Department of Computer Science
Technical University of Cluj-Napoca

Artificial Intelligence II

Laboratory activity 2015-2016

Name: Ichim Daniel Alexandru
Group: 30232
Email: ichim.daniel.alexandru@gmail.com
Laboratory assistant: Roxana Szabo

Assoc . Prof. dr. eng. Adrian Groza
Adrian.Groza@cs.utcluj.ro

Contents

Chapter 1

Rules and policies

Lab organisation. There are 4 deliverables in total:

Deliverable	Description	Pages	Deadline
Proposal:	Description of your proposed project	(about a page)	W_5
Midway report:	Review of related work, details of the proposed method, and preliminary results if available	4-5 pages.	W_7.
Final report	A full academic paper, including: problem definition and motivation, background and related work, details of the proposed method, details of experiments and results, conclusion and future work, references and appendix	8 pages	W_{13}
Presentation	Present your work to the collegues, instructors	10 slides	W_{14}

1. Laboratory work has 20% from the final grade.

2. The schedulling of your work is listed in Table 1.

3. Before each deadline, you have to send your work (latex documentation/code) via *ssh*, exactly as follows:
   ```
   scp WX_Name_Group_Tool.tar.gz iia@192.168.1.32:
   pass:  IIA2016
   ```
   where the deadline $WX \in \{W_5, W_7, W_{13}, W_{14}\}$

4. Realistic and original scenarious are encouraged. Well known toy problems (salesmen, map coloring, logistic planning, wumpus, sudoku, queens, missionaires and canibals, etc.) do not worth much for your grade. Your scenario should be realistic and should be bussiness oriented. Note that the focus is both on programming and on modelling the reality into a formal representation.

5. *Laptop policy*: you can use your own laptop as long you have Linux. One goal of the laboratory is to increase your competency in Linux. It is **your** task to set static IPs:

IP:	192.168.1.51[1]
MASK:	255.255.255.0
GATEWAY:	192.168.1.2
DNS:	192.168.1.2
PROXY	192.168.1.2:3128

Table 1.1: Lab schedulling.

Activity	**Deadline**
Installing the tool.	W_2
Running and understanding examples attached to each tool.	W_4
Selecting an adequate scenario to be implemented using the tool. *Describing the specifications* of your own scenario. Describing the top level design of your scenario.	W_5
Identifying and describing the knowledge bases (data sets, articles, studies, etc.) planned to be used for realistic modeling of your scenario.	W_5
Implementing your scenario. We try to evaluate how much your solution reflects the reality. This is a quantitative criteria: we evaluate how many aspects from real world have been covered by your solution.	W_{10}
Explointing the expressivity of the tool. We try to evaluate how many capabilies provided by the tool have been enacted within your implementation. This is a calitative criteria: a complex realistic scenario requires more expressivity.	W_{11}
Validating the correctness/efficiency of your solution through graphs and/or experiments.	W_{12}
Comparing your solution with related work. Illustrate the advantages and weak points of your solution (description and code showing adavantages/disatvantages).	W_{13}
Using Latex in your documentation. You have to show some competency on writing documentation in Latex. For instance, you have to employ various latex elements: lists, citations, footnotes, verbatim, maths, code, etc.	W_{14}

6. *Group change policy.* Maximum number of students in a class is 14.

Grading. Grade inflation makes difficult to distinguish between students. It also discourages the best students to do their best. In my quest for "optimal ranking of the students", I do not use the following heuristics:

- "He worked hard at the project". Our society do not like anymore individuals that are *trying*, but individual that *do* stuff. Such heuristic is not admissible in education, except the primary school.

- "I knew he chould do much better". Such a heuristic is not admissible because it does not encourage you to spread yourself.

- 10 means that you did very impressive work or more efficient that I expected or handled a lot of special cases for realistic scenarios.

- 7 means that you: i) constantly worked during classes, ii) you proved competent to use the tool and its expressivity for a realistic scenario, iii) you understood theoretical concepts on which the tool rely on.

- 8,9 mean that your code has enough quantity and the results are significant.
- 5 means that you managed to develop something of your own, functional, with your own piece of code substantially different from the examples available.
- You obtain less than 5 in one of the following situations:

    1. few code written by yourself.
    2. too much similarity with the provided examples.
    3. non-seriosity (i.e. re-current late at classes, playing games, worked for other disciplines, poor/unprofessional documentation of your work, etc[2].

  You got 2 if you present the project but fail to submit the documentation or code. You got 1 if you do not present your project before the deadline. You got 0 for any line of code taken from other parts that appear in section *My own code.* For information on TUCN's regulations on plagiarism do consult the active norms.

  If your grade is 0, 1, or 2, you do not satisfy the preconditions for participating to the written exam. The only possibility to increase your laboratory grade is to take another project in the next year, at the same class, and to make all the steps again.

Plagiarism. Most of you consider plagiarism only a minor form of cheating. This is far from accurate. Plagiarism is passing off the work of others as your own to gain unfair advantage.

During your project presentation and documentation, I must not be left with doubts on which parts of your project are your work or not. Always identify both: 1) who you worked with and 2) where you got your part of the code or solution. You should sign the declaration of originality.

Describe clearly the starting point of your solution. List explicitly any code re-used in your project. List explicitly any help (including debugging help, design discussions) provided by others (including colleagues or teaching assistant). Keep in mind that it is your own project and not the teaching assistant's project. Learning by collaborating does remain an effective method. You can use it, but don't forget to mention any kind of support. The assignment is designed to be individual and to pose you some difficulties (both technological and scientific) that you should identify a working solution by the end of the semester.

Class attendance. We are all grown-ups, when and whether you attend lecture is up to you. However, the exam can include any topic that was covered during class, explained on the board, or which emerged from discussions among participants. Missing lab assignments or midterm leads to minimum grade for that part. You are free to manage your laboratory classes, aiming to submit the project earlier, as long as you meet all the constraints and deadlines. However, it is mandatory to participate at the final public presentation of your project.

[2]Consider non-seriosity as a immutable boolean value that is unconsciously activated in my brain when one of the above conditions occurs for the first time.

Chapter 2

AI projects and tools (W_1)

The objectives for this week are:

1. To identify existing projects for undergraduate level in the AI domain.
2. To get awareness of the effort expected for the laboratory activity by browsing past projects at TUCN and other universities.
3. To get used with the technical instrumentation used during this semester.

An ideal project should be one that demonstrates some creativity, attempts to answer an interesting research question, or offers an interesting AI solution to a real scenario.

2.1 AI undergraduate projects

This section aims to provide you starting ideas on student AI projects. Browse the following resources:

1. Project proposals in AI at Roskilde University, Denmark
2. Machine learning project suggestions at School of Computer Science, Carnegie Mellon University (http://www.cs.cmu.edu/ epxing/Class/10701/project.html)
3. Challenges in artificial intelligence domain at HacckerRank
4. Repository of AI assignments at AI-repository
5. Collection of projects at MIT open coursware for the Knowledge-Based Applications Systems class MITOpenCoursware

The projects are intended to let you look in depth at some area of artificial intelligence that may only be covered briefly in class or AIMA. You can see the AI lab as an opportunity for you to explore an interesting problem of your choice in the context of a real-world scenario. Hence, we encourage you to do some original research in the AI domain that is of interest to you. To give you an idea, about how your fellows work as students, take a look at the examples below:

Here is a list of project ideas:

1. A machine learning approach for identifying patterns at Eurovision musical competition.
2. Performance comparison of AI algorithms in the Wumpus world.
3. A multi-agent system for playing the board game Risk. Assess the relative strength of each player and the strategic value of each country.

4. Intersection situation awareness and normative reasoning.
5. Summarize an article written in wikipedia or other technical text.
6. Checking if a small text is entailed by a larger text.
7. Extracting arguments in a persuasive or legal text.
8. A knowledge-based computer purchasing adviser.
9. Build an ontology from a collection of documents using machine learning.
10. Legal assistant in case of divorces. Splitting marital property between spouse.
11. Advisor for installing a wind turbine.
12. Compliance checking of architectural plans for buildings.
13. Knowledge based system for automobile fault diagnosis .
14. Real Estate Analyzer System.

Projects at other universities:

1. A project submitted by Greg Barish entitled "Approaches to integrating abstractions in graphplan-based planning systems" for the Artificial Intelligence Planning class at University of Southern California. Final Report
2. A project submitted by Victor L. Williamson entitled "What to do With a Patient Who Has Chest Pain?" Final report ()

2.2 AI-related competitions

As part of your laboratory work, we encourage you to participate to various students competitions in AI:

1. Machine learning competitions: Kaggle
2. Ontology development competition: Ontology Competition
3. Competition on computational models of argumentation: ICCMA
4. Competitive programming HacckerRank
5. Student StarCraft AI tournament 2016 SSCAIT
6. Power trading agent competition PTAC
7. Ontology alignment evaluation initiative OAEI2015

Disseminating your work. Presenting your results at student conferences increases your chances to obtain a master scholarship:

1. Computer Science Students Conference at UBB and TUCN: CSSC

2.3 Running latex

Very Brief Introduction to Latex by Radu Slavescu

2.4 Linux support

Brief Synopsis of Linux by Radu Slavescu

Chapter 3

Installing the tool (W_2)

The objectives for this week are:

1. To install the tool on a Linux system
2. To get aware of tool plugins, extensions, and running parameters.

3.1 Downloading GATE

To download GATE point your web browser at http://gate.ac.uk/download/

3.2 Installing and Running GATE

GATE will run anywhere that supports Java 7 or later, including Solaris, Linux, Mac OS X and Windows platforms. This time we installed the tool on Fedora (Linux).

1. Download the Generic installer for any platform. This is a JAR file, so it will require Java to be ran.
2. Go to the location of the downloaded file and open the terminal in the specific folder.
3. Write the following command (replace the gate name with your current version that you downloaded).

   ```
   java -jar gate-8.1-build5169-installer.jar
   ```

4. Follow the installation process within the visual installer, setting the location and some configuration for installing.
5. After installing, in order to run gate you have to go in the installation folder, then open the bin folder. And run the following command in the terminal, within this folder.

   ```
       ./gate.sh
   ```

3.3 Configuring GATE

When GATE Developer is started, or when Gate.init() is called from GATE Embedded, GATE loads various sorts of configuration data stored as XML in files generally called something like gate.xml or .gate.xml. This data holds information such as:

- whether to save settings on exit;
- whether to save session on exit;
- what fonts GATE Developer should use;
- plugins to load at start;
- colours of the annotations;
- locations of files for the file chooser;
- and a lot of other GUI related options;

This type of data is stored at two levels (in order from general to specific):

- the site-wide level, which by default is located the gate.xml file in top level directory of the GATE installation (i.e. the GATE home. This location can be overridden by the Java system property gate.site.config;
- the user level, which lives in the user's HOME directory on UNIX or their profile directory on Windows (note that parts of this file are overwritten when saving user settings). The default location for this file can be overridden by the Java system property gate.user.config.

3.4 Uninstalling GATE

For the uninstaller, you have to run the following command:

```
java -jar uninstaller.jar
```

or just delete the whole of the installation directory (the one containing bin, lib, Uninstaller, etc.). The installer doesn't install anything outside this directory, but for completeness you might also want to delete the settings files GATE creates in your home directory (.gate.xml and .gate.session).

Chapter 4

Running and understanding examples (W_3)

The objectives for this week are:

1. To run and understand the toy examples provided by the tool
2. To identify what realistic problems are adequate for your tool

GATE includes an information extraction system called ANNIE (A Nearly-New Information Extraction System) which is a set of modules comprising a tokenizer, a gazetteer, a sentence splitter, a part of speech tagger, a named entities transducer and a coreference tagger. ANNIE can be used as-is to provide basic information extraction functionality, or provide a starting point for more specific tasks.

4.1 Example 1 - Using Part of Speech (POS) features to extract entities

Let's say that we want to identify a sports location through its context. For example, identifying a stadium location where the word "Stadium" is used in various combinations, i.e. "Emirates Stadium", "Wembley Stadium", and "Ben Hill Griffin Stadium". We will have to consider these generalities to write a generic rule that is applicable in all these contexts.

"The sound of the Millennium Stadium when a crowd is watching a football or rugby match is amazing. If you sit right at the back of a full stadium, you can experience a tidal wave of cheering approach you from the other side. Invisible, but just as infectious.

True positives we are after: Millennium Stadium, Sardar Patel Stadium, Emirates Stadium, Wembley Stadium, Ben Hill Griffin Stadium.

False positive we want to ignore are: Stadium
We can use POS features to achieve the desired results. We know that word "Stadium" can be used in conjunction with one or two words (Emirates or Ben Hill Griffin) that must be in upper initial letters, is represented as Token word (Token.kind== "word") and is a Noun hence Token.category=NNP (Noun singular) or NNPS (Noun Plural).

The jape grammar that achieves required objectives is: B.1

Let's understand how this is achieved. Loop 1 is the whole pattern we are trying to match and consist of pattern 2 and pattern 5.

Loop2 consists of two patterns: patterns 3 and 4. Let's see what it tries to find. Loop3 is looking for any word, with NNP and upperinital or anyword, with NNPS and upperinital this will match Millennium, Sardar, Emirates, Wembley, Ben from our example.

Now lets move on to pattern 4, which is repetition of pattern in 3. However notice the presence of "?" at the end meaning optional. Basically, we want to capture the possibility of having two words before the word "Stadium". Using loop 3 and 4 in loop 2 will match further Sardar Patel, Ben Hill from our examples howeverMillennium , Emirates, Wembley will use ? and escape any matching.

After Loop 2 is run, loop 5 runs, which basically further exploits possibility of another word. Again there is ? to indicate the optional nature of this loop. Hence this time it will match as follows for our running example:

Ben Hill Griffin, while for others it will be optional.

Hence when the outer loop runs it detects following:

Millennium, Sardar Patel, Emirates, Wembley, Ben Hill Griffin

Now the rest of the program is straightforward, we are trying to make sure that there is a Stadium, Circuit, Golf Club word suffix at the end of this.

4.2 Example 2 - 3-stage procedure using the tokeniser, gazetteer and named- entity grammar

An example of a 3-stage procedure using the tokeniser, gazetteer and named- entity grammar. Suppose we wish to recognise the phrase '800,000 US dollars' as an entity of type 'Number', with the feature 'money'.

First of all, we give an example of a grammar rule (and corresponding macros) for money, which would recognise this type of pattern.

```
 Macro: MILLION_BILLION
({Token.string == "m"}|
{Token.string == "million"}|
{Token.string == "b"}|
{Token.string == "billion"}
)

Macro: AMOUNT_NUMBER
({Token.kind == number}
(({Token.string == ","}|
{Token.string == "."})
{Token.kind == number})*
(({SpaceToken.kind == space})?
(MILLION_BILLION)?)
)

Rule: Money1
// e.g. 30 pounds
(
(AMOUNT_NUMBER)
(SpaceToken.kind == space)?
({Lookup.majorType == currency_unit})
)
:money -->
:money.Number = {kind = "money", rule = "Money1"}
```

4.2.1 Step 1 - Tokenisation

The tokeniser separates this phrase into the following tokens. In general, a word is comprised of any number of letters of either case, including a hyphen, but nothing else; a number is composed of any sequence of digits; punctuation is recognised individually (each character is a separate token), and any number of consecutive spaces and/or control characters are recognised as a single spacetoken.

```
 Token, string = ‘800’, kind = number, length = 3
 Token, string = ‘,’, kind = punctuation, length = 1
 Token, string = ‘000’, kind = number, length = 3
 SpaceToken, string = ‘ ’, kind = space, length = 1
 Token, string = ‘US’, kind = word, length = 2, orth = allCaps
 SpaceToken, string = ‘ ’, kind = space, length = 1
 Token, string = ‘dollars’, kind = word, length = 7, orth = lowercase
```

4.2.2 Step 2 - List Lookup

The gazetteer lists are then searched to find all occurrences of matching words in the text. It finds the following match for the string ‘US dollars’: Lookup, minorType = post_amount, majorType = currency_unit

4.2.3 Step 3 - Grammar Rules

The grammar rule for money is then invoked. The macro MILLION_BILLION recognises any of the strings ‘m’, ‘million’, ‘b’, ‘billion’. Since none of these exist in the text, it passes onto the next macro. The AMOUNT_NUMBER macro recognises a number, optionally followed by any number of sequences of the form‘dot or comma plus number’, followed by an optional space and an optional MILLION_BILLION. In this case, ‘800,000’ will be recognised. Finally, the rule Money1 is invoked. This recognises the string identified by the AMOUNT_NUMBER macro, followed by an optional space, followed by a unit of currency (as determined by the gazetteer). In this case, ‘US dollars’ has been identified as a currency unit, so the rule Money1 recognises the entire string ‘800,000 US dollars’. Following the rule, it will be annotated as a Number entity of type Money:
Number, kind = money, rule = Money1

Chapter 5

Understanding conceptual instrumentation (W_4)

GATE includes resources for common LE data structures and algorithms, including documents, corpora and various annotation types, a set of language analysis components for Information Extraction and a range of data visualisation and editing components.

GATE supports documents in a variety of formats including XML, RTF, email, HTML, SGML and plain text. In all cases the format is analysed and converted into a sin- gle unified model of annotation.

Algorithm 1: Required steps for annotating a specific text.

```
Input: $\mathcal{T}$ - set of texts on different formats $\mathcal{W}$ - a JAPE grammar rules for annotation
       creation; $\mathcal{X}$ - a gazetteer list
Output: $\langle annotated..text \rangle$, the original text but annotated with the rules provided by
        the JAPE grammar file
1  foreach $t \in \mathcal{T}$ do
2      $DocumentReset(t, t1)$
3  foreach $t1 \in \mathcal{T}$ do
4      $EnglishTokeniser(t1, t2)$
5  foreach $t2 \in \mathcal{T}$ do
6      $SentenceSplitter(t2, t3)$
7  foreach $t3 \in \mathcal{T}$ do
8      $POSTagger(t3, t4)$
9  foreach $t4 \in \mathcal{T}$ do
10     $Gazetteer(t4, X, t5)$
11 foreach $t5 \in \mathcal{T}$ do
12     $TrandsducerJAPE(t5, W, t6)$
13 foreach $t6 \in \mathcal{T}$ do
14     display each t6 (being the annotated text)
```

Chapter 6

Project description (W_5)

The objectives for this week are:

1. To have a clear description of what you intend to develop.
2. To point to specific resources (datasets, knowledge bases, external tools) that support the development of your idea and which minimise the risk of failure.
3. To identify related work (articles) that are relevant or similar to your approach.

Realistic and original scenarious are encouraged. Well known toy problems (salesmen, map coloring, logistic planning, wumpus, sudoku, queens, missionaires and canibals, etc.) do not worth much for your grade. Your scenario should be realistic and should be business oriented. Note that the focus is both on programming and on modelling the reality into a formal representation.

Consider answering to the following questions:

1. What will your system do?
2. Which is the scope of coverage your system aims for?
3. What will be the input of your program?
4. What will be the output of your program?
5. What will be the knowledge of your system?
6. Which would be the narrative description of running scenario(s)?

Example 1 (What will system do)

Example 2 (Scope of the program)

6.1 Narrative description

The project that i will develop around the GATE tool, will be about the concept of opinion mining/sentiment analysis. More specifically, i will try to use text processing for identifying consumers reviews, the domain that i will target, will be product reviews on different online shopping websites. The reviews of the consumers will be annotated and classified as positive, negative or neutral.

6.2 Specifications

List of specifications.

The applications that can be built around the concept of sentiment analysis are diversified: business intelligence, opinion mining, reputation monitoring etc. The main problems that are we facing when using opinion mining.

- identyfing opionated segments of text.
- identyfing the opinion-holder.
- identify the argumentative structures that separates different arguments
- classifying the opinion(positive, negative, neutral).

The preprocessing of the text will be done using the ANNIE built in processing resources, the tokeniser, sentence splitter, pos tagger, gazetteer.
The reviews on the text will be identified using the JAPE grammar loaded into a Transducer, those reviews will be processed and classified as positive/negative/neutral. We can also identify the date of the review, or the author name, if one exists.

6.3 Top level design of the scenario

The final application will be made using some processing resources provided by GATE tool (the ANNIE plugin), and using grammar JAPE files for identifying the specifications. The result will be an annotated text, which will highlight the user reviews based on their arguments.

6.4 Knowledge acquisition

The documents that hold the data for processing, will be HTML TXT XML files from some online shopping websites, like Amazon, if the HTML format will be supported. After converting the HTML pages into GATE documents, we will label each review with a comment annotation, also pointing out the arguments of the reviews, positive or negative.

How do represent knowledge? Your system relies on a knowledge base. You have to describe how do you represent this knowledge. You might choose between different logics: propositional logic, first order logic, modal logics, description logics, epistemic logics, temporal logics, and so on.

Where are you getting the required knowledge/data Point towards the knowledge bases that you plan to exploit. The existence of these sources are required to prove that your approach is realistic.

Examples of knowledge sources include:

- Data sets: i.e., https://archive.ics.uci.edu/ml/datasets.html
- Statistics: i.e., http://ec.europa.eu/eurostat

- Ontology repositories

If you will be using books give their reference. If you hope to exploit people give their names. If you aim to use data sources or knowledge repositories list them and be sure that you have access to the needed knowledge. Indicate what you have accomplished so far in knowledge acquisition.

6.5 Related work

- Argumentation Mining: The Detection, Classification and Structure of Arguments in Text

  The paper Argumentation Mining: The Detection, Classification and Structure of Arguments in text, introduce some fundamental knowledge about the concept of argumentation mining. Argumentation is t he process of constructing arguments and handling them. Argumentation constitutes a major component of human intelligence. The purpose of argumentation mining, is to detect arguments from a text document, identify the relations made between them and also an internal structure for every argument individually. This paper analyses the main research questions when dealing with argumentation mining.

  First is presented the motivation of work, for this research on argumentation mining. Then, it is defined and motivated the formalism of argumentation used in their research, including knowledge on rhetorical structure, argumentation and natural language processing. Finally, it is discussed different problems encountered, when dealing with argumentation mining. For each problem it is analyzed and evaluated the possible solutions in both legal and non-legal domain texts.

- John Lawrence and Chris Reed. Combining argument mining techniques.

  The second paper that I have read regarding this topic of argumentation mining is, Combining Argument Mining Techniques by John Lawrence and Chris Reed. This paper, presents different methods of extracting the argumentative structure from a piece of natural language text. Those methods cover linguistic features, changes in the topic being discussed and a supervised machine learning approach to identify the components of argumentation schemes, patterns of human reasoning which have been detaile d extensively in philosophy and psychology. For each of these approaches they achieved results comparable to those previously reported. Finally the results from these individual techniques, applied in combination, for further improving the argument structure identification.

- Jodi Schneider. An informatics perspective on argumentation mining.

  The third paper that I will be referring to is, An Informatics Perspective on Argumentation Mining by Jodi Schneider. This paper is introduced with a series of important questions on the task of argumentation mining,then we are presented a knowledge representation section, which describes some models for structured representation of arguments, that are used in the current ontologies based on argumentative texts.

  The section that I was mostly interested in was, the task of Mining from Social Media. To identify arguments in social media, there are some specific things that might be considered: the intention of the author might be relevant, the type of messages, whether they are instruction, information, discussion, recreation or recommendation. Also the genre, metadata, properties of users, goals of a particular dialogue, sentiment and subjectivity might be relevant.

6.6 Scenario Details

As previously stated, my approach for studying GATE use on argumentation mining, is the challenge to find user reviews from a piece of text, and classify them as positive or negative, based on the user review, and also to identify the argumentative components for those piece of texts. I will use the GATE Java API, within Eclipse for developing my project.

Firstly I will do a preprocessing on the web page, using the ANNIE plugins, I will run the following PIPELINE based application, tokeniser, sentence splitter, pos tagger and gazetteer. Using the JAPE grammar boundary rules I will identify the comments section for each user, and creating an annotation set with every comment that I will find.

Within the previously annotated set I will try to identify the argumentative connectors used in reviews, for separating the positive facts from the negative ones, for a specific product. After identifying the connectors, I will annotate the words that reflect a positive opinion on a product, and separately the words reflecting a negative one.

The input data for the GATE application will be an HTML file relevant to the topic of online shopping websites, and some JAPE grammar files that will be used for creating annotations. One JAPE grammar file will be used for separating user comments, i will use two boundary tokens for detecting comments, another JAPE grammar file will be used for annotating the relevant words for identifying positive and negative comments, and also the argumentative connectors between arguments of the review.
The output of the GATE application will be an annotated document corpus.

Chapter 7

Implementation details

The objectives for this week are:

1. Illustrate each aspect of the reality that you have modelled in your solution.

   (a) The solution provided, is a basic application used to scrape comments from an Amazon review webpage of a product. The comment is extracted using the relevant tag provided in the HTML code of the webpage, using a JAPE transducer.

   (b) Comments are classified as positive or negative based on different criteria.

   (c) The comments are extracted in a CSV file, that can be used by the product support employee to understand the needs of the users, and also to review critical opinions on their products.

2. To explain the relevant code from your scenario.

7.1 Relevant code

7.1.1 Jape Code

The main objective of the Jape code is to provide grammar rules for identifying different segments of text, and also for creating new annotations on relevant keywords / segments. I have separated my grammar into 6 Jape files, each one being used for a separate task.

1. The 'title.jape' file used for identifying the product title on the HTML page, based on the specific tag. SourceCode

2. The 'comments.jape' file used for the purpose of identifying comments within the HTML file. SourceCode

3. The 'goodAdjective.jape' file used to identify the words that would cause a comment to be more likely positive than negative. And the 'positiveComments.jape' file used with the positive adjectives, to identify the positive comments. SourceCode

4. The 'badAdjective.jape' file and the 'negativeComments.jape' file, are similar to the ones provided above, excepting the fact that the last one is using another gazetteer for finding the bad adjectives, relevant for the negative comments.

7.1.2 Gate Java Code

I developed my small application in Java, where i'm initiating and running the Gate tool, also loading the specified resources needed for processing and the documents from the dataset, that must be saved on the local computer, in the 'data' folder of the Java application. Some of the relevant Java code used for creating this application is described below:

1. For initializing and loading the processing resources needed. My application is using the ANNIE plugin, more specifically the Tokeniser and the Gazetteer. ExampleCode

2. The datasets used are loaded into Gate for further processing. ExampleCode

3. The annotations created by the Jape grammar are manipulated, being saved for further processing. In our case, we are saving them on a CSV file. ExampleCode

   The classes used for developing the functionalities:

   - LoadPRCLass.java - used for loading the processing resources needed into our application.
   - DocumentProcessorClass.java - used for iterating over the documents on the dataset and applying the processing resources specified in the previous class.
   - WriteXLSClass.java - used for exporting the annotations created in the previous class, for each document, creating a new CSV row, where we are saving the following things: number of comments, number of positive comments, number of negative comments, title of the product, and followed by the comments for each product.

Chapter 8

Tool expressivity (W_{11})

1. Gate - as a General Architecture Text Processing is a powerfull tool for natural language processing, having a lot of features, plugins, and with plenty of applications in the real world. In my application I have managed to touch a small amount of the full potential of the tool.

2. The Plugin that i have used for performing my task, is the ANNIE plugin that is providing a vast amount of functionalities. Firstly I had to separate the text into tokens before further processing, this step was possible using the Tokeniser provided in the Annie plugin. The second plugin used in this project, is the Gazetteer provided also in the ANNIE plugin. I created two new gazetteers, one for positive adjectives and another one for negative adjectives, using the words defined within those gazetteers to identify the positive and the negative comments.

3. Another feature provided by the Gate tool, is the Jape Transducer, I have used this for defining my grammar needed to identify the opionated segments of text.

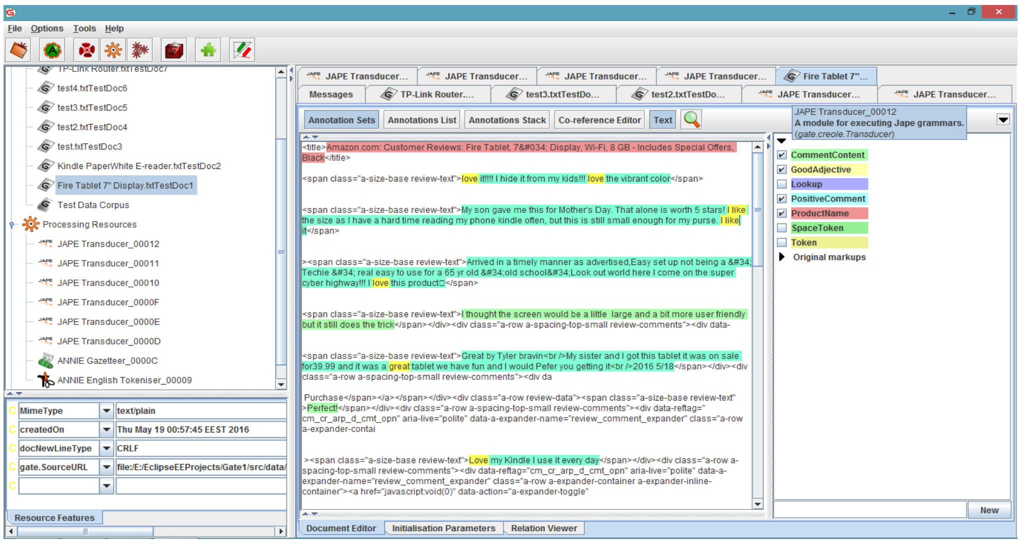
4. Managing the datasets is another good point of Gate, being able to create a Corpus from the documents provided. This corpus being processed by the processing resources within the program. Gate is using a PIPELINE based application, doing each processing step by step, in the order provided by the user.

Chapter 9

Graph and experiments (W_{12})

9.1 Results

The following picture shows the result of running our application. We can distinguish between 4 types of annotations on the image. The RED annotation is for the title of the product being reviewed by the customers. The YELLOW annotations represent the positive adjectives, that are likely to indicate that the whole comment is positive, being annotated with TEAL color. The GREEN annotations, are used for comments in general, not being specifically positive or negative.



MimeType	text/plain
createdOn	Thu May 19 00:57:45 EEST 2016
docNewLineType	CRLF
gate.SourceURL	file:/E:/EclipseEEProjects/Gate1/src/data/

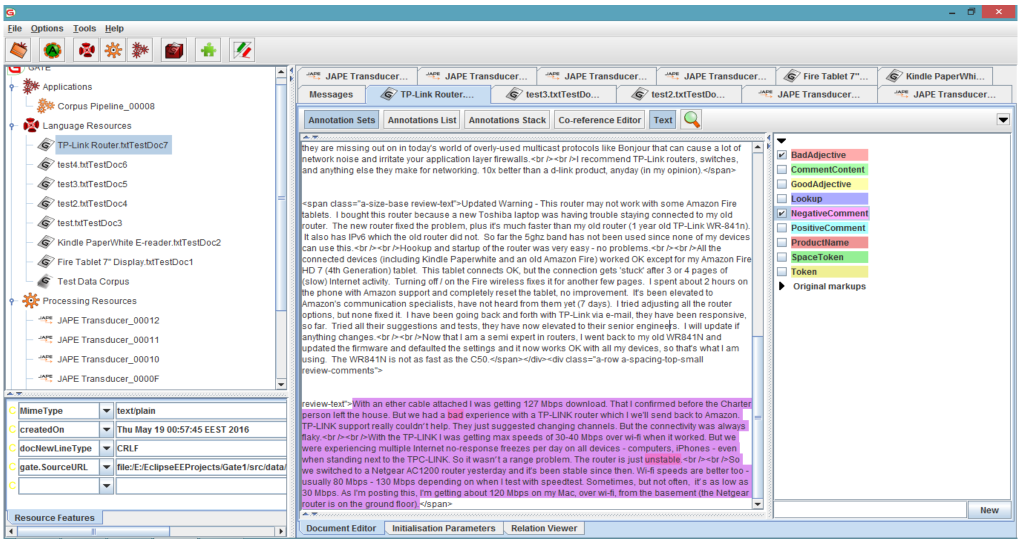
JAPE Transducer_00012
A module for executing Jape grammars.
(gate.creole.Transducer)

- ✓ CommentContent
- ✓ GoodAdjective
- ☐ Lookup
- ✓ PositiveComment
- ✓ ProductName
- ☐ SpaceToken
- ☐ Token
- Original markups

```
<title>Amazon.com: Customer Reviews: Fire Tablet, 7&#034; Display, Wi-Fi, 8 GB - Includes Special Offers,
Black</title>

<span class="a-size-base review-text">love it!!!! I hide it from my kids!!! love the vibrant color</span>


<span class="a-size-base review-text">My son gave me this for Mother's Day. That alone is worth 5 stars! I like
the size as I have a hard time reading my phone kindle often, but this is still small enough for my purse. I like
it</span>


><span class="a-size-base review-text">Arrived in a timely manner as advertised,Easy set up not being a &#34;
Techie &#34; real easy to use for a 65 yr old &#34;old school&#34;Look out world here I come on the super
cyber highway!!! I love this product□</span>


<span class="a-size-base review-text">I thought the screen would be a little  large and a bit more user friendly
but it still does the trick</span></div><div class="a-row a-spacing-top-small review-comments"><div data-


<span class="a-size-base review-text">Great by Tyler bravin<br />My sister and I got this tablet it was on sale
for39.99 and it was a great tablet we have fun and I would Pefer you getting it<br />2016 5/18</span></div><div
class="a-row a-spacing-top-small review-comments"><div da

 Purchase</span></a></span></div><div class="a-row review-data"><span class="a-size-base review-text"
>Perfect!</span></div><div class="a-row a-spacing-top-small review-comments"><div data-reftag="
cm_cr_arp_d_cmt_opn" aria-live="polite" data-a-expander-name="review_comment_expander" class="a-row
a-expander-contai


 ><span class="a-size-base review-text">Love my Kindle I use it every day</span></div><div class="a-row a-
spacing-top-small review-comments"><div data-reftag="cm_cr_arp_d_cmt_opn" aria-live="polite" data-a-
expander-name="review_comment_expander" class="a-row a-expander-container a-expander-inline-
container"><a href="javascript:void(0)" data-action="a-expander-toggle"
```

Figure 9.1: Test 1 - Finding Positive Comments

The picture below, shows an example of finding an negative comment, using the adjectives provided in the negative adjective gazetteer. Note that if a comment contains positive and negative adjectives, it will be classified both as positive and negative.



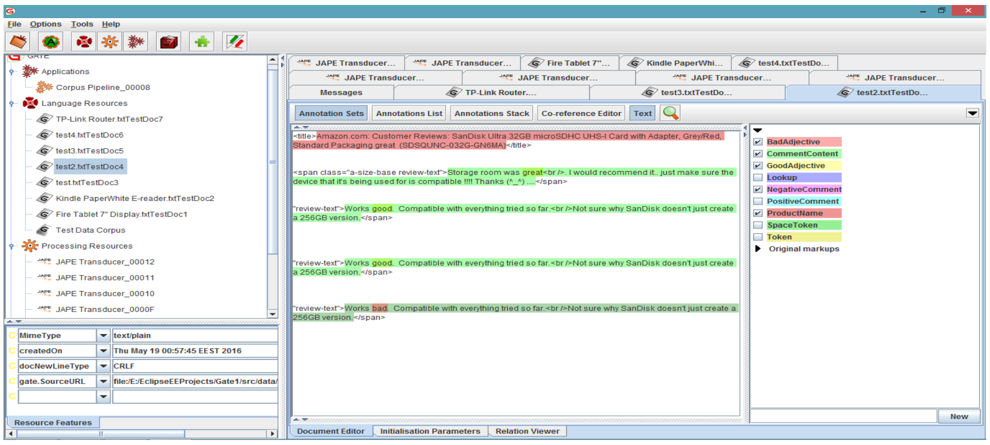
```
they are missing out on in today's world of overly-used multicast protocols like Bonjour that can cause a lot of
network noise and irritate your application layer firewalls.<br /><br />I recommend TP-Link routers, switches,
and anything else they make for networking. 10x better than a d-link product, anyday (in my opinion).</span>


<span class="a-size-base review-text">Updated Warning - This router may not work with some Amazon Fire
tablets.  I bought this router because a new Toshiba laptop was having trouble staying connected to my old
router.  The new router fixed the problem, plus it's much faster than my old router (1 year old TP-Link WR-841n).
 It also has IPv6 which the old router did not.  So far the 5ghz band has not been used since none of my devices
can use this.<br /><br />Hookup and startup of the router was very easy - no problems.<br /><br />All the
connected devices (including Kindle Paperwhite and an old Amazon Fire) worked OK except for my Amazon Fire
HD 7 (4th Generation) tablet.  This tablet connects OK, but the connection gets 'stuck' after 3 or 4 pages of
(slow) Internet activity.  Turning off / on the Fire wireless fixes it for another few pages.  I spent about 2 hours on
the phone with Amazon support and completely reset the tablet, no improvement.  It's been elevated to
Amazon's communication specialists, have not heard from them yet (7 days).  I tried adjusting all the router
options, but none fixed it.  I have been going back and forth with TP-Link via e-mail, they have been responsive,
so far.  Tried all their suggestions and tests, they have now elevated to their senior engineers.  I will update if
anything changes.<br /><br />Now that I am a semi expert in routers, I went back to my old WR841N and
updated the firmware and defaulted the settings and it now works OK with all my devices, so that's what I am
using.  The WR841N is not as fast as the C50.</span></div><div class="a-row a-spacing-top-small
review-comments">


review-text">With an ether cable attached I was getting 127 Mbps download. That I confirmed before the Charter
person left the house. But we had a bad experience with a TP-LINK router which I we'll send back to Amazon.
TP-LINK support really couldn't help. They just suggested changing channels. But the connectivity was always
flaky.<br /><br />With the TP-LINK I was getting max speeds of 30-40 Mbps over wi-fi when it worked. But we
were experiencing multiple Internet no-response freezes per day on all devices - computers, iPhones - even
when standing next to the TPC-LINK. So it wasn't a range problem. The router is just unstable.<br /><br />So
we switched to a Netgear AC1200 router yesterday and it's been stable since then. Wi-fi speeds are better too -
usually 80 Mbps - 130 Mbps depending on when I test with speedtest. Sometimes, but not often,  it's as low as
30 Mbps. As I'm posting this, I'm getting about 120 Mbps on my Mac, over wi-fi, from the basement (the Netgear
router is on the ground floor).</span>
```

MimeType	text/plain
createdOn	Thu May 19 00:57:45 EEST 2016
docNewLineType	CRLF
gate.SourceURL	file:/E:/EclipseEEProjects/Gate1/src/data/

- [✓] BadAdjective
- [ ] CommentContent
- [ ] GoodAdjective
- [ ] Lookup
- [✓] NegativeComment
- [ ] PositiveComment
- [ ] ProductName
- [ ] SpaceToken
- [ ] Token

Figure 9.2: Test 2 - Finding Negative Comments

The last image, display annotated text of a combination between positive and negative comments.



```
<title>Amazon.com: Customer Reviews: SanDisk Ultra 32GB microSDHC UHS-I Card with Adapter, Grey/Red,
Standard Packaging great  (SDSQUNC-032G-GN6MA)</title>


<span class="a-size-base review-text">Storage room was great<br />. I would recommend it.. just make sure the
device that it's being used for is compatible !!!! Thanks (^_^) ....</span>


"review-text">Works good.  Compatible with everything tried so far.<br />Not sure why SanDisk doesn't just create
a 256GB version.</span>


"review-text">Works good.  Compatible with everything tried so far.<br />Not sure why SanDisk doesn't just create
a 256GB version.</span>


"review-text">Works bad.  Compatible with everything tried so far.<br />Not sure why SanDisk doesn't just create a
256GB version.</span>
```

MimeType	text/plain
createdOn	Thu May 19 00:57:45 EEST 2016
docNewLineType	CRLF
gate.SourceURL	file:/E:/EclipseEEProjects/Gate1/src/data/

- [✓] BadAdjective
- [✓] CommentContent
- [✓] GoodAdjective
- [ ] Lookup
- [✓] NegativeComment
- [ ] PositiveComment
- [✓] ProductName
- [ ] SpaceToken
- [ ] Token

Figure 9.3: Test 3 - Finding Negative/Positive Comments

Chapter 10

Related Work and Discussion (W_{13})

The objectives for this week are:

1. To compare your results to related work.
2. To discuss the advantages and limitations of your solution.

Identify and describe other solutions for solving the same scenario like yours. For instance: [?]

10.1 Related approaches

The related approaches that I have discussed in the Related Work section, were mostly based in the topic of argumentation mining as a general thing. Identyfing the argumentative structures of the text based on different criteria, was the main topic of discussion in those papsers. My approach is using just basic functionalities of Gate for identifying the opionated segment of text, not classifying it as a structure having premises / arguments / conclusion, as it was explained in those papers.

10.2 Advantages and limitations of your solution

- Advantges
    1. The solution developed for this scenario is very simple and easy to undestand. I am using the basic functionalities of Gate for retrieving comments from text. The technique appplied for mining comments, is not very complicated, it's based only on identifying the positive/negative adjectives inside a comment and classify it based on those two criteria.
    2. One advantage provided in our application, is the exporting step, after identifying the comments and the product title, we can export those in a CSV file for further research. We are saving in the file the number of comments, number of positive/negative comments, title of the product and also the comments from the text.
- Limitations
    1. One of the main disadvantages of the solution provided, comes when treating ambiguos comments, people that may refer to other products in their review, comparing those with this one, and providing different kinds of adjectives in this comparison, so we can't really tell if they are referring to the specific product (subject of most reviews) or another one.

2. Another limitation of this approach, is the fact that we are not treating the neutral comments, some of the comments might have a positive section and also a negative section, providing pros and cons about the product. Our solution, will identify this kind of comments, both as positive and negative.
3. When it comes to data source, another problem that we are facing, is about the size of the HTML source file that must be processed. Before running the tokeniser on the source files, we have to separate the part of the HTML files, saving only the relevant information needed, in our case we need the header part of the HTML file where the title is contained, and the comment section. If we are processing the whole HTML file, this will take a lot of time, and even running out of memory, because Gate is storing all the annotations that we are creating on the heap.

10.3 Possible extensions of the current work

The current solution is just a template for a further more complicated comment minining approach. We can add plenty of new functionalities, that would improve our review retrieval technique, and also the review classifying part of the application, providing more accurate classification. Some of the improvements that will point out are listed below.

- Adding support for multiple websites, this can be done by changing the comment Jape file, adding the new tag-value identification approach for the specified website.
- Classifying the reviews based on the user rating value, a lot of websites provide a system to rate a product after making a review, we can take advantage of those rating for further processing of our reviews, classifying them as positive/negative based on this criteria.
- As we presented in the Limitations section, our current system approach when detecting comments is based only on the adjectives presented in the comment, we can change this to provide a wider approach, classifying only one sentence from the review as positive/negative based on the content.
- Introducing a new category of comments, 'Neutral' comments, those annotations will be created when the reviewer is referring to the good and to the bad parts of the product.
- Improving the mechanism for classifying comments, using other approaches, not only adjectives. For example we can give a review a grade from -1 to 1, where -1 means the comment is negative and 1 means the comment is positive. For calculating this grade, we can check the kind of sentences in the review (exclamative/affirmative), the attitude of the reviewer aggressive using multiple verbs, the argumentative connectors but//although/... .

Chapter 11

Documentation (W_{14})

The objectives for this week are:

1. To deliver a professional documentation of your work.

   This template (filled of course).

Appendix A

Your original code

This section should contain only code developed by you, without any line re-used from other sources. This section helps me to correctly evaluate your amount of work and results obtained. Including in this section any line of code taken from someone else leads to failure of IS class this year.

A.1 Jape Code

A.1.1 Title Jape Code

```
Phase: TitleFinder
Input: Token
Options: control = first
Rule: TitleFind
(
 {Token.string == "title"}
 {Token.string == ">"}
 ({Token})* :value
 {Token.string=="<"}
)
-->
:value.ProductName ={title=:value@cleanString}
```

A.1.2 Comments Jape Code

```
Phase: ValueFinder
Input: Token
Options: control=first
Rule: Value
(
 {Token.string=="review-text"}
 {Token.string=="\""}
 {Token.string==">"}
 ({Token})* :value
 {Token.string=="<"}
 {Token.string=="/"}
 {Token.string=="span"}
```

```
   )
   -->
   :value.CommentContent = {string=:value@cleanString}
```

A.1.3 Positive Comments Jape Code

```
  Phase: AdjectiveFinder
  Input: Lookup Token
  Options: control = brill
  Rule: PosAdj
  (
         {Token.string=="review-text"}
         {Token.string=="\""}
         {Token.string==">"}

         (
                ({Token})*
                ({Lookup.majorType == comment}):title
                ({Token})*
         )


         {Token.string=="<"}
         {Token.string=="/"}
         {Token.string=="span"}
  )
-->
:title.GoodAdjective ={rule= PosAdj }


  Phase: PositiveCommentFinder

  Input: Token Lookup

  Options: control = first

  Rule: PosComm
  (
         {Token.string=="review-text"}
         {Token.string=="\""}
         {Token.string==">"}
         ({Token})* :commentValue
         {Token.string=="<"}
         {Token.string=="/"}
         {Token.string=="span"}
  ):CommentAnn

  -->
  {
         AnnotationSet CommentAnnotation = (AnnotationSet) bindings.get("
             commentValue");
```

```
        //get goodAdjective from the input comment annotation , itf exists make it
           a positive comment
        AnnotationSet PosAdj = inputAS.get("GoodAdjective", CommentAnnotation.
           firstNode().getOffset(), CommentAnnotation.lastNode().getOffset());

        if (PosAdj.size() != 0 ) {
               FeatureMap features = Factory.newFeatureMap();
               outputAS.add(CommentAnnotation.firstNode(), CommentAnnotation.
                  lastNode(), "PositiveComment", features);
        }
}
```

Appendix B

Other Relevant Code

B.1 Example 1 - Jape Grammar

```
Phase: locationcontext1
Input:  Lookup Token
Options: control = applet debug = false

//rule identifies stadiums/circuit/golf clubs
Rule: locationcontext1
Priority:50
(
     (
       (
       (
        (
           {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}
           |
           {Token.kind == word, Token.category == NNPS, Token.orth == upperInitial}

        (
           {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}
           |
           {Token.kind == word, Token.category == NNPS, Token.orth == upperInitial}
        )?
       )
       (
           {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}
           |
           {Token.kind == word, Token.category == NNPS, Token.orth == upperInitial}
       )?

       )

 )
 (
 {Token.string =~ "[Ss]tadium"}
 |
 {Token.string =~ "[Cc]ircuit"}
 |
 (
 {Token.string =~ "[Gg]olf"}
 ({Token}{Token})?
 {Token.string =~ "[Cc]lub"}
 )
 |
 {Token.string =~ "[Aa]rena"}
    )%\input{mycode}
):location
 -->
:location.Location = {rule= "locationcontext1-locationcontext1" }
```

B.2 Java Relevant Code

B.2.1 Loading Processing Resources Ex

```
//Init Gate Library
Gate.init();

// to show the GATE Developer interface
MainFrame.getInstance().setVisible(true);

File pluginDir = Gate.getPluginsHome(); // get plugins home directory
URL anniePlugin = new File(pluginDir, "ANNIE").toURI().toURL(); //
    specify plugin to be loaded
Gate.getCreoleRegister().registerDirectories(anniePlugin); // finally
     register the plugin

// setting up searialAnalyserController
SerialAnalyserController sac = (SerialAnalyserController)Factory.
    createResource("gate.creole.SerialAnalyserController");

// setting up processing resources, only tokeniser needed
ProcessingResource aEngTokeniser = (ProcessingResource) Factory.
    createResource("gate.creole.tokeniser.DefaultTokeniser");

//gazetteer
ProcessingResource annGazetteer = (ProcessingResource) Factory.
    createResource("gate.creole.gazetteer.DefaultGazetteer");

//Creating a featureMap for getting the positive adjectives
FeatureMap positiveAdjectiveJapeFeature = Factory.newFeatureMap();
positiveAdjectiveJapeFeature.put("grammarURL", new File("src/grammar/
    goodAdjective.jape").toURI().toURL());

//Creating the Jape Transducer
LanguageAnalyser positiveAdjectivesJape = (LanguageAnalyser) Factory.
    createResource("gate.creole.Transducer",
    positiveAdjectiveJapeFeature);
```

```
        //Adding Jape Transducer to the SAC after running the Tokeniser and
           the Gazetteer
        sac.add(aEngTokeniser);
        sac.add(annGazetteer);
        sac.add(positiveAdjectivesJape);
```

B.2.2 Loading Documents Ex

```
        // create a corpus for storing results
        Corpus corpus = Factory.newCorpus("Test Data Corpus");

        // arraylist to store document resources
        ArrayList<Document> documentResList = new ArrayList<Document>();

        FeatureMap features = Factory.newFeatureMap();
        features.put("createdOn", new Date());


        //create doc with specified params, features and unique name
        Document doc = (Document) Factory.createResource("gate.corpora.
           DocumentImpl", params, features, f.getName() + "TestDoc" + i);

        //add doc to corpus
        corpus.add(doc);

        //also maintain a list of these documents
        documentResList.add(doc);
```

B.2.3 Using Created Annotations

```
//start analyzing each data from the corpus
for (Iterator<Document> corpIterator = corpus.iterator(); corpIterator.hasNext();) {
       //increment rowCount since doc row added; represents current document number
       rowCount++;

       //get the document from corpus
       Document corpDoc = corpIterator.next();

       //getting default set of annotations
       AnnotationSet defaultSet = corpDoc.getAnnotations();

       //Getting annotations of Comments
       AnnotationSet commentTypeAnnotations = defaultSet.get("CommentContent");

       //Getting ProductTitle Annotations
       AnnotationSet titleTypeAnnotations = defaultSet.get("ProductName");

       //Getting annotations of PositiveComments
       AnnotationSet positiveCommentTypeAnnotations = defaultSet.get("
          PositiveComment");

       //Getting annotations of NegativeComments
```

```
        AnnotationSet negativeCommentTypeAnnotations = defaultSet.get("
            NegativeComment");


        //ArrayList for annotations to be saved in the Excel
        ArrayList<String> annList = new ArrayList<String>();

        //Adding the number of commentAnnotations
        annList.add(String.valueOf(commentTypeAnnotations.size()));

        //Adding the number of positiveCommentAnnotations
        annList.add(String.valueOf(positiveCommentTypeAnnotations.size()));

        //Adding the number of negativeCommentAnnotations
        annList.add(String.valueOf(negativeCommentTypeAnnotations.size()));


        //Adding the title of the product to the annList
        if (titleTypeAnnotations.size() > 1) {
                annList.add("NoTitle");
        }
        else {
                for (Annotation titleAnnotation : titleTypeAnnotations) {
                        FeatureMap titleFeatureMap = titleAnnotation.getFeatures();
                        String titleNamesString = titleFeatureMap.get("title").
                            toString();
                        annList.add(titleNamesString.trim());
                }
        }

        //iterate over these commentAnnotations and add them to list
        for (Annotation commentAnnotation : commentTypeAnnotations) {
        FeatureMap commentFeatureMap = commentAnnotation.getFeatures();
        String commentString = commentFeatureMap.get("string").toString();
        annList.add(commentString.trim());
}

rows.add(annList);
currentDOc++;
}
```

Bibliography

[1] John Lawrence and Chris Reed. Combining argument mining techniques. *NAACL HLT 2015*, page 127, 2015.

[2] Raquel Mochales and Marie-Francine Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22, 2011.

[3] Raquel Mochales Palau and Marie-Francine Moens. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107. ACM, 2009.

[4] Jodi Schneider. An informatics perspective on argumentation mining. In *ArgNLP*, 2014.