Now, let us look at what seems to a computer as simply a stream of strings but which we can see has a structure. Let us look at the text of a play by Shakespeare.

Download the text version of [The Tragedie of Macbeth](http://www.gutenberg.org/cache/epub/2264/pg2264.txt) and compare/contrast with the data in the Terrorism data file.

In no more than 250 words, reply to the following:

What structure can you discover by examining the file of text which is the Macbeth data? How hard would it be to programmatically change the file to add machine-readable structure? Would a table or set of tables be the best structure?

Type your response in the text field and submit it to the dropbox.

It would be complicated, but not impossible, to programmatically change this version of The Tragedie of Macbeth into machine-readable format. Note this doesn’t translate to all etexts.

We could determine how many times “The Tragedie of Macbeth” shows up prior to the actual start of the story. Using that, we could program the computer to separate the story into pretext and actual story. We could take the pretext and separate it into its own set of tables by looking for any time there are greater than 2 asterisks but less than 6 and creating a table out of what is contained in that section. Greater than 2 asterisks distinguishes separate parts of the pretext, while 1-2 asterisks are used for emphasis. On the other hand, more than ~6 asterisks are used as visual formatting and don’t generally distinguish separate parts of the text. Breaking it down further, we could separate the sections of the pretext into paragraphs and then sentences if that level of granularity is desired.

For the story text, we could programmatically determine start/end of scenes by looking for lines that start with “Scena” and break those into tables. We do a regular expression search within each of those to locate when a character is speaking: “^ +[A-Za-z0-9]+.“ and separate those into smaller tables (note the quotes are mine to indicate start/end of regex, and the number is to capture the witches). Finally, we could use other regex’s to locate non-speaking parts (e.g. “Witches vanish.” = “^[A-Z][a-z ]+.$”).