไรหากจับจังหวะซื้อขายได้ถูกทางอย่างไรก็ตาม ทองคำก็มีความเสี่ยงจึงต้องทำการวิเคราะห์ถึงปัจจัยที่มีผลต่อทิศทางราคาทองคำ โดยหลัก ๆ แล้ว ควรพิจารณาปัจจัยใน 3 สัญญาณ ได้แก่ สัญญาณระยะยาว สัญญาณระยะกลาง และสัญญาณระยะสั้น และอีกหนึ่งอย่างที่สำคัญคือข่าวสาร

##### 2.1.1.1 สัญญาณระยะยาว

นักลงทุนควรพิจารณาราคาทองคำย้อนหลังในอดีตไปประมาณ 7-10 ปี เพื่อเห็นภาพทิศทางราคาทองคำที่แม่นยำมากขึ้น ตัวอย่างเช่น สถิติราคาทองคำ 7 ปีย้อนหลัง ตั้งแต่ปี 2558 จนถึงต้นปี 2564 พบว่าราคาทองคำ (Gold Spot) มีระดับต่ำสุดของแต่ละปีสูงขึ้นเรื่อย ๆ (ภาษาในตลาดทองคำเรียกว่า การยกฐานราคาในระดับต่ำสุดขึ้น) ซึ่งเหตุการณ์นี้ทำให้นักวิเคราะห์ทองคำทั่วโลกประเมินว่าทิศทางราคาทองคำยังเป็นขาขึ้นในระยะยาวค่อนข้างชัดเจน

โดยสภาพทองคำโลก ได้อธิบายถึงปัจจัยที่ทำให้ทิศทางราคาทองคำจะยังคงเป็นขาขึ้นต่อไป นั่นคือ ในช่วงวิกฤติ COVID-19 ที่ผ่านมา ทองคำเป็นสินทรัพย์ที่มีราคาผันผวนน้อย ขณะเดียวกันก็ให้ผลตอบแทนค่อนข้างสม่ำเสมอ ประกอบกับนักลงทุนยังคงมองว่าการลงทุนในสินทรัพย์อื่น ยังคงมีความเสี่ยงสูง ขณะที่ อัตราดอกเบี้ยทั่วโลกยังคงอยู่ในระดับต่ำและเศรษฐกิจโลกอยู่ในภาวะชะลอตัว จึงทำให้กระแสเงินลงทุนไหลเข้าสู่ตลาดทองคำอย่างต่อเนื่อง โดยเฉพาะความต้องการจากผู้บริโภคชาวจีนและอินเดีย ซึ่งล้วนแล้วแต่เป็นปัจจัยบวกต่อทิศทางราคาทองคำทั้งสิ้น

สำหรับนักลงทุนที่กำลังตัดสินใจลงทุนหรือว่าถือทองคำอยู่แล้วและเน้นลงทุนระยะยาว ยังสามารถลงทุนและถือต่อไปได้ โดยพฤติกรรมการลงทุนทองคำในระยะยาวจะลงทุนตั้งแต่ 6 เดือนขึ้นไป (มากกว่า 2 ไตรมาส) หรืออาจถือข้ามปี ซึ่งกลยุทธ์ในการลงทุน ก็คือ หายใจซื้อในช่วงต้นปีหรือช่วงตรุษจีน จากนั้นให้ถือและรอจังหวะทยอยขายในช่วงปลายไตรมาส 3 หรือ ก่อนสิ้นปี นอกจากนี้ ยังมีกลุ่มนักลงทุนทองคำที่ทยอยลงทุนไปเรื่อย ๆ และถือเป็นระยะเวลาหลายปีหรือสะสมเพื่อเป็นมรดก เพราะเชื่อว่าการถือทองคำเกิน 10 ปี จะมีแต่กำไร

#### 2.1.1.2 สัญญาณระยะปานกลาง

นักลงทุนควรพิจารณาราคาทองคำเป็นรายไตรมาส โดยสถิติในช่วง 3 ปีที่ผ่านมา พบว่าราคาทองคำมักปรับขึ้นสู่ระดับสูงสุดของปี ในช่วงไตรมาส 3 และราคาจะปรับลดลงเมื่อเข้าสู่ไตรมาส 4 และไตรมาส 1 ของปีถัดไป เช่น ราคาปรับขึ้นไปที่ระดับ 2,075 เหรียญสหรัฐต่อออนซ์ในช่วงเดือนสิงหาคม ปี 2563 หลังจากนั้นราคาเริ่มอ่อนตัวลง และล่าสุดไตรมาส 1 ปี 2564 ราคาทองคำอ่อนตัวลงสู่ระดับ 1,767 เหรียญสหรัฐต่อออนซ์ และปรับลดลงสู่ระดับต่ำสุดที่บริเวณ 1,676 เหรียญสหรัฐต่อออนซ์ และเมื่อเข้าสู่ไตรมาส 2 ราคาจะมีสัญญาณค่อย ๆ ปรับขึ้น โดยประเมินว่าในไตรมาส 3 ปีนี้ ราคาทองคำก็จะปรับขึ้นไปที่ระดับสูงสุดของปีเหมือนภาพในอดีต

หากมองจากปัจจัยพื้นฐานจะพบว่า ช่วงต้นปีนักลงทุนเริ่มคาดการณ์ว่าเศรษฐกิจโลกจะฟื้นตัวอย่างชัดเจนหลังจากวัคซีน COVID-19 มีประสิทธิภาพ จึงเห็นการปรับประมาณการการเติบโตทางเศรษฐกิจ ถือเป็นปัจจัยกดดันให้ราคาทองคำ ซึ่งเป็นสินทรัพย์ปลอดภัย (Safe Haven) อ่อนตัวลง

อย่างไรก็ตาม หลังจากการแพร่ระบาด COVID-19 รอบล่าสุด นักลงทุนประเมินว่าเศรษฐกิจโลกในปีนี้จะฟื้นตัวในลักษณะค่อยเป็นค่อยไป ประกอบกับ มาตรการผ่อนคลายทางการเงินของธนาคารกลางต่าง ๆ ทั่วโลก จะไม่เปลี่ยนแปลงในช่วงครึ่งปีหลัง จึงเริ่มเห็นการเปลี่ยนทิศทางของราคาทองคำที่มีสัญญาณฟื้นตัวขึ้น จากแนวโน้มที่จะยังคงมีเม็ดเงินถูกอัดฉีดเข้ามาเพื่อกระตุ้นเศรษฐกิจ ส่งผลผลักดันให้ราคาทองคำปรับตัวขึ้น

สำหรับนักลงทุนทองคำในระยะปานกลางจะเน้นลงทุนเป็นรายเดือนและไม่เกิน 3 เดือน (1 ไตรมาส) โดยพยายามจับจังหวะการแกว่งตัวของราคาทองคำเพื่อหาจังหวะซื้อและขายเพื่อทำกำไร ซึ่งกลยุทธ์ในการลงทุน ก็คือ รอจังหวะลงทุนเมื่อเห็นราคาทองคำอ่อนตัวลง และรอขายทำกำไรเมื่อราคาเริ่มปรับตัวขึ้น

### 2.1.1.3 สัญญาณระยะสั้น

นักลงทุนจะวิเคราะห์ราคาทองคำเป็นรายวันด้วยการพิจารณาราคาสินทรัพย์อื่น ๆ ประกอบ เพื่อดูทองคำกับสินทรัพย์อื่น ๆ ว่ามีความเคลื่อนไหวด้านราคาอย่างไร โดยจะพิจารณาใน 2 ปัจจัย ได้แก่ ปัจจัยที่ส่งผลไปในทิศทางตรงข้ามกับราคาทองคำ และปัจจัยที่ส่งผลไปในทิศทางเดียวกันกับราคาทองคำ เช่น ตลาดหุ้นสหรัฐอเมริกา อัตราผลตอบแทนพันธบัตรรัฐบาล และค่าเงินสกุลดอลลาร์สหรัฐ

#### 2.1.1.3.1 ตลาดหุ้นสหรัฐอเมริกา

หากตลาดหุ้นสหรัฐฯ ปรับขึ้น ราคาทองคำมีแนวโน้มปรับตัวลดลง เนื่องจากตลาดหุ้นเป็นสินทรัพย์เสี่ยง ขณะที่ทองคำเป็นสินทรัพย์ปลอดภัย ราคาจึงเคลื่อนไหวในทิศทางตรงกันข้าม

#### 2.1.1.3.2 อัตราผลตอบแทนพันธบัตรรัฐบาล

จะพิจารณาจากอัตราผลตอบแทนพันธบัตรรัฐบาลสหรัฐอเมริกา อายุ 10 ปี ซึ่งถือเป็นปัจจัยสำคัญที่สะท้อนภาวะเศรษฐกิจและทิศทางอัตราดอกเบี้ยในตลาดเงินและตลาดทุนของสหรัฐฯ หากอัตราผลตอบแทนพันธบัตรรัฐบาลสหรัฐอเมริกาอายุ 10 ปีปรับตัวขึ้น ส่วนใหญ่ค่าเงินดอลลาร์จะแข็งค่าขึ้นตามไปด้วย รวมทั้งนักลงทุนประเมินว่าเศรษฐกิจสหรัฐอเมริกาคงเติบโตและอัตราดอกเบี้ยมีแนวโน้มเป็นขาขึ้น ก็จะส่งผลให้ราคาทองคำปรับตัวลดลง เนื่องจากทองคำไม่มีผลตอบแทนอยู่ในรูปของดอกเบี้ย ดังนั้นเมื่ออัตราดอกเบี้ยปรับตัวขึ้น การลงทุนในทองคำจึงถูกลดความน่าสนใจ

#### 2.1.1.3.3 ค่าเงินสกุลดอลลาร์สหรัฐ

จะมีความสัมพันธ์ในเชิงลบกับราคาทองคำโลก กล่าวคือ ถ้าค่าเงินดอลลาร์สหรัฐอ่อนลงเมื่อเทียบกับเงินสกุลสำคัญของโลก เช่น เงินยูโร เงินเยน หรือพิจารณาจาก US Dollar Index ก็ได้เช่นกัน ราคาทองคำโลกจะสูงขึ้น เพราะราคาทองคำซื้อขายเป็นสกุลเงินดอลลาร์สหรัฐ เมื่อค่าเงินดอลลาร์อ่อนลง ทองคำจะมีราคาถูกลงเมื่อเทียบกับเงินสกุลอื่นที่นักลงทุนถือไว้ จึงสร้างแรงซื้อเข้ามาดันให้ราคาทองคำปรับตัวเพิ่มสูงขึ้น

#### 2.1.1.4 ข่าวสาร

ข่าวสารเป็นสิ่งสำคัญอีกอย่างที่ราคาทองคำจะมีความผันผวนเนื่องจากข่าวสารจะมีทั้งข่าวดีที่ส่งผลดีกับทองคำแล้ว แต่ก็ยังมีข่าวสารที่ไม่ดีส่งผลต่อทองคำในทิศทางลงส่วนใหญ่ ข่าวสารที่กระทบถึงทองคำ เช่น ข่าวสารเกี่ยวกับน้ำมัน ข่าวสารเกี่ยวกับเศรษฐกิจ อัตราการว่างงาน หรือโรงงานต่าง ภาวะเงินเฟ้อ หรือ เงินฝืด และข่าวสารที่สำคัญมากสำหรับทองคำจะเป็นข่าวสารจากธนาคารโลก หรือ ธนาคารกลางต่าง ๆ

ข่าวสาร	ข่าวประเภท	อ้างอิงค์
1. Census Bureau	<u>กระทรวงพาณิชย์</u> <u>สหรัฐ</u>	[7]
2. Us Department of labo	กระทรวงแรงงาน	[8]
3. Energy information Administration	<u>ข้อมูลด้านพลังงาน</u>	[9]
4. กองทุน SPDR	การซื้อขายทองคำ ของกองทุน SPDR	[10]
5. Federal Reserve	ธนาคารกลางสหรัฐ	[11]
6. Bloomberg	รายงานข่าวทั่วไป รอบโลก	[12]
7. Twitter	ข่าวทั่วไป	[13]

รูปที่ 1 แหล่งข่าวสาร

ในปัจจุบัน เทคโนโลยี Machine Learning NLP เป็นที่นิยมในการเอามาทำ AI เช่น การดึงข้อมูล การแปลภาษา การทำให้ข้อความง่ายขึ้น การวิเคราะห์ให้ความรู้สึกของผู้ใช้ การสรุปข้อความ ตัวกรองสแปม คัดการณ์ผลการค้นหาของผู้ใช้ แก้ไขข้อผิดพลาดอัตโนมัติ เป็นต้น Natural Language Processing(NLP) for Machine Learning หรือการประมวลผลภาษาธรรมชาติด้วย Python ซึ่งภาษา Python เป็นภาษาที่รวดเร็วและ

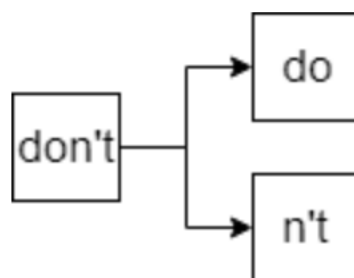


ในการทำ NLP จะใช้ Natural Language Toolkit (NLTK) เป็น Library Opensource ยอดนิยมใน Python ซึ่งมีรายละเอียดดังนี้

## 2.2 Processing text using NLP

ทั่วไปสำหรับข้อความก่อนการประมวลผลประกอบด้วย 4 อย่าง

2.2.1 Sentence segmentation ในขั้นตอนแรกของการเตรียมประโยคข้อความจะถูกแบ่งออกเป็นประโยคข้อความที่ใช้ เช่น ภาษาอังกฤษ เครื่องหมายวรรคตอน โดยเฉพาะอักขระหยุด เครื่องหมายอัศเจรีย์และเครื่องหมายคำถามสามารถใช้ระบุจุดสิ้นสุดของประโยคได้อย่างไรก็ตาม อักขระจุดยังสามารถใช้เป็นตัวย่อของข้อความได้ เช่น Ms. หรือ UK ซึ่งในกรณีนี้ อักขระหยุดไม่ได้หมายถึงจุดสิ้นสุดของประโยค ในกรณีเหล่านี้ใช้อักขระย่อเพื่อหลีกเลี่ยงการแบ่งประเภทขอบเขตประโยคที่ไม่ถูกต้อง เมื่อข้อความมีคำศัพท์เฉพาะ เราจะต้องสร้างพจนานุกรมคำย่อเพิ่มเติมเพื่อหลีกเลี่ยงการทำเครื่องหมายผิดหลักธรรมชาติ ตัวอย่างการทำให้เป็นมาตรฐาน



รูปที่ 2 เป็นการทำให้ Tokenization [6]

2.2.2 Tokenization คือ การแบ่งข้อความออกเป็นคำและเครื่องหมายวรรคตอนที่เป็นเครื่องหมายเช่น เดียวกับการแบ่งประโยคเครื่องหมายวรรคตอน ตัวอย่างเช่น U.K. ควรจะเป็นเครื่องหมาย และ don't ควรแบ่งออกเป็นสองเครื่องหมาย do และ n't

Stemming และ lemmatization เป็นส่วนสำคัญของกระบวนการทำให้เป็นมาตรฐาน การทำให้เป็นมาตรฐานประกอบด้วยการสกัดคำที่ต้องระบุต้นคำโดยการลบต่อท้าย เช่น -ed และ -ing ไม่จำเป็นต้องเป็นคำ ในทำนองเดียวกัน lemmatization เกี่ยวข้องกับการลบคำนำหน้าและส่วนต่อท้าย ความแตกต่างที่สำคัญคือผลที่ได้คือภาษา ผลที่ได้นี้เรียกว่าการอ้างอิง ตัวอย่างของ Stemming และ lemmatization ดังรูปที่ 2

	Word 1	Word 2	Word 3
Original	studies	playing	best
Stem	studi	play	best
Lemma	study	play	good

รูปที่ 3 รูปแบบของการใช้งาน Stemming และ lemmatization [6]

ทั้ง 2 เทคนิคที่กล่าวมาจะช่วยในการลดสัญญาณรบกวนในข้อความโดยแปลงคำให้อยู่ในรูปแบบพื้นฐาน เช่น การประเภทข้อความหรือการจัดกลุ่มเอกสาร ซึ่งการรักษาความหมายของคำเป็นสิ่งสำคัญ ควรใช้ lemmatization มากกว่าการวิเคราะห์ ตัวอย่างเช่น คำนามและคำกริยา ซึ่งทำให้สูญเสียความหมายดั้งเดิมไป เทคนิคการทำให้เป็นมาตรฐานอื่น ๆ ได้แก่ การขยายคำย่อ การลบตัวเลขและเครื่องหมายวรรคตอน การแก้ไขคำผิดพลาตทางไวยากรณ์ การดำเนินการเหล่านี้ส่วนใหญ่สามารถทำได้โดยใช้นิพจน์ทั่วไป

2.2.3 Part of speech tagging ขั้นตอนนี้จะเป็นการแบ่งเครื่องหมายเป็น part of speech (POS) หรือที่เรียกว่าคำศัพท์ หรือ หมวดหมู่คำศัพท์ คำที่ประกอบด้วยคำนาม,คำกริยา,คำบุพบท,คำกริยาวิเศษณ์ ดังตารางต่อไปนี้จะแสดงคำและตัวอย่าง ส่วยสัญลักษณ์ จะใช้ lemmatization ซึ่งเป็นสิ่งจำเป็นสำหรับการตั้งชื่อ บุคคล

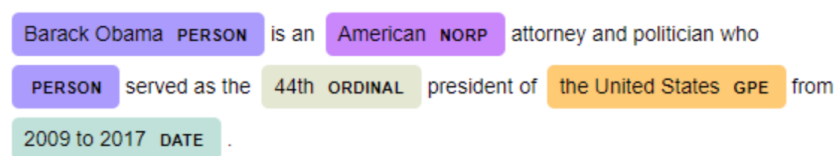
Lexical category	Example
Noun	book, girl, forest, moss
Verb	play, study, write, choose
Adjective	happy, short, brown, cool
Preposition	at, about, over, on
Determiner	the, a, this, those
Conjunction	and, but, or, if
Pronoun	I, she, you, they

รูปที่ 4 ตัวอย่างการติดแท็กจัดกลุ่มข้อความ POS-taggers [6]

POS-taggers มี 3 ประเภท ได้แก่ ตามกฎสถิติและตามการเรียนรู้เชิงลึก กฎตามเครื่องหมายขึ้นอยู่กับว่ากฎที่ชัดเจนเพื่อทำเครื่องหมาย เช่น บทความต้องตามด้วยคำนาม เพื่อกำหนดเครื่องหมาย ตามกฎสถิติใช้แบบจำลองความน่าจะเป็นในการมาร์คแต่ละคำหรือลำดับของคำ กฎตามแท็กตาม

กฎนั้นแน่นยำมาก แต่ก็ยังขึ้นอยู่กับภาษาด้วย การขยาย tagger เพื่อรองรับภาษาอื่น ๆ ตัวติดแท็กภาษาอังกฤษนั้นสร้างได้ง่ายกว่าและไม่ขึ้นกับภาษา และมีการใช้วิธีผสมผสานของแบบจำลองตามกฎและแบบจำลองทางสถิติ โดยที่แบบจำลองจะได้รับการฝึกอบรมเกี่ยวกับชุดประโยคที่ติดแท็กล่วงหน้า วิธีการแบบไฮบริดและการเรียนรู้เชิงลึกจะสามารถปรับปรุงการติดแท็กได้ตามบริบท

2.2.4 Named entity Recognition คือการแบ่งกลุ่มของเครื่องหมาย การแบ่งกลุ่มหมายถึงการติดแท็ก หนึ่งในกลุ่มคำที่ใช้มากที่สุด คือกลุ่มคำนามที่ประกอบด้วยตัวกำหนด คำคุณศัพท์ และคำนาม เช่น a happy unicorn ประโยค He found a happy unicorn ประกอบด้วยสองส่วน he และ a happy unicorn



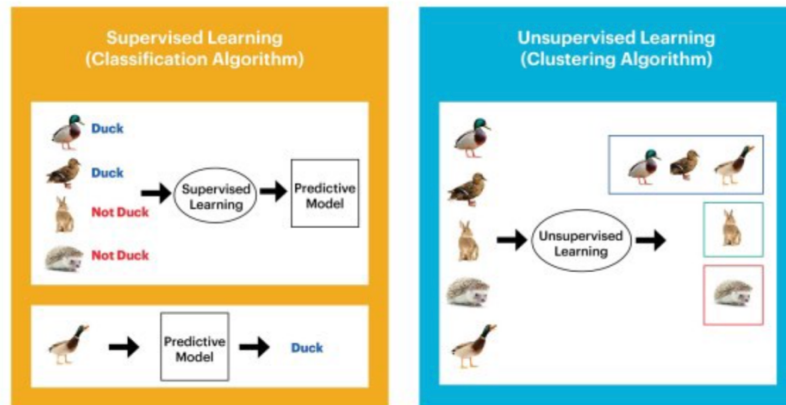
รูป 5 การติดแท็กที่อ้างอิงถึงวัตถุเฉพาะ Named entity [6]

Named entity เป็นคำนามที่อ้างอิงถึงวัตถุเฉพาะ เช่น บุคคล องค์กร สถานที่ วันที่ และหน่วยงาน ภูมิศาสตร์เป้าหมายของขั้นตอน Named entity Recognition คือการระบุชื่อบุคคลที่กล่าวถึงในข้อความ

## 2.3 Machine Learning

As Brink และคนอื่น ๆ ให้คำจำกัดความไว้ว่า Machine Learning (ML) คือการใช้ประโยชน์จากรูปแบบของข้อมูลในอดีตเพื่อตัดสินใจเกี่ยวกับข้อมูลใหม่ หรือเป็นทาง Google หัวหน้านักวิทยาศาสตร์ด้านการตัดสินใจ หรือ Cassie Kozyrkov ซึ่งให้เห็นว่า Machine Learning เป็นเพียงตัวติดฉลาก อธิบายไว้เกี่ยวกับบางสิ่งบางอย่างและบอกให้รู้ว่าควรได้รับฉลากอะไร การใช้เทคนิค ML มีประโยชน์เมื่อปัญหานั้นซับซ้อนเกินกว่าจะแก้ไขด้วยการเขียนโปรแกรม เช่น แยกแยะสายพันธุ์แมวต่าง บนรูปภาพ หรือโซลูชันจำเป็นต้องปรับเปลี่ยนเมื่อเวลาผ่านไป เช่น การจดจำข้อความที่เขียนด้วยลายมือ

โดยทั่วไปแล้ว Machine Learning จะแบ่งออกเป็น Machine Learning ที่จะต้องดูแล และไม่ต้องดูแล เราสามารถการเรียนรู้ภายใต้การดูแลเมื่อข้อมูลการฝึกอบรมในอดีตของเรามีป้ายกำกับ (เช่น duck และ no duck ในรูปตัวอย่างด้านล่าง) ในทางกลับกันการเรียนรู้แบบไม่มีผู้ดูแลจะถูกนำมาใช้เมื่อไม่มีป้ายกำกับในข้อมูล วิธีการเรียนรู้ของเครื่องที่ไม่ได้รับการดูแลมีเป้าหมายเพื่อสรุปหรือบีบอัดข้อมูลการฝึกอบรมพร้อมป้ายกำกับ สแปม/ไม่สแปม ในกรณีหลัง เราจะต้องตรวจหาอีเมลผิดปกติตามชุดการฝึกอบรมของอีเมล



รูปที่ 6 ความแตกต่างระหว่างการเรียนรู้แบบมีผู้ดูแล และ ไม่มีผู้ดูแล [6]

## 2.4 Extracting features from text

อัลกอริทึม ของ Machine Learning ทั้งหมดต้องการข้อมูลดิจิทัลเป็นอินพุต ซึ่งหมายความว่าข้อมูลและข้อความจะต้องถูกแปลงเป็นตัวเลข ขั้นตอนการแยกคุณลักษณะของ NLP

### 2.4.1 Count-based strategies

เป็นวิธีที่ง่ายที่สุดในการแปลงข้อความเป็นเวกเตอร์ตัวเลขคือการใช้วิธี Bag-of-Words (BoW) หลักการของ BoW คือการแยกคำที่ไม่ซ้ำกันทั้งหมดจากข้อความและสร้างคลังข้อความที่เรียกว่าคำศัพท์ การใช้คำศัพท์แต่ละประโยคสามารถแสดงเวกเตอร์ประกอบด้วย 1 และ 0 ขึ้นอยู่กับว่ามีคำศัพท์อยู่ในประโยคหรือไม่ รูปด้านล่างแสดงตัวอย่างของเมทริกซ์ที่สร้างขึ้นโดยใช้วิธี BOW ในประโยคทำประโยคที่ทำให้เป็นมาตรฐาน

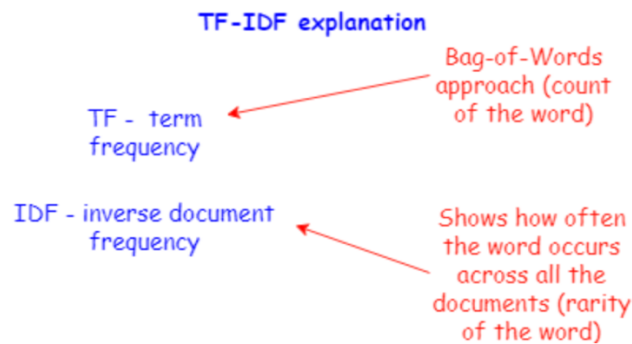
```
[ 'Rabbit jumped over a large fox.',
  'Unicorns are magical creatures living in dark forests.',
  'Unicorns and rabbits live in forests.',
  'Google is being sued by European Union',
  'Apple and Google are some of the biggest companies in the world' ]
```

รูปที่ 7 เป็นขั้นตอนการเคลียข้อความ ตัดอักขระออก Sentence [6]

	apple	big	company	creature	dark	european	forest	fox	google	jump	large	live	magical	rabbit	sue	unicorn	union	world
0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	0	0
1	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0	1	0	0
2	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	0
3	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0
4	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

รูปที่ 8 เป็นสร้างครั้งคำศัพท์เพื่อแยกว่าคำศัพท์ตัวไหนมีการใช้บ่อย หลักการของ BoW [6]

จะสามารถจัดกลุ่มแพ็กเข้าด้วยกันเพื่อเพิ่มบริบทเพิ่มเติมไปยังคำศัพท์ วิธีนี้เรียกว่า N-gram วิธี N-gram คือลำดับของเครื่องหมาย N เช่น 2-gram คำลำดับของคำสองคำ ในขณะที่ trigram คือลำดับของสามเมื่อเลือกคำศัพท์แล้ว ไม่ว่าจะเป็น 1-, 2- หรือ 3- gram จะต้องนับจำนวน gram เราสามารถใช้วิธี BoW ได้ ข้อเสียของแนวทางนี้คือคำที่นิยมมีความสำคัญเกินไป ดังนั้น วิธีที่นิยมใช้กันมากที่สุดจึงเรียกว่า term frequency - inverse document frequency (TF-IDF)



รูปที่ 9 เป็นการจัดความสำคัญของคำเทียบกับความยาวประโยค [6]

TF-IDF ประกอบด้วย term frequency (TF) เป็นการจัดความสำคัญของคำเทียบกับความยาวประโยคและ inverse document frequency (IDF) ซึ่งจัดจำนวนแถวเอกสารที่ gram เกิดเมื่อเทียบกับจำนวนของแถวในเอกสารเพื่อเน้นความหายากของคำ ตามที่คิดไว้ คำที่ปรากฏอยู่บ่อยในเอกสารแต่ไม่ค่อยปรากฏในเอกสารทั้งหมด คำหนึ่งจะมีคะแนน TF-IDF สูงกว่า หากพบบ่อยในเอกสารแต่จะไม่พบพบ่อยในชุดเอกสารทั้งหมด ดังรูปที่ 9 ตัวอย่างของเมทริกซ์ที่สร้างขึ้นโดยใช้วิธี TF-IDF ในประโยคตัวอย่างที่เห็นก่อนหน้านี้ สังเกตว่าคะแนนของคำว่า fox แตกต่างจากคะแนนที่กำหนดให้กับ rabbit

	apple	big	company	creature	dark	european	forest	fox	google	jump	large	live	magical	rabbit	sue	unicorn	union	world
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.52	0.00	0.52	0.52	0.00	0.00	0.42	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.45	0.45	0.00	0.36	0.00	0.00	0.00	0.00	0.36	0.45	0.00	0.00	0.36	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.50	0.00	0.50	0.00	0.50	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.52	0.00	0.00	0.42	0.00	0.00	0.00	0.00	0.00	0.52	0.00	0.52	0.00
4	0.46	0.46	0.46	0.00	0.00	0.00	0.00	0.00	0.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.46

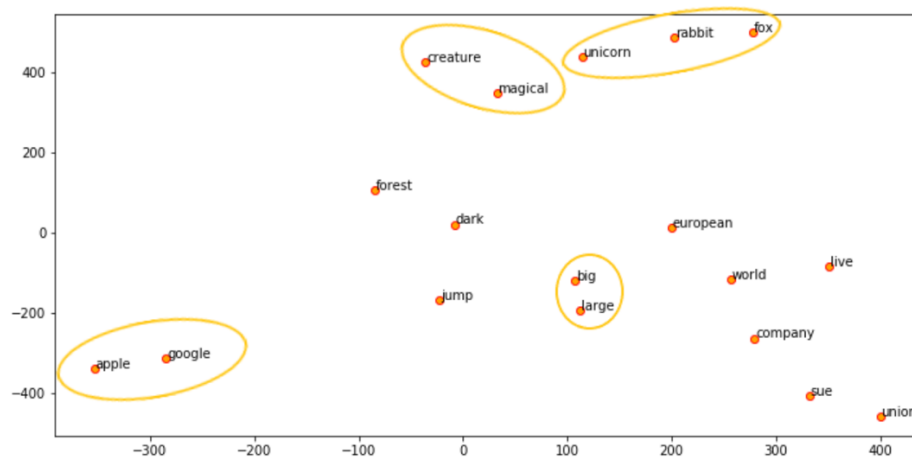
รูปที่ 10 แสดงถึงความสำคัญของแต่ละประโยค [6]

## 2.4.2 Advanced strategies

วิธีการ count-based แม้จะสามารถใช้เพื่อจัดลำดับของคำ (N-grams) แต่ก็ไม่ได้จัดบริบททางความหมายของคำซึ่งเป็นแกนหลักของแอปพลิเคชัน NLP จำนวนมาก เทคนิคการฝังคำใช้เพื่อแก้ปัญหา การฝังคำคำศัพท์จะถูกแปลงเป็นเวกเตอร์เพื่อให้คำที่มีบริบทคล้ายกันอยู่ใกล้กัน

Word2Vec เป็นเฟรมเวิร์คจาก Google ที่ใช้โครงข่ายประสาทเทียมแบบต้นเพื่อฝึกโมเดลฝังคำสั้ง อัลกอริธึม โดย Word2Vec มี 2 ประเภท ประเภทที่ 1 Skip-gram ซึ่งใช้ทำนายบริบทรอบๆ คำที่กำหนด ในขณะที่โมเดล Continuous Bag of Words ( CBOW) ใช้เพื่อทำนายคำถัดไปตามบริบทที่กำหนด

วิธีที่ 2 GloVe วิธี Global Vector ใช้สถิติการเกิดขึ้นร่วมเพื่อสร้างช่องว่างเวกเตอร์ วิธีนี้เป็นส่วนขยายต่อมาจาก Word2Vec ที่มีแนวโน้มว่าจะให้การฝังคำที่ดีกว่า

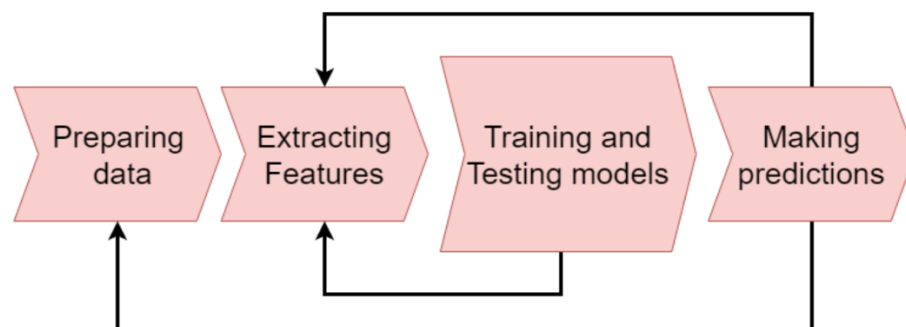


รูปที่ 11 เป็นการแสดงบริบทที่คล้ายกันจัดอยู่ในกลุ่มเดียวกัน [6]

## 7.5 Supervised learning on text

### 7.5.1 Supervised learning

การดูแลเครื่องการเรียนรู้งานแบ่งออกเป็นสองส่วน ตามรูปแบบของป้ายกำกับ (เรียกอีกอย่างว่าเป้าหมาย) หากเป้าหมายคือการจำแนกค่า (cat/dog) แสดงว่าเป็นปัญหาการจัดหมวดหมู่ ในทางกลับกัน หากเป้าหมายเป็นตัวเลข (ราคาของบ้าน) แสดงว่าปัญหาการถดถอย เมื่อต้องจัดการกับข้อความ ปัญหาที่ตามมาจะเป็นการจำแนกประเภท



## รูปที่ 12 workflow supervised [6]

จากรูปด้านบนแสดง workflow ทั่วไปของระบบการจัดการประเภทข้อความ เราเริ่มต้นด้วยการแบ่งข้อมูลออกเป็นชุดฝึกอบรม และชุดทดสอบ ข้อมูลชุดฝึกและข้อมูลทดสอบต้องได้รับการประมวลผลล่วงหน้าและทำให้เป็นมาตรฐาน หลังจากนั้นจึงจะสามารถดึงคุณลักษณะออกมาได้ เทคนิคการแยกคุณลักษณะยอดนิยมสำหรับข้อมูลประเภทข้อความครอบคลุมอยู่ในส่วนก่อนหน้านี้ เมื่อข้อมูลข้อความถูกแปลงเป็นรูปแบบตัวเลขแล้ว สามารถใช้อัลกอริธึมการเรียนรู้ของเครื่องได้ กระบวนการนี้เรียกว่าฝึกโมเดล โมเดลเรียนรู้รูปแบบจากคุณสมบัติต่าง ๆ เพื่อนำมาทำนายผล โมเดลสามารถปรับให้เหมาะสมเพื่อประสิทธิภาพที่ดีขึ้นโดยใช้ พารามิเตอร์โมเดลผ่านกระบวนการที่เรียกว่าการปรับแต่งไฮเปอร์พารามิเตอร์ แบบจะลองผลลัพธ์ถูกประเมินบนข้อมูลการทดสอบที่มองไม่เห็นก่อนหน้านี้ ประสิทธิภาพของโมเดลวัดโดยใช้เมตริกต่าง ๆ เช่นความแม่นยำ การเรียกคืนคะแนนF1และอื่นๆ อัลกอริธึมที่ใช้สำหรับการจัดประเภทข้อความเช่น

2.5.1.1 Multinomial Naive Bayes อยู่ในตระกูลของอัลกอริธึม Naïve Bayes ซึ่งสร้างขึ้นจากการใช้ทฤษฎี ของ Bayes โดยใช้สมมติฐานที่มีป้ายกำกับต่างกันมากกว่าสองป้ายที่แตกต่างกัน

2.5.1.2 Logistic Regression เป็นอัลกอริธึมที่ใช้ฟังก์ชัน Simoid เพื่อทำนายค่าการจำแนก แพคเกจซอฟต์แวร์ที่ได้รับความนิยม SKLearn จะอนุญาตให้ปรับพารามิเตอร์ของโมเดลในลักษณะที่อัลกอริธึมสามารถใช้สำหรับการจำแนกประเภทหลายป้ายกำกับได้เช่นกัน

2.5.1.3 Support Vector Machines (SVM) อัลกอริธึมที่ใช้เส้นหรือไฮเปอร์เพลน (ในกรณีที่มีคุณสมบัติมากกว่าสองอย่าง จึงสร้างพื้นที่หลายมิติ) เพื่อแยกคลาส

2.5.1.4 Random Forest การบูรณาการวิธีการฝึกต้นไม้ตัดสินใจหลายขนานในเซตย่อยข้อมูลที่แตกต่างกัน

2.5.1.5 Gradient Boosting Machine (GBM) ชุดของวิธีการแบบบูรณาการที่ใช้ในการฝึกชุดของผู้เรียนที่อ่อนแอเช่นต้นไม้การตัดสินใจที่จะได้รับผลลัพธ์ที่ถูกต้อง XGboost เป็นหนึ่งในการทำงานที่นิยมมากที่สุดของชุดนี้

Random Forest และ Random Forest เป็นอัลกอริธึมการจัดหมวดหมู่เป็นวิธีการแบบบูรณาการซึ่งใช้ขั้นตอนวิธีการพยากรณ์หลายเพื่อให้บรรลุทั่วไปดีกว่า ผลของวิธีการตั้งค่านั้นจะ

เฉลี่ยมากกว่ารุ่นเดียวและมีประสิทธิภาพมากขึ้นในชุดข้อมูลที่มีขนาดใหญ่กว่า อย่างไรก็ตามเป็น sarkar ได้พิสูจน์แล้วว่าใน วิธีการบูรณาการไม่จำเป็นต้องจัดการกับข้อมูลข้อความดีกว่า

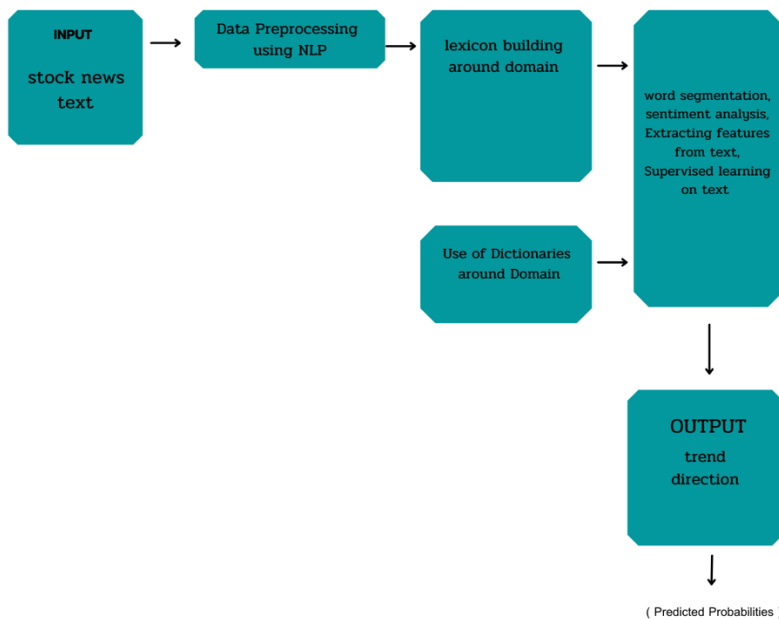


### บทที่ 3

#### รายละเอียดการดำเนินงาน

#### 1. ขั้นตอนและวิธีการดำเนินงาน

##### 1.1 ขั้นตอนการทำงานของระบบ



รูปที่ 13 เป็นการแสดงขั้นตอนการทำงานของ NLP

#### 2.1.1 ขั้นตอน INPUT (แหล่งข้อมูลข่าวสาร)

##### 9.1.1.1 ดังรูปที่ 1

#### 2.1.2 ขั้นตอน Processing

##### 9.1.2.1 การแยกตัวย่อและคำจำกัดความ

##### 9.1.2.2 แยกหน่วยงาน (เช่น คน บริษัท ผลิตภัณฑ์ จำนวนเงิน สถานที่ ฯลฯ)

##### 9.1.2.3 ดึงข้อมูลอ้างอิงไปยังเอกสารอื่น ๆ

##### 9.1.2.4 การแยกอารมณ์ความรู้สึก (ข่าวเชิงบวก/เชิงลบและการอ้างอิง)

##### 9.1.2.5 ดึงคำพูดจากบุคคลที่มีการอ้างอิงถึงผู้เขียน

##### 9.1.2.6 สกัดเงื่อนไขสัญญา

#### 2.1.3 ขั้นตอน Output

9.3.1.1 สรุปข้อความ

9.3.1.2 คาดการณ์ล่วงหน้า

9.3.1.3 แสดงทิศทางแนวโน้มราคาขึ้นหรือลง

9.3.1.3 แสดงขึ้นบนเว็บไซต์

## 2.1 ขั้นตอนการดำเนินงาน

2.1.1 พัฒนาอัลกอริทึม

2.1.2 พัฒนาเขียนโปรแกรมด้วย Python

2.1.3 เทรนสมองกล

2.1.4 ทดสอบระบบ

2.1.5 ทดลองนำไปใช้จริง กับบัญชี Forex demo

2.1.6 ปรับปรุงแก้ไขอุปกรณ์ให้สมบูรณ์

2.1.7 จัดทำเว็บไซต์

2.1.8 จัดทำรายงานให้สมบูรณ์

## บทที่ 4

### ความก้าวหน้าการดำเนินงาน

#### 4.1. ความก้าวหน้า 1 ศึกษา Library Python NLP และทดสอบ

##### 4.1.1. Library Python NLP

###### 4.1.1.1. NLTK Natural Language Toolkit

อินเทอร์เฟซที่ใช้งาน

1. Tokenisation
2. Stemming
3. Tagging
4. Parsing
5. Semantic reasoning
6. Wrappers for industrial-strength NLP libraries

WordNet

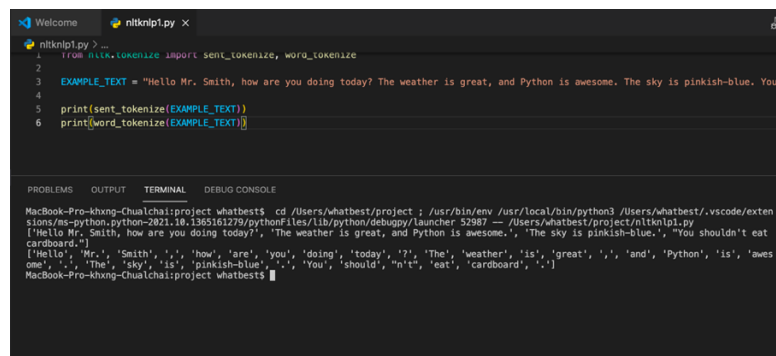
###### 4.1.1.2. TextBlob คือเป็น Library ของภาษา Python

อินเทอร์เฟซที่ใช้งาน

1. Tagging
2. noun phrase extraction
3. sentiment analysis
4. classification
5. language translation
6. word inflection, parsing
7. n-grams
8. WordNet integration.

##### 4.1.2. รายละเอียดการทดลอง

###### 4.1.2.1. Tokenizing Words and Sentences with NLTK



```
nlknlp1.py x
nlknlp1.py > ...
1 from nltk.tokenize import sent_tokenize, word_tokenize
2
3 EXAMPLE_TEXT = "Hello Mr. Smith, how are you doing today? The weather is great, and Python is awesome. The sky is pinkish-blue. You
4
5 print(sent_tokenize(EXAMPLE_TEXT))
6 print(word_tokenize(EXAMPLE_TEXT))

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
MacBook-Pro-khong-Chualchai:project whatbest$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extension
ns/rs-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 52987 -- /Users/whatbest/project/nlknlp1.py
['Hello Mr. Smith, how are you doing today?', 'The weather is great, and Python is awesome.', 'The sky is pinkish-blue.', 'You shouldn't eat
cardboard.']
['Hello', 'Mr.', 'Smith', ',', 'how', 'are', 'you', 'doing', 'today', '?', 'The', 'weather', 'is', 'great', ',', 'and', 'Python', 'is', 'awes
ome', ',', 'The', 'sky', 'is', 'pinkish-blue', '.', 'You', 'should', 'n't', 'eat', 'cardboard', '.']
MacBook-Pro-khong-Chualchai:project whatbest$
```

Sentences \_Tokenizing – การแยกประโยคและคำออกจากเนื้อความของข้อความ

Words \_Tokenizing เป็นการแยกคำออกมาจากประโยค

#### 4.1.2.2. Stop words with NLTK

```
stopwords.py > ...
1 from nltk.corpus import stopwords
2 from nltk.tokenize import word_tokenize
3
4 example_sent = "This is a sample sentence, showing off the stop words filtration."
5
6 stop_words = set(stopwords.words('english'))
7
8 word_tokens = word_tokenize(example_sent)
9
10 filtered_sentence = [w for w in word_tokens if not w in stop_words]
11
12 filtered_sentence = []
13
14 for w in word_tokens:
15     if w not in stop_words:
16         filtered_sentence.append(w)
17
18 print("stop_words = ",stop_words)
19 print("word_tokens = ",word_tokens)
20 print("filtered_sentence = ",filtered_sentence)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

MacBook-Pro-khng-Chualchai:project whatbest\$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 53097 -- /Users/whatbest/project/stopwords.py

เป็นการใช้ คำหยุด โดยอ้างอิงจากคลังข้อมูลของภาษาอังกฤษ คำหยุด คือ คำที่เป็นคำวิเศษณ์หรือเป็นคำที่เสริมทำให้ประโยคดีขึ้น แต่ในเชิงคอมพิวเตอร์ คำหยุด คือคำที่ไม่มีความหมาย เหตุผลที่ต้องลบคำหยุดเนื่องจากประโยคที่ยาวขึ้นจะกินทรัพยากรและเวลาในการประมวลผล

## 4.2. ความก้าวหน้า 2 ทดสอบ Library NLTK

### 4.2.1. รายละเอียดการทดลอง

#### 4.2.1.1. Stemming words with NLTK

```
stem.py > ...
1 from nltk.stem import PorterStemmer
2 from nltk.tokenize import sent_tokenize, word_tokenize
3
4 ps = PorterStemmer()
5
6 example_words = ["python", "pythoner", "pythoning", "pythoned", "pythonly", "feeling"]
7
8 for w in example_words:
9     print(ps.stem(w))
10
11 new_text = "It is important to by very pythonly while you are pythoning with python. All pythoners have pythoned poorly at least onc"
12 words = word_tokenize(new_text)
13
14 for w in words:
15     print(ps.stem(w))
16
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

MacBook-Pro-khng-Chualchai:project whatbest\$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 53446 -- /Users/whatbest/project/stem.py

Stemming หรือที่เรียกว่า suffix stripping เป็นเทคนิคที่ใช้ในการลดขนาดข้อความ ต้นกำเนิดยังเป็นประเภทของข้อความที่ทำให้เป็นมาตรฐานที่ช่วยให้สร้างมาตรฐานของคำบางคำให้เป็นนิพจน์เฉพาะ ข้อเสีย ในไฟฟ้ไล่นการทำให้มืองข้อความหลายๆ แบบ มีตัวเลือกมากมายที่เกี่ยวข้องซึ่งอาจทำให้ข้อมูลสูญหายได้

#### 4.2.1.2. Part of Speech Tagging with NLTK

```
part of speech.py >...
1 import nltk
2 from nltk.corpus import state_union
3
4 from nltk.tokenize import PunktSentenceTokenizer
5 train_text = state_union.raw("2005-GWBush.txt")
6 sample_text = state_union.raw("2006-GWBush.txt")
7
8 custom_sent_tokenizer = PunktSentenceTokenizer(train_text)
9 tokenized = custom_sent_tokenizer.tokenize(sample_text)
10
11 def process_content(i):
12     try:
13         for i in tokenized[i:]:
14             words = nltk.word_tokenize(i)
15             tagged = nltk.pos_tag(words)
16             print(tagged)
17     except Exception as e:
18         print(str(e))
19
20 process_content(i)
```

MacBook-Pro-kheng-Chulalai:project whatbest\$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonfiles/lib/python/debugpy/launcher 57279 -- /Users/whatbest/project/part\_of\_speech.py

```
[(('PRESIDENT', 'NNP'), ('CONGRESS', 'NNP'), ('he', 'NP'), ('supreme', 'NNP'), ('the', 'POS'), ('ADDRESS', 'NNP'), ('before', 'IN'), ('a', 'NP'), ('JOINT', 'NNP'), ('SESSION', 'NNP'), ('of', 'IN'), ('the', 'NP'), ('the', 'NP'), ('CONGRESS', 'NNP'), ('ON', 'NNP'), ('the', 'NP'), ('STATE', 'NNP'), ('TO', 'IN'), ('the', 'NP'), ('UNION', 'NNP'), ('January', 'NNP'), ('31', 'CD'), ('of', 'IN'), ('the', 'NP'), ('2006', 'CD'), ('the', 'NP'), ('PRESIDENT', 'NNP'), ('I', 'PRP'), ('Thank', 'NNP'), ('you', 'NP'), ('all', 'DT'), ('for', 'IN'))]
[(('Mr.', 'NNP'), ('Speaker', 'NNP'), ('I', 'PRP'), ('Vice', 'NNP'), ('President', 'NNP'), ('Cheney', 'NNP'), ('I', 'PRP'), ('members', 'NNS'), ('of', 'IN'), ('the', 'NP'), ('Congress', 'NNP'), ('I', 'PRP'), ('members', 'NNS'), ('of', 'IN'), ('the', 'DT'), ('Supreme', 'NNP'), ('Court', 'NNP'), ('and', 'CC'), ('diplomatic', 'JJ'), ('corps', 'NN'), ('I', 'PRP'), ('distinguished', 'JJ'), ('guests', 'NNS'), ('I', 'PRP'), ('and', 'CC'), ('fellow', 'JJ'), ('citizens', 'NNS'), ('I', 'PRP'), ('Today', 'NP'), ('I', 'PRP'), ('want', 'VBP'), ('to', 'IN'), ('call', 'VBD'), ('to', 'IN'), ('the', 'NP'), ('beloved', 'VBN'), ('to', 'IN'), ('the', 'NP'), ('graceful', 'JJ'), ('I', 'PRP'), ('courageous', 'JJ'), ('woman', 'NN'), ('who', 'NP'), ('called', 'VBD'), ('America', 'NNP'), ('to', 'IN'), ('its', 'NP'), ('founding', 'NN'), ('ideals', 'NNS'), ('and', 'CC'), ('carried', 'VBD'), ('on', 'IN'), ('a', 'DT'), ('noble', 'JJ'), ('dream', 'NN'), ('I', 'PRP'))]
[(('Tonight', 'NNP'), ('we', 'PRP'), ('are', 'VBP'), ('comforted', 'VBN'), ('by', 'IN'), ('the', 'DT'), ('hope', 'NN'), ('of', 'IN'), ('a', 'NP'), ('DT'), ('glad', 'JJ'), ('freedom', 'NN'), ('with', 'IN'), ('the', 'DT'), ('band', 'NN'), ('who', 'NP'), ('we', 'VBD'), ('taken', 'VBN'), ('as', 'NP'), ('long', 'RB'), ('ago', 'RB'), ('I', 'PRP'), ('and', 'CC'), ('we', 'PRP'), ('are', 'VBP'), ('grateful', 'JJ'), ('for', 'IN'), ('the', 'DT'), ('good', 'JJ'), ('life'), ('of', 'IN'), ('Coretta', 'NNP'), ('Scott', 'NNP'), ('King', 'NNP'), ('I', 'PRP'))]
[(('I', 'PRP'), ('Applause', 'NNP'), ('I', 'PRP'), ('I', 'PRP'))]
[(('President', 'NNP'), ('George', 'NNP'), ('W.', 'NNP'), ('Bush', 'NNP'), ('receives', 'VBZ'), ('to', 'TO'), ('to', 'TO'), ('Applause', 'VB'), ('during', 'IN'), ('this', 'NP'), ('State', 'NNP'), ('of', 'IN'), ('the', 'DT'), ('Union', 'NNP'), ('Address', 'NNP'), ('at', 'IN'), ('the', 'DT'), ('Capitol', 'NNP'), ('I', 'PRP'), ('Tuesday', 'NNP'), ('I', 'PRP'), ('Jan', 'NNP'), ('I', 'PRP'))]
```

POS Tagging (ส่วนหนึ่งของการแท็กคำพูด) เป็นกระบวนการในการทำเครื่องหมายคำในรูปแบบข้อความสำหรับส่วนใดส่วนหนึ่งของคำพูดตามคำจำกัดความและบริบท มีหน้าที่รับผิดชอบในการอ่านข้อความในภาษาและกำหนดโทเค็นเฉพาะ (Parts of Speech) ให้กับแต่ละคำ เรียกอีกอย่างว่าการติดแท็กทางไวยากรณ์

ตัวอย่างแท็ก NLTK POS มีดังนี้:

Abbreviation	Meaning
CC	coordinating conjunction
CD	cardinal digit
DT	determiner
EX	existential there
FW	foreign word
IN	preposition/subordinating conjunction
JJ	This NLTK POS Tag is an adjective (large)
JJR	adjective, comparative (larger)
JJS	adjective, superlative (largest)
LS	list marker
MD	modal (could, will)
NN	noun, singular (cat, tree)
NNS	noun plural (desks)
NNP	proper noun, singular (sarah)
NNPS	proper noun, plural (indians or americans)
PDT	predeterminer (all, both, half)
POS	possessive ending (parent\ 's)
PRP	personal pronoun (hers, herself, him, himself)
PRP\$	possessive pronoun (her, his, mine, my, our )
RB	adverb (occasionally, swiftly)
RBR	adverb, comparative (greater)
RBS	adverb, superlative (biggest)
RP	particle (about)
TO	infinite marker (to)
UH	interjection (goodbye)
VB	verb (ask)
VBG	verb gerund (judging)
VBD	verb past tense (pleaded)
VCN	verb past participle (reunified)
VBP	verb, present tense not 3rd person singular(wrap)

VBZ	verb, present tense with 3rd person singular (bases)
WDT	wh-determiner (that, what)
WP	wh- pronoun (who)
WRB	wh- adverb (how)

#### 4.2.1.3. Lemmatizing with NLTK

```

Lemmatizing.py > ...
1  from nltk.stem import WordNetLemmatizer
2
3  lemmatizer = WordNetLemmatizer()
4
5  print(lemmatizer.lemmatize("cats"))
6  print(lemmatizer.lemmatize("cacti"))
7  print(lemmatizer.lemmatize("geese"))
8  print(lemmatizer.lemmatize("rocks"))
9  print(lemmatizer.lemmatize("python"))
10 print(lemmatizer.lemmatize("better", pos="a"))
11 print(lemmatizer.lemmatize("best", pos="a"))
12 print(lemmatizer.lemmatize("run"))
13 print(lemmatizer.lemmatize("run", 'v'))

```

```

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
MacBook-Pro-khxng-Chualchai:project whatbest$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extension/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 57403 -- /Users/whatbest/project/Lemmatizing.py
cat
cactus
goose
rock
python
good
best
run
run
MacBook-Pro-khxng-Chualchai:project whatbest$

```

Lemmatization เป็นกระบวนการของการจัดกลุ่มคำในรูปแบบผันแปรต่างๆ เพื่อให้สามารถวิเคราะห์เป็นรายการเดียวได้ Lemmatization คล้ายกับการกำเนิด แต่นำบริบทมาสู่คำ ดังนั้นจึงเชื่อมโยงคำที่มีความหมายคล้ายกันเป็นคำเดียว

### 4.3. ความก้าวหน้า 3 ทดสอบ Library NLTK

#### 4.3.1. รายละเอียดการทดลอง

##### 4.3.1.1. Wordnet with NLTK

```

Wordnet with NLTK.py > ...
1  from nltk.corpus import wordnet
2  syns = wordnet.synsets("program")
3  print(syns[0].name())
4  print(syns[0].lemmas()[0].name())
5  print(syns[0].definition())
6  print(syns[0].examples())
7  synonyms = []
8  antonyms = []
9
10 for syn in wordnet.synsets("good"):
11     for l in syn.lemmas():
12         synonyms.append(l.name())
13         if l.antonyms():
14             antonyms.append(l.antonyms()[0].name())
15
16 print(set(synonyms))
17 print(set(antonyms))

```

```

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
MacBook-Pro-khxng-Chualchai:project whatbest$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extension/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 57441 -- "/Users/whatbest/project/Wordnet with NLTK.py"
plan.n.01
plan
a series of steps to be carried out or goals to be accomplished
['they drew up a six-step plan', 'they discussed plans for a new bond issue']
{'near', 'effective', 'practiced', 'proficient', 'unspoiled', 'respectable', 'skilful', 'secure', 'unspoilt', 'estimable', 'beneficial', 'goodness', 'upright', 'right', 'in_effect', 'expert', 'honorable', 'dependable', 'ripe', 'serious', 'soundly', 'just', 'undecomposed', 'safe', 'full', 'dear', 'in_force', 'thoroughly', 'adept', 'sound', 'salutary', 'skillful', 'trade_good', 'commodity', 'good', 'well', 'honest'}
{'evil', 'bad', 'evilness', 'badness', 'ill'}
MacBook-Pro-khxng-Chualchai:project whatbest$

```

WordNetเป็นฐานข้อมูลคำศัพท์สำหรับภาษาอังกฤษ ซึ่งสร้างโดย Princeton และเป็นส่วนหนึ่งของคลังข้อมูล NLTK

### 4.3.1.2. Text Summarization with NLTK in Python

main ▾Project / summarize / Auto-Summarize an article .py / <> Jump to ▾Go to file...

Chualchai Apichatitiworn commitLatest commit 44eb6a6 28 minutes ago🕒 History

0 contributors

51 lines (42 sloc) | 1.67 KB

RawBlame🔍📄🗑️

```
1 import bs4 as bs
2 import urllib.request
3 import re
4 import nltk
5
6 scraped_data = urllib.request.urlopen('https://www.eia.gov/petroleum/weekly/')
7 article = scraped_data.read()
8
9 parsed_article = bs.BeautifulSoup(article, 'lxml')
10 paragraphs = parsed_article.find_all('p')
11
12 article_text = ""
13
14
15 for p in paragraphs:
16     article_text += p.text
17 # Removing Square Brackets and Extra Spaces
18 article_text = re.sub(r'\[[0-9]*\]', ' ', article_text)
19 article_text = re.sub(r'\s+', ' ', article_text)
20 # Removing special characters and digits
21 formatted_article_text = re.sub('[^a-zA-Z]', ' ', article_text)
22 formatted_article_text = re.sub(r'\s+', ' ', formatted_article_text)
23 sentence_list = nltk.sent_tokenize(article_text)
24 stopwords = nltk.corpus.stopwords.words('english')
25
26 word_frequencies = {}
27 for word in nltk.word_tokenize(formatted_article_text):
28     if word not in stopwords:
29         if word not in word_frequencies.keys():
30             word_frequencies[word] = 1
31         else:
32             word_frequencies[word] += 1
33 maximum_frequency = max(word_frequencies.values())
34
35 for word in word_frequencies.keys():
36     word_frequencies[word] = (word_frequencies[word]/maximum_frequency)
37 sentence_scores = {}
38 for sent in sentence_list:
39     for word in nltk.word_tokenize(sent.lower()):
40         if word in word_frequencies.keys():
41             if len(sent.split(' ')) < 30:
42                 if sent not in sentence_scores.keys():
43                     sentence_scores[sent] = word_frequencies[word]
44                 else:
45                     sentence_scores[sent] += word_frequencies[word]
46 import heapq
47 summary_sentences = heapq.nlargest(7, sentence_scores, key=sentence_scores.get)
48
49 summary = ' '.join(summary_sentences)
50
51 print(summary)
```

PROBLEMSOUTPUTTERMINALDEBUG CONSOLE

MacBook-Pro-khxng-Chualchai:Project whatbest\$ cd /Users/whatbest/Documents/GitHub/Project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 60443 -- "/Users/whatbest/Documents/GitHub/Project/summarize/Auto-Summarize an article .py"  
In 2021, we estimate the average Eagle Ford formation well continued to produce more oil at 21,900 barrels in its first month (Figure 3). We developed these production decline curves for Eagle Ford formation wells from approximately 750 sub-county areas, called grids, which are approximately 14 square miles. This is based on the known number of wells already drilled in a grid, their past decline profile, and developing all future potential well sites. Virtually all of this production has occurred in 16 of the 30 counties and within a producing subset of that total area of approximately 7.2 million acres. Without a price capable of providing a return on investment, producers will not invest capital in drilling a well. A substantially larger amount of the area becomes more profitable as a result of higher prices in 2022. However, not all possible acreage will be developed because of future surface infrastructure considerations or leased acreage that is unavailable for development.  
MacBook-Pro-khxng-Chualchai:Project whatbest\$



ในสคริปต์ด้านบน นำเข้าไลบรารีที่สำคัญที่จำเป็นสำหรับการดึงข้อมูลจากเว็บ ก่อน จากนั้นใช้urlopenฟังก์ชันจากurllib.requestยูทิลิตี้เพื่อขูดข้อมูล ต่อไป ต้องเรียกreadใช้ฟังก์ชันบนวัตถุที่ส่งคืนโดยurlopenฟังก์ชันเพื่ออ่านข้อมูล ในการแยกวิเคราะห์ข้อมูล ใช้BeautifulSoupอ็อบเจกต์และส่งผ่านอ็อบเจกต์ข้อมูลที่คัดลอกมา เช่นarticleและxmlตัวแยกวิเคราะห์ article\_textมีข้อความโดยไม่มีวงเล็บ อย่างไรก็ตาม จะไม่ลบอะไรไปจากบทความเนื่องจากเป็นบทความต้นฉบับ จะไม่ลบตัวเลข เครื่องหมายวรรคตอน และสัญลักษณ์พิเศษอื่นๆ ออกจากข้อความนี้ เนื่องจากเราจะใช้ข้อความนี้เพื่อสร้างบทสรุปและความถี่ของคำแบบถ่วงน้ำหนัก มีสองอ็อบเจกต์article\_textซึ่งประกอบด้วยบทความต้นฉบับและบทความformatted\_article\_textที่จัดรูปแบบ จะใช้formatted\_article\_textเพื่อสร้างฮิสโทแกรมความถี่ถ่วงน้ำหนักสำหรับคำนั้นๆ และจะแทนที่ความถี่ถ่วงน้ำหนักเหล่านี้ด้วยค่าในarticle\_textอ็อบเจกต์

**Converting Text To Sentences** ได้ทำการประมวลผลข้อมูลถ่วงน้ำหนักแล้ว ต่อไป ต้องแปลงบทความให้เป็นประโยค จะใช้article\_textวัตถุสำหรับ tokenizing บทความเป็นประโยคเนื่องจากมีการหยุดเต็ม formatted\_article\_textไม่มีเครื่องหมายวรรคตอนใดๆ ดังนั้นจึงไม่สามารถแปลงเป็นประโยคโดยใช้จุดเต็มเป็นพารามิเตอร์ได้

**Converting Text To Sentences** ในการหาความถี่ของการเกิดขึ้นของแต่ละคำ ใช้formatted\_article\_textตัวแปร ใช้ตัวแปรนี้เพื่อค้นหาความถี่ของการเกิดเนื่องจากไม่มีเครื่องหมายวรรคตอน ตัวเลข หรืออักขระพิเศษอื่นๆ ขั้นแรกจะเก็บคำหยุดภาษาอังกฤษทั้งหมดจากnlkไลบรารีลงในstopwordsตัวแปร ต่อไป จะวนรอบประโยคทั้งหมดแล้วตามด้วยคำที่เกี่ยวข้องเพื่อตรวจสอบก่อนว่าเป็นคำหยุดหรือไม่ หากไม่เป็นเช่นนั้น จะดำเนินการตรวจสอบว่าคำนั้นมีอยู่ในword\_frequencyพจนานุกรมหรือไม่ เช่นword\_frequenciesหรือไม่ หากพบคำนี้เป็นครั้งแรก คำนั้นจะถูกเพิ่มลงในพจนานุกรมเป็นคีย์และตั้งค่าเป็น 1 มิฉะนั้น หากคำนั้นมีอยู่ในพจนานุกรมก่อนหน้านี้ ค่าของคำนั้นจะถูกอัปเดตเพียง 1

**Calculating Sentence Scores** ตอนนี้ได้คำนวณความถี่ถ่วงน้ำหนักสำหรับคำทั้งหมดแล้ว ตอนนี้เป็นเวลาที่จะคำนวณคะแนนสำหรับแต่ละประโยคโดยการเพิ่มความถี่ถ่วงน้ำหนักของคำที่เกิดขึ้นในประโยคนั้นโดยเฉพาะ ขั้นแรกจะสร้างsentence\_scoresพจนานุกรม เปล่า กฎของพจนานุกรมนี้จะเป็นตัวประโยคเอง และค่าจะเป็นคะแนนที่สอดคล้องกันของประโยค ต่อไป จะวนรอบแต่ละประโยคใน the sentence\_listและแปลงประโยคเป็นคำ

จากนั้นจะตรวจสอบว่าคำนั้นมีอยู่ในword\_frequenciesพจนานุกรมหรือไม่ การตรวจสอบนี้ดำเนินการเนื่องจากสร้างsentence\_listรายการจากarticle\_textวัตถุ ในทางกลับกัน ความถี่ของคำถูกคำนวณโดยใช้formatted\_article\_textออบเจกต์ ซึ่งไม่มีคำหยุด ตัวเลข ฯลฯ

ไม่ต้องการประโยคที่ยาวมากในการสรุป ดังนั้นจึงคำนวณคะแนนสำหรับประโยคที่มีค่าน้อยกว่า 30 คำเท่านั้น (แม้ว่าจะปรับแต่งพารามิเตอร์นี้สำหรับกรณีการใช้งานของตนเองได้ก็ตาม) ต่อไปจะตรวจสอบว่าประโยคนั้นมีอยู่ในsentence\_scoresพจนานุกรมหรือไม่ หากไม่มีประโยคดังกล่าว จะเพิ่มลงในsentence\_scoresพจนานุกรมเป็นคีย์และกำหนดความถี่ถ่วงน้ำหนักของคำแรกในประโยคเป็นค่าของประโยค ในทางตรงกันข้าม หากประโยคนั้นมีอยู่ในพจนานุกรม เพียงแค่เพิ่มความถี่ถ่วงน้ำหนักของคำนั้นให้กับค่าที่มีอยู่

**Getting the Summary** ตอนนี้มีsentence\_scoresพจนานุกรมที่มีประโยคที่มีคะแนนตรงกัน เพื่อสรุปบทความสามารถใช้ประโยค N อันดับแรกที่มีคะแนนสูงสุด สคริปต์ต่อไปนี้ดึงประโยค 7 อันดับแรกใช้ไลบรารีheapqและเรียกใช้nlargestฟังก์ชันเพื่อดึงประโยค 7 อันดับแรกที่มีคะแนนสูงสุด

#### 4.3.1.3.Sentiment Analysis

```
textbox > sentimentddd.py > ...
1 from textblob import TextBlob
2
3 # Preparing an input sentence
4 sentence = 'The platform provides universal access to the world's best education, partnering with top universities and organizations to help millions of students succeed.'
5
6 analysisPol = TextBlob(sentence).polarity
7 analysisSub = TextBlob(sentence).subjectivity
8
9 print(analysisPol)
10 print(analysisSub)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

MacBook-Pro-khxng-Chualchai:Project whatbest\$ cd /Users/whatbest/Documents/GitHub/Project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 61369 -- /Users/whatbest/Documents/GitHub/Project/textbox/sentimentddd.py  
0.5  
0.26666666666666666  
MacBook-Pro-khxng-Chualchai:Project whatbest\$

อัตราส่วนของ TextBlob สำหรับงานวิเคราะห์การวิเคราะห์จะภายในช่วง[-1.0, 1.0] ที่-1.0เป็นขั้วลบและ1.0เป็นบวก คะแนนนี้ยังสามารถเท่ากับ0ซึ่งหมายถึงการประเมินที่เป็นกลางของคำ เนื่องจากไม่มีคำใด ๆ จากชุดการฝึก

ในขณะที่งานการ ระบุ อัตนัย / ความเป็นวัตถุรายงานการลอยตัวภายในช่วง[0.0, 1.0]ที่0.0เป็นประโยคที่เป็นกลางและ1.0เป็นอัตนัยมาก

เมื่อนำเข้าแล้ว เราจะโหลดประโยคเพื่อวิเคราะห์และสร้างอินสแตนซ์ของTextBlobวัตถุ รวมทั้งกำหนด sentimentคุณสมบัติให้กับของเราเองanalysis:

คุณสมบัติsentimentเป็น a ของ แบบnamedtupleฟอร์มSentiment(polarity, subjectivity)

ผลลัพธ์ที่คาดหวังของการวิเคราะห์คือ:

สิ่งที่เชื่อมต่ออย่างหนึ่งเกี่ยวกับ TextBlob คือช่วยให้ผู้ใช้สามารถเลือกอัลกอริทึมสำหรับการใช้งาน NLP ระดับสูงได้:

PatternAnalyzer- ตัวแยกประเภทเริ่มต้นที่สร้างขึ้นบนไลบรารีรูปแบบ

NaiveBayesAnalyzer- โมเดล NLTK ที่ได้รับการฝึกอบรมเกี่ยวกับคลังบทวิจารณ์ภาพยนตร์

## บทที่ 5

### สรุป

#### 5.1. สรุปผลการดำเนินงาน

1. สามารถสรุปบทความได้ ใช้ Library NLTK NLP
2. สามารถวิเคราะห์ บทความได้ระดับเบื้องต้น

#### 5.2. ปัญหาและอุปสรรค

1. ยังใช้ภาษา Python ได้ไม่ชำนาญ
2. Library ที่จะใช้ค่อนข้างเยอะ จึงทำให้เริ่ม Project ช้า
3. มีปัญหาเรื่อง Version ที่จะติดตั้ง Library

#### 5.3. งานที่จะดำเนินการต่อไป

1. ปรับปรุงแก้ไขการทำ sentiment Analysis
2. เทรนข้อมูลที่เกี่ยวข้องกับ Forex
3. ทดสอบ Algorithm ระหว่าง SpaCy กับ NLTK

### บรรณานุกรม

[1] S. Shalev-Shwartz, S. Ben-David, [Understanding Machine Learning: From Theory to Algorithms](#) (2014), Cambridge University Press , เข้าถึงล่าสุด 15 มกราคม 2565

[2] T. Mikolov, I. Sutskever, K. Chen, G. S Corrado, and J. Dean. [Distributed Representations of Words and Phrases and their Compositionality](#) (2013), Advances in Neural Information Processing Systems 26 เข้าถึงล่าสุด 15 มกราคม 2565

[3] J. Pennington, R. Socher, and C. D. Manning, [GloVe: Global Vectors for Word Representation](#) (2014), In EMNLP. เข้าถึงล่าสุด 16 มกราคม 2565

[4] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. [Enriching word vectors with subword information](#) (2016), arXiv preprint เข้าถึงล่าสุด 17 มกราคม 2565

[5] NLP Implementations : URL : <https://towardsdatascience.com/gentle-start-to-natural-language-processing-using-python-6e46c07addf3> เข้าถึงล่าสุด 18 มกราคม 2565

[6] The theory you need to know before you start an NLP : URL : <https://towardsdatascience.com/the-theory-you-need-to-know-before-you-start-an-nlp-project-1890f5bbb793> เข้าถึงล่าสุด 12 มีนาคม 2565

[7] Us Department of labor : URL : <https://www.dol.gov/> เข้าถึงล่าสุด 12 มีนาคม 2565

[8] Energy information Administration : URL : <https://www.eia.gov/> เข้าถึงล่าสุด 12 มีนาคม 2565

[9] กองทุน SPDR : URL : <https://traderider.com/forex/spdr-%E0%B8%81%E0%B8%AD%E0%B8%87%E0%B8%97%E0%B8%B8%E0%B8%99%E0%B8%97%E0%B8%AD%E0%B8%87%E0%B8%84%E0%B8%B3%E0%B9%81%E0%B8%97%E0%B9%88%E0%B8%87> เข้าถึงล่าสุด 12 มีนาคม 2565

[10] Federal Reserve : URL : <https://www.federalreserve.gov/> เข้าถึงล่าสุด 12 มีนาคม 2565

[11] Bloomberg : URL : <https://www.bloomberg.com/asia> เข้าถึงล่าสุด 12 มีนาคม 2565

[12] Twitter : URL : <https://twitter.com/> เข้าถึงล่าสุด 12 มีนาคม 2562