1. GUI Analyze
   1. เพิ่ม mode การแบ่งประโยค : TH หรือ EN
   2. เมื่อกดปุ่ม Analyze แล้ว ยังไม่ล้าง input text box จะได้เช็คค่ากับ result ได้
   3. เพิ่มปุ่มให้เลือกว่าจะเก็บผลลัพธ์ record นี้ หรือไม่ (yes, no) เมื่อกดปุ่มแล้วค่อยล้าง input text box
2. message.extractor.gui.AnalyzeScreenFXMLController.java
   1. เพิ่มตัวแปร

private boolean thSentenceMode = true;

@FXML

private Button yesBtn;

@FXML

private Button noBtn;

@FXML

private RadioButton thRadio;

@FXML

private RadioButton enRadio;

@FXML

private ToggleGroup sentenceToggle;

* 1. private void analyzeBtnHandler(ActionEvent event) {

เพิ่มส่วนที่ handle วิธีแบ่งประโยค

if (!isInFileMode) {

…

msgExtractor.setSentenceMode(thSentenceMode);

arffWriter.setSentenceMode(thSentenceMode);

arffWriterWeighted.setSentenceMode(thSentenceMode);

…

thRadio.setDisable(true);

enRadio.setDisable(true);

// comment lines that clear & reset input text

}

แบบอ่านจาก file ก็ทำเหมือนกัน

* 1. private void saveBtnHandler(ActionEvent event) {

เช็คว่ามี record ให้เขียนหรือไม่ ถ้ามีแค่หัว arff ก็ไม่ต้องเขียน

if (msgExtractor.getNumRecords() == 0) {

Dialogs.showInformationDialog(null, "No data to be saved !!");

return;

}

* 1. ส่วนที่ handle ปุ่ม yes/no และ radio button

@FXML

private void noBtnHandler(ActionEvent event) {

// Clear Input and Result Fields

authorField.setText("");

inputTextArea.setText("");

resultArea.setText("");

}

@FXML

private void yesBtnHandler(ActionEvent event) {

// Save this line to arff

msgExtractor.saveArffRecord();

// Clear Input and Result Fields

authorField.setText("");

inputTextArea.setText("");

resultArea.setText("");

}

@FXML

private void thRadioHandler(ActionEvent event) {

if (thRadio.isSelected()) thSentenceMode = true;

}

@FXML

private void enRadioHandler(ActionEvent event) {

if (enRadio.isSelected()) thSentenceMode = false;

}

1. message.extractor.arff.ArffFileWriter.java
   1. เพิ่มตัวแปร

private boolean thSentenceMode = true;

* 1. เพิ่มเมธอด

public void setSentenceMode(boolean stm) {

thSentenceMode = stm;

}

public int getNumRecords() {

return data.numInstances();

}

1. message.extractor.analyzer.MessageExtractor.java
   1. เพิ่มตัวแปร

private String pendingAuthor = "?";

private Analyzer[] pendingVars;

private boolean thSentenceMode = true;

* 1. เพิ่มเมธอด

public void setSentenceMode(boolean stm) {

thSentenceMode = stm;

}

public int getNumRecords() {

return arffFileWriter.getNumRecords();

}

กดปุ่ม yes แล้วค่อย add record ใส่ arff

public void saveArffRecord() {

if (pendingVars != null) {

CharacterAnalysis C = (CharacterAnalysis)pendingVars[0];

WordAnalysis W = (WordAnalysis)pendingVars[1];

PunctuationAnalysis P = (PunctuationAnalysis)pendingVars[2];

EmotionMarkerAnalysis E = (EmotionMarkerAnalysis)pendingVars[3];

StructureAnalysis S = (StructureAnalysis)pendingVars[4];

ContentAnalysis T = (ContentAnalysis)pendingVars[5];

arffFileWriter.addNewRow(pendingAuthor, C.getVariables(),

W.getVariables(), P.getVariables(),

E.getVariables(), S.getVariables(),

T.getVariables());

arffweightedFileWriter.addNewWeightedRow(pendingAuthor,

C.getVariables(), W.getVariables(),

P.getVariables(), E.getVariables(),

S.getVariables(), T.getVariables());

pendingAuthor = "?";

pendingVars = null;

}

}

* 1. public void extract(String author, String msg) {

//StructureAnalysis S = new StructureAnalysis();

StructureAnalysis S = new StructureAnalysis(thSentenceMode);

…

// Analyze each line of inputs

while ((line = br.readLine()) != null) {

…

while (tokenizer.hasNext()) {

end = tokenizer.next();

// don’t count spaces as words

if ( !line.substring(begin, end).matches("\\s+") )

W.analyze(line.substring(begin, end));

begin = end;

}

}

อ่านครบทุกบรรทัดแล้วค่อยนับจำนวนประโยค

S.analyze();

ยังไม่เพิ่มใน arff รอให้กดปุ่ม yes ค่อยเพิ่ม

//arffFileWriter.addNewRow(…);

//arffweightedFileWriter.addNewWeightedRow(…);

pendingAuthor = author;

pendingVars = new Analyzer[6];

pendingVars[0] = C;

pendingVars[1] = W;

pendingVars[2] = P;

pendingVars[3] = E;

pendingVars[4] = S;

pendingVars[5] = T;

1. message.extractor.analyzer.StructureAnalysis.java
   1. เพิ่มตัวแปร

private String globalInput = "";

private boolean thSentenceMode = true;

* 1. public void analyze(String input) {

S[1].addValue();

if(input.length() == 0)

S[2].addValue();

else

//S[3].addValue(input.split("(\\s){1,}").length);

globalInput = globalInput + input;

}

* 1. เพิ่มเมธอดสำหรับแบ่งประโยค : TH แบ่งที่ช่องว่างกับขึ้นบรรทัดใหม่, EN แบ่งที่ punctuation mark

public void analyze() {

if (thSentenceMode)

S[3].addValue(globalInput.split("(\\s){1,}").length);

else

S[3].addValue(globalInput.split("[!?.:;]+").length);

}