Exercise Sheet 8

GATE

*Submit your solutions until **Monday, 02.05.2016, 12h00** by uploading them to ILIAS. Later submissions won't be considered. Every solution should contain the **name(s), email adress(es)** and **registration number(s)** of its (co-)editor(s).*

1 Preparation

In order to ensure that I can read the GATE applications and files you create, GATE has to be installed in the following way:

- Download and install GATE from http://gate.ac.uk/download/ Do not use the installer, but the **gate-7.1-buildXXXX-ALL.zip** file (more than 400MB). You will need to provide your own Java VM (version 6.0 or higher) and set your JAVA_HOME appropriately. Details are in the "how to" chapter of the user guide (http://gate.ac.uk/sale/tao/splitch2.html)
- Please create a new folder, where to extract the ZIP-file containing GATE. Do not use whitespace or special characters in the name. You could, for instance, extract it to the folder *GATE*.
- In the generated path *GATE/gate-7.1-buildXXXX-ALL*, create a new folder *km*. In this folder, please create another folder with the last name of one of your group members. Use this folder as your working directory to store your results. **For submitting your application code, please compress (.zip) this folder and upload it to ILIAS.**
- As a result, if your name is *John Doe*, the path should look like this: *GATE/gate-7.1-buildXXXX-ALL/km/Doe*

2 ANNIE System (34 Points)

In this task we will test the ANNIE Pipeline on the documents **japan.html** and **brown.html**. These documents are available in ILIAS.

Therefore, several preparations are necessary:

- Start GATE
- Make sure, that no processing resources have been loaded so far.
- Load the ANNIE Processing Resources with default values. This should automatically create an application named *ANNIE* with the following components:
  1. Document Reset PR
  2. ANNIE English Tokeniser
  3. ANNIE Gazetter
  4. ANNIE Sentence Splitter
  5. ANNIE POS Tagger
  6. ANNIE NE Transducer
  7. ANNIE OrthoMatcher
- Generate a *Gate Corpus* in the *Language Resources* folder: Right click on the generated corpus and click *populate*. Afterwards, choose the folder where your *japan.html* and *brown.html* documents are stored. Now, the two documents should have been added to the Language Resource folder as well.
- Finally, run your application on the generated corpus.

After that, the two documents **japan.html** and **brown.html** have been annotated by the ANNIE Pipeline. You can see the annotations, if you click on a document and choose an annotation set, e.g. *Location*.

If you have problems running GATE, first refer to the user manual at `http://gate.ac.uk/sale/tao/split.html`. Furthermore, there are many useful tutorials on the internet, for example you can find video tutorials at the official website `http://gate.ac.uk/demos/developer-videos/`.

2.1 Annotation Results (8 Points)

Analyze ANNIE's annotations of the two texts:

a) (5 Points)

Look out for wrong and missing annotations. Name at least 5 cases where ANNIE has created no or incorrect annotations.

b) (3 Points)

Inspect the words annotated with *Lookup* by clicking on the *Annotation List* tab.

- List 10 values that are assigned to the *majorType* attribute!
- How is the *minorType* attribute related to *majorType*

2.2 Components (8 Points)

Explain the general purpose of ANNIE's components *ANNIE English Tokeniser, ANNIE Gazetter, ANNIE POS Tagger,* and *ANNIE NE Transducer*! Provide at least one example for the annotations of each of these components in the *japan.html* or *brown.html* text to support your explanations.

2.3 Introduction to JAPE Rules (14 Points)

JAPE (Java Annotation Patterns Engine) provides finite state transduction over annotations based on regular expressions. That means annotated data is matched against a regular expression and based on that match new annotations are added. For details about JAPE (the script language of GATE) refer to http://gate.ac.uk/sale/thakker-jape-tutorial/index.html

Jape rules of ANNIE's Named Entity (NE) recognition can be found in the folder *GATE/plugins/ANNIE/resources/NE*

a) (8 Points)

In the *japan.html* text, the annotation *Percent* has been added four times.

- Identify and name the rule in *number.jape* that is responsible for creating the annotation.
- Explain the rule's functionality in detail: Which lines can be grouped together to be responsible for partial matches? How can those partial matches look like? Which parts are mandatory, which are optional?

b) (6 Points)

Would the following expressions be matched by the left-hand side (head) of the rule you found? Why (not)?

1. four hundred and forty five percent
2. 100 per cent
3. − ninety-nine cent
4. +− ninety-nine percent
5. +74
6. + percent
7. +35 point seventy-nine per cent
8. 45 − 55 percent

2.4 More on JAPE Rules (10 Bonus Points)

Why is *Ban Ki-moon* not annotated with the *Person* attribute in the last sentence within the *brown.html* text? Why does the annotation creation work with *Mirek Topolanek*? This task is not as easy as it seems. You have to study the relevant rules which are responsible for the *Person* annotations. They are located in *plugins/ANNIE/resources/NE*.

- Give a detailed explanation. Name and explain the relevant rules in your answer.
- What could be done to ensure that Ban Ki-Moon is also annotated as *Person*. Provide two possible solutions.

2.5 Custom JAPE Rule (6 Bonus Points)

Write your own JAPE rule which annotates the string "G20" with *VeryImportantCountries*. Your rule should also add the attribute *Population* to the *VeryImportantCountries* annotation, which contains the overall population of all G20 countries which is *4385* (million).

Introduce your rule as a Processing Resource of the type JAPE Transducer. Then, add your Processing Resource to the existing ANNIE application. Finally, run the application on the *brown.html* text.

Make a screenshot of the new annotations your rule has created. Furthermore add your source code to your submission (directly on the submission, not as a separate file!) .

3 WordNet (11 Points)

WordNet is a database that stores information about words and their relations. In the following tasks you will work with WordNet and learn about its structure. A browsable version of WordNet can be found at http://wordnetweb.princeton.edu/perl/webwn, you can start browsing by looking up any word. The terms used by WordNet, e.g. *synset*, are explained in this glossary http://wordnet.princeton.edu/man/wngloss.7WN.html

3.1 Word Senses (5 Points)

WordNet provides so called *synsets* which are "synonym sets; a set of words that are interchangeable in some context without changing the truth value of the proposition in which they are embedded." Each synset represents one *sense* of a word which is the word's meaning. The same word can occurr in different synsets if it has more than one meaning. For example the first sense of the word "bank" is "sloping land (especially the slope beside a body of water)", its second sense "a financial institution that accepts deposits and channels the money into lending activities". You can find out the *sense number*, if you activate *show sense numbers* in the display options of the WordNet browser.

Find the correct WordNet sense number for the highlighted words in the following sentences!

> *I*n botany, a tree is a **plant** with an elongated **stem**, or **trunk**, supporting **leaves** or **branches**.

> *A* manufacturing **plant** is an industrial site, usually consisting of buildings and machinery, or more commonly a **complex** having several buildings, where workers manufacture **good**s or operate machines processing one **product** into another.

3.2 Finding Relations (6 Points)

WordNet defines different relations between two words:

1. synonym: X and Y share almost the same meaning
2. hyponym: X is a type of Y
3. hypernym: X is a generalization of Y
4. holonym: X consists (partially) of Y
5. meronym: X is part of Y

As you can see some relations just are the inverse of each other.

Some examples:

- *mouse* is a hyponym of *rodent*
- *mouse* is a hypernym of *field mouse*.
- *field mouse* is a hyponym of *rodent*.
- *serpent* is a synonym of *snake* (and the other way around)
- *motor* is a meronym of *car*
- *car* is a holonym of *tire*

Note that the relation between *field mouse* and *rodent* is inherited. If *mouse* is a hyponym of *rodent* and *field mouse* is a hyponym of *mouse* then *field mouse* is a hyponym of *rodent* as well.

Find the relations given above between the following words in WordNet. The relations may either be direct or inherited. Also provide the respective sense number for each word.

1. stem – trunk
2. trunk – tree
3. tree – branch
4. manufacturing plant – plant
5. building complex – plant
6. tree – plant