The story of the TA

Once upon a time, there was a TA who had many papers to grade for Mr. Adviser. Mr. TA would meet Mr. Adviser every Friday morning to discuss planning for class and to turn in the previous weeks homework assignments.
Every Friday was the same for Mr. TA until one particular day, when Mr. TA woke up late because he had been working all night on Project IV for his Theory of Computation class.

Fearing that he might be late for the first time in his life, Mr. TA rushed to his meeting with Mr. Adviser.
On his way to school, Mr. TA saluted the new school mascot, the digestion fighting tartan dinosaur. He looked like he was fighting a digestion problem at the moment, so Mr. TA wished him well, and headed to his appointment with Mr. Adviser.
Mr. TA said hi to Mr. Adviser, but he was too busy with his new proof that theory of computation is only meant to be studied by people with an IQ over 150. Since he was engrossed in the proof of the argument by induction, Mr. TA hoped to leave the appointment early.
As Mr. TA walked towards the door, Mr. Adviser turned on him like a madman, and demanded reasons why Mr. TA had not turned in the previous grading assignment. As a measure of his good faith, Mr. TA agreed to take an impromptu quiz, thinking it would only involve definitions of computational theory concepts.
Little did Mr. TA know that the test looked like this! Mr. Advisor also said he needed at least 4.5 points in order to pass.
Because Mr. TA did not pass the test, he was punished by having to work more during the weekend.
Mr. TA told himself: “Oh brother! I have to waste another weekend away from my friends because of this. Good thing that I had the assignments in electronic format. I can grade these papers more easily.” Mr. TA grabbed his laptop and started.
Since progress was slow, Mr. TA decided to put his thinking cap on, and then TATA!
Instead of losing time checking if references are correct, why not use theory of computation to determine where in the document the quotations start and end?
The DFA design would have three states

- In the first, read all elements in the alphabet other than " and remain in the same state.
- We would enter the second state if we read in symbol "
- Remain in the second state, for all other inputs other than "
- Move from second state to accept state on the symbol "

Therefore, Mr. TA thought of writing a program using his knowledge of Java and theory of computation in order to create a tool that would enable him to check student quotations.

Mr. TA knew that the tool would not be comprehensive, but it would allow him to obtain all web searches at the same time for some randomly selected string within quotations.
Software specifications:

* input files are text files
* valid input files contain an even number of apostrophes and do not contain two consecutive apostrophes
* attempting to open other types of files might work (e.g. java, c files), but is not supported
* on an invalid input file, program throws appropriate exceptions, but does not terminate, awaiting for the next valid input
Software Capabilities:

* opening files to inspect and modify
* checking if selected file is valid
* opening a browser window with different searches for each quotation
* text files can only be visualized in Windows, because opening Notepad is hard coded
* default browser is OS-independent
Possible Software Extensions:

* save every web page containing a search and create a database with them
* use the saved forms, by opening them in their own format and identifying the first hyperlink; open document referenced in the link
* program could be extended to check for matches between quotations and bibliography
Mr. Adviser finally discovers that in another corner of the world, someone has already proven that theory of computation should be studied by everyone, regardless of their IQ. Happy with his finding, he summarizes the entire experience for his cat.