**Natural Language Process: COMS4705**

**Kathy McKeown, Fall 2009**

**Due: Sept. 30th, 2009, midnight**

**Homework 1: Stock Market QA System (100 points)**

Please post any questions concerning this assignment to the Courseworks (*http://courseworks.columbia.edu*) discussion board, under the Homework 1 topic.

**1 General Instructions**

The main goal of Homework 1 is to produce a simple question answering (QA) system using regular expressions to retrieve information from a single news article related to the stock market.

Your system should be able to respond to a list of queries from a file, where each query is on a separate line. It should also be able to respond to a single query from interactive console input. When responding to queries from a file, your QA system should respond to each line, and terminate; when responding to console input, it should loop until the user quits. We will provide a sample file as input for ‘file mode'. You can use this file as `development data' as you build your QA system. We will also provide financial articles that can be used to train your system. You can see a link to these files next to the link to HW1.

You must write the code yourself. Don't use publicly available code (please refer to the Academic Integrity policy if you have questions in this regard). You may use NLP analysis tools available on the web (e.g. part-of-speech taggers, morphological analyzers) but you must indicate which ones you use, for what purpose, and where they were obtained. Unless you have discussed it with the TA’s beforehand, your assignment should be written in Perl or Java **and should compile and run on the CS cluster machines**.

Your submission must include a README file as specified in Section 2.2 below. Also include code for your program and any supporting data files you use, as well as the required compilation and execution scripts as described in Section 3 below.

Your submission should not include compiled code. If your program that does not compile or execute, you have the option of trying again by turning in a late submission the next day and taking the penalty. Otherwise, a program that does not compile or execute will lose a minimum of 50% of the grade.

**2 Grading**

You will be graded on the following elements:

**2.1 Functionality (80 points total)**

**Functionality (42 points)**

Your system should be able to correctly answer the following questions. It should also be able to handle paraphrases of the questions for indices such as the *Dow Jones* (ex: *Dow Jones industrials*, *the industrial* *average*) and verbs (e.g. *rise*, *climb*). You should always give the source for the answer (i.e., the sentence where the answer came from). Do not worry about time. If the input file indicates any rise or fall throughout the day or in the past, you should mention it.

1. Did <index> rise or fall?
2. Did <company stock> rise or fall?
3. How much did <index> rise/fall?
4. How much did <company stock> rise/fall?
5. How much did <index> close/open at?
6. How much did <company stock> close/open at?

**Examples:**

Q: How much did the Dow Jones close at?

A: 2569.6

Source: “The Dow Jones industrials closed at 2659.26” (line 10)

Q: How much did the Industrials Average close at?

A: 2569.6

Source: “The Dow Jones industrials closed at 2659.26” (line 10)

Q: Did Delta Airlines rise or fall?

A: It fell.

Source: For example, their selling caused trading halts to be declared in USAir Group, which closed down 3 7/8 to 41 1/2, Delta Air Lines, which fell 7 3/4 to 69 1/4, and Philips Industries, which sank 3 to 21 1/2. (line 43)

Q; How much did Delta Airlines drop?

A: 7 3/4

Source: For example, their selling caused trading halts to be declared in USAir Group, which closed down 3 7/8 to 41 1/2, Delta Air Lines, which fell 7 3/4 to 69 1/4, and Philips Industries, which sank 3 to 21 1/2. (line 43)

**Regular Expression templates (8 points)**

You should have good regular expression templates for the questions and answers. Quality is more important than quantity. More general regular expressions, that can match multiple questions or multiple kinds of sentences for the answer, are better than rigid regular expressions that match only one string. With more general regular expressions, you will need to create fewer overall.

**Preciseness of answer (10 points)**

Your answer should be as specific as possible. For example, for the first question above, the answer should be **“2569.6”** not **“closed at 2569.6”**.

**Multiple answers (10 points)**

List all possible answers where applicable. You will be penalized for missed answers

**Example:**

Q: How much did the Dow Jones fall?

A1: 55

Source 1: The Dow industrials were down 55 points at 3 p.m. (line 10)

A2: 114.76

Source: At 3:30 p.m., at the end of the "cooling off" period, the average was down 114.76 points. (line 15)

**No answer available (10 points)**

Correctly identify when there is no answer available.

**Example:**

Q: How much did the S&P market rise?

A: No information available.

**Incorrect Answer**

You will be penalized for an incorrect answer

**Example:**

Q: How much did the Dow fall?

A1: Dow Richardson

Source: -- Dow Richardson. (line 5)

**2.2 Software Engineering (includes documentation) (20 pts.)**

**Your README file must include the following:**

* Your name and email address.
* Homework number
* A description of every file in your solution, the programming language used, supporting files, any NLP tools used, etc.
* How your QA system operates, in detail.
* A description of special features (or limitations) of your QA system.

**Within Code Documentation:**

* Methods/functions/procedures should be documented in a meaningful way. This can mean expressive function/variable names as well as explicit documentation.
* Informative method/procedure/function/variable names.
* Efficient implementation
* Programmer, Memory, and Processor efficiency. Don't sacrifice one unless another is improved
* Don't hardcode variable values, etc.

**3 Submission instructions**

If you use a language that requires compilation, you must include a shell script that automatically compiles your code **on the CS cluster machines**. This should be called make.sh.

Regardless of language used, you must include a simple perl wrapper called

hw1.pl

so that your program can be run as follows (again, on the cluster machines):

$ perl hw1.pl <financial article> (<qa filename>)

Note that <qa filename> is optional, as your QA must respond both to queries from a file, one line at a time, and take interactive console input.

Hint: this simple perl script lists the contents of the present working directory:

print ` ls`

When you have completed your system, you will submit your solution electronically using instructions under the “submission” link at the top of the NLP class website (http://www.cs.columbia.edu/~kathy/NLP).

**4 Academic Integrity**

Copying or paraphrasing someone's work (code included), or permitting your own work to be copied or paraphrased, even if only in part, is not allowed, and will result in an automatic grade of 0 for the entire assignment or exam in which the copying or paraphrasing was done. Your grade should reflect your own work. If you believe you are going to have trouble completing an assignment, please talk to the instructor or TA in advance of the due date.