How To Run Alistair

(The Rule Based Body Function Extraction Tool)

## Prerequisites

The data you intend to run on is in a directory that can be referenced with the –inputDir= option.

You have Java runtime 1.8 installed or a version of Java SDK 1.8 or higher installed and referenced in the %PATH% variable.

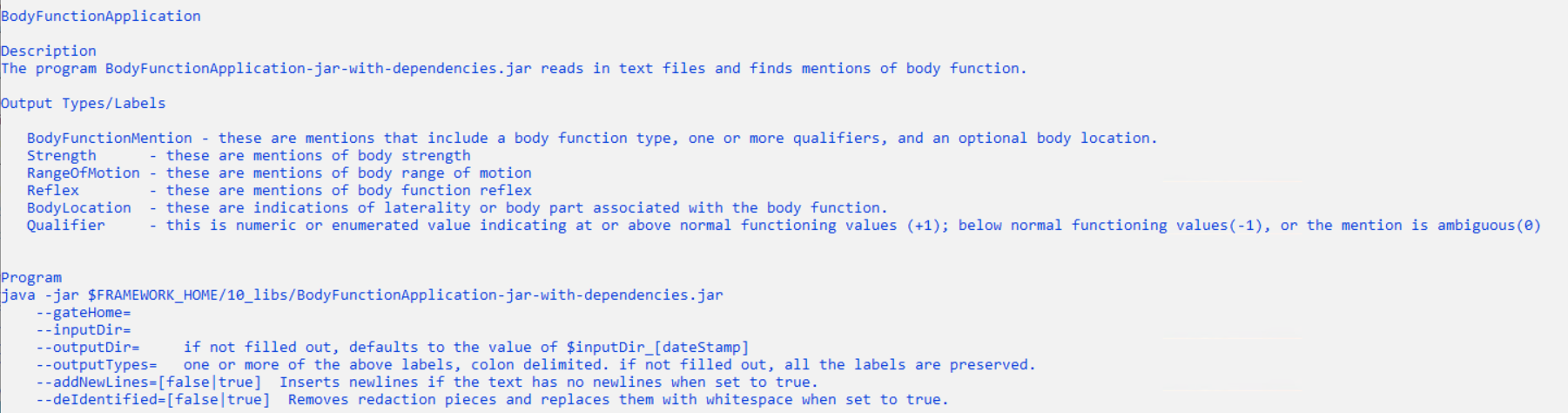
You have CYGWIN or other Borne shell scripting tool installed. (If not, use the .bat in these examples rather than the .sh script)

## Kick off the script

A wrapper script can be found at

.\600\_BodyFunction\60\_01\_libs\runExample.sh

This will kick off the program which starts with the help:



## Example Data

This example script uses data in the directory:

.\600\_BodyFunction\60\_01\_data\examples

## Output

The example script puts the output files in the directory:

.\600\_BodyFunction\60\_01\_data\examples\_output

## Viewing the Processed Files

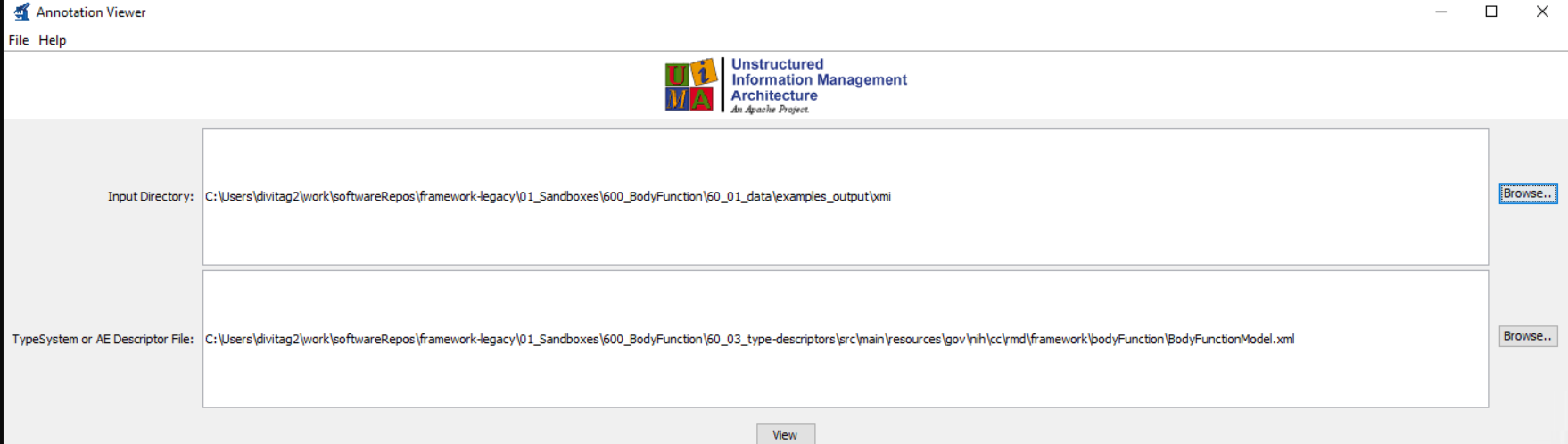
The Alastair tool, by default creates UIMA XMI, GATE xml, VTT, BIO, CSV, and just the text of the processed documents.

### UIMA

The UIMA xmi files can be viewed once UIMA has been installed. Invoke the $UIMA/bin/annotationViewer.sh

*(White lie here)*

*Correcting the white lie – you will need to add to the underlying jar, the path to where type descriptors are. This is conveniently done if you invoke this program from the eclipse IDE. (Beyond the scope of this file)*

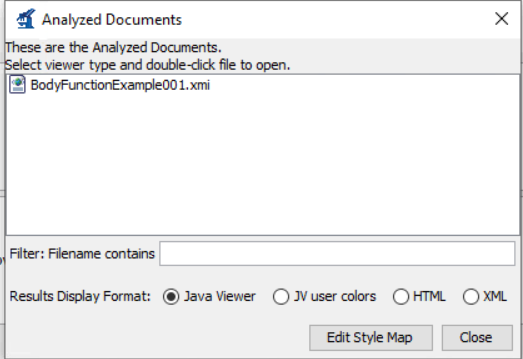


Put the full path to the directory where the UIMA output XMI files reside. In our example these are in ./600\_BodyFunction/60\_01\_data/examples\_output/xmi

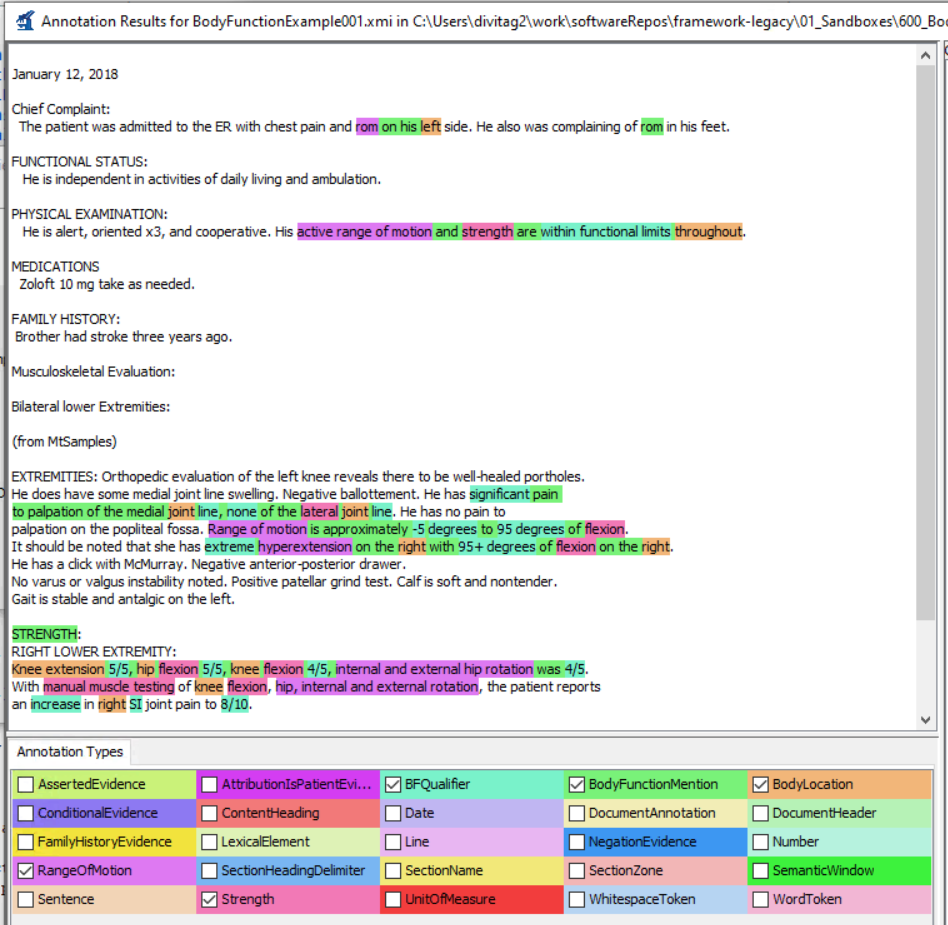
You will also need to refer to the type descriptor, located at

./600\_BodyFunction/60\_03\_type-descriptors/src/main/resources/gov/nih/cc/rmd/framework/bodyFunction/BodyFunctionModel.xml

Once these values are entered into the dialog, press the View button. This will bring up a dialog with the list of output xmi files. Select one of these files.

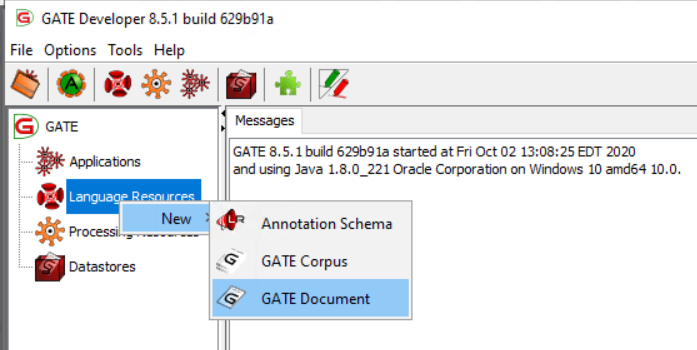


Double clicking on each of these will bring up the viewer:

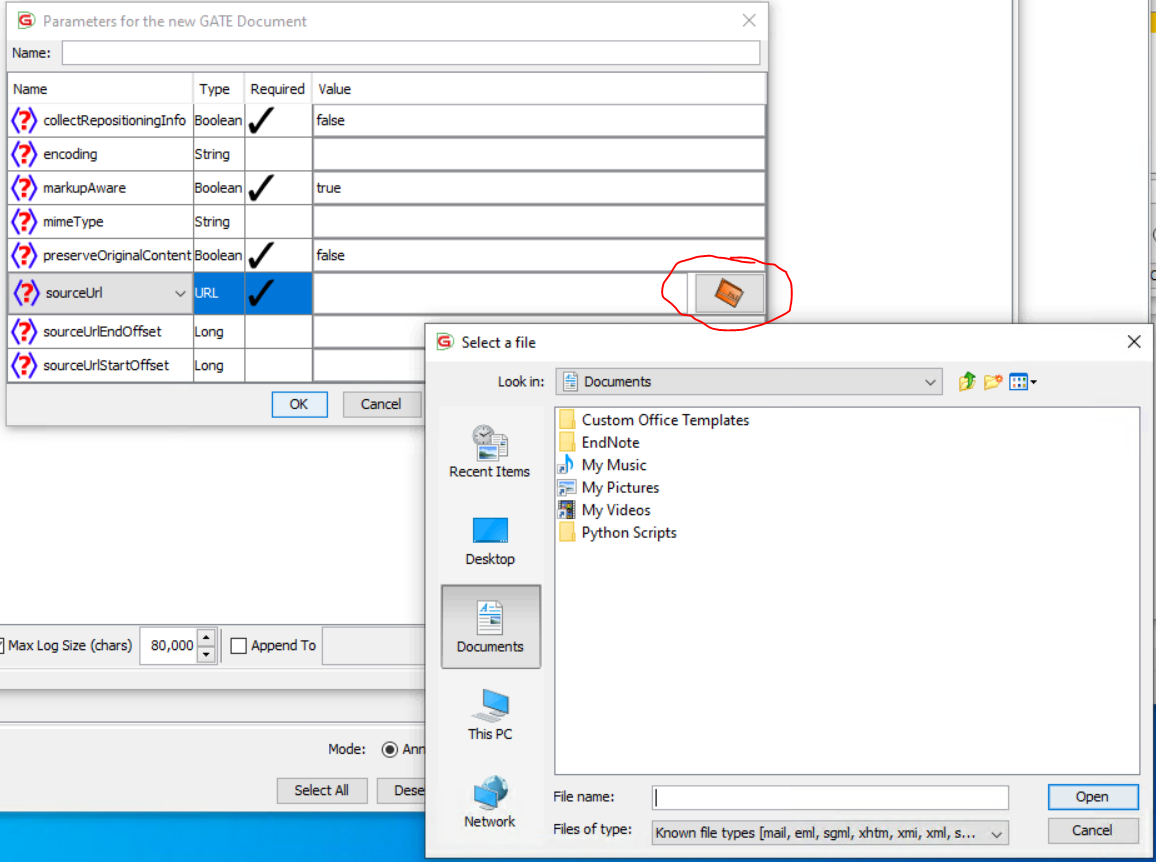


### GATE

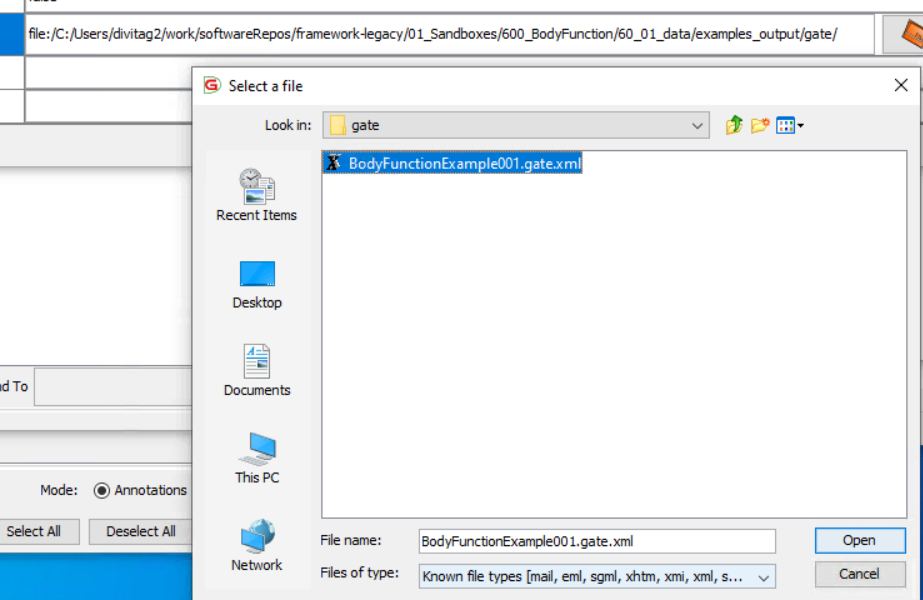
Once GATE has been installed, kick off the GATE (Developer) program. Once done, select the menu item Language Resources -> new -> GATE Document



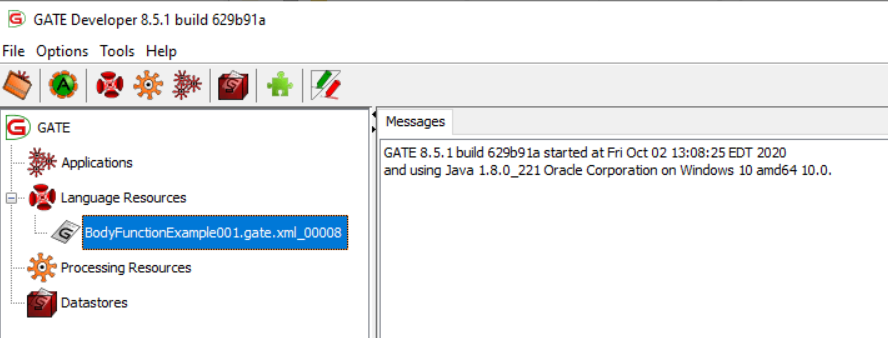
This will bring up a dialog, to which enter the location of the GATE processed document once you’ve pressed the circled icon to bring up the file dialog box.



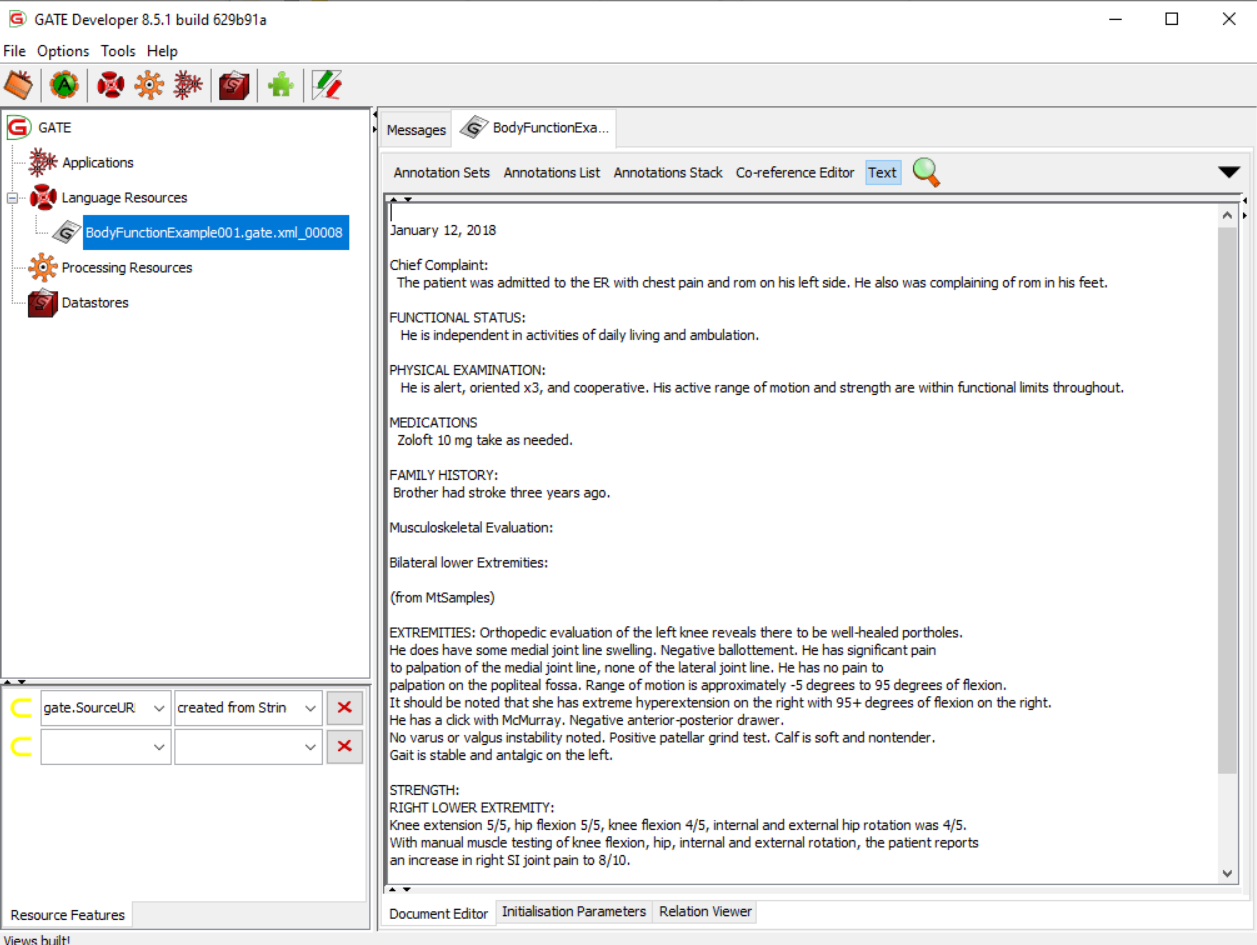
The location of the processed GATE file is at .\600\_BodyFunction\60\_01\_data\examples\_output\GATE\BodyFunctionExample001.gate.xml



This will load the processed file and make it accessible as seen in the left column of the GATE developer window.

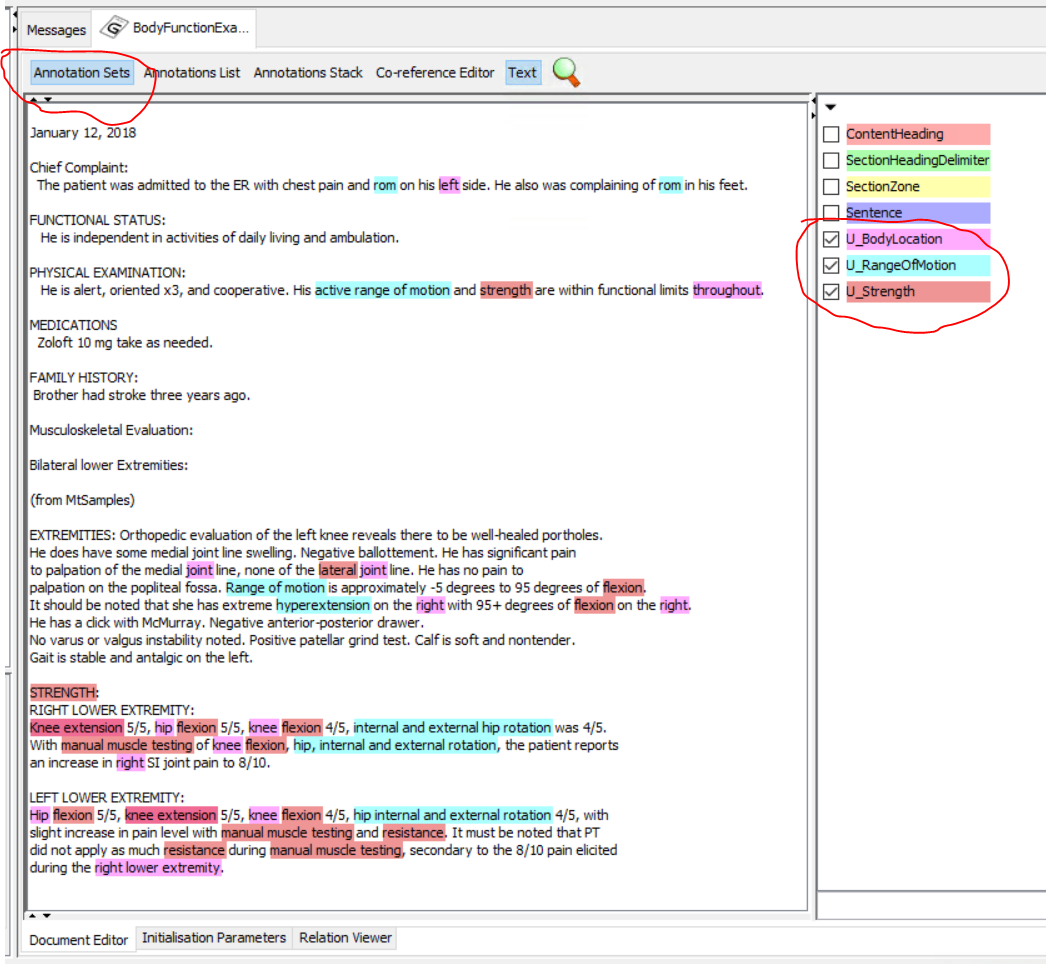


Double Clicking on highlighted gate.xml file will bring that file into the GATE editor windows on the right side of the window.



Selecting the Annotations tab at the top, and checking the annotation kinds on the right, will highlight those annotations in the text:

.\600\_BodyFunction\60\_01\_data\examples\_output\GATE\BodyFunctionExample001.gate.xml

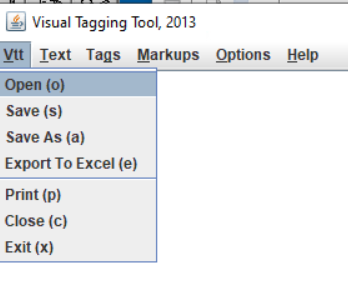


### VTT

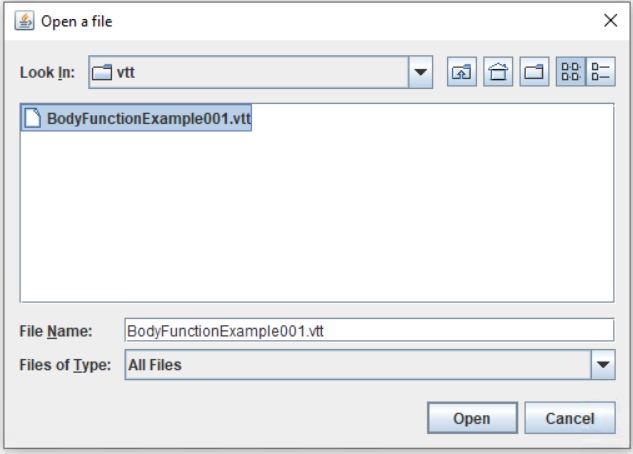
There is a copy of VTT in the 60\_01\_libs directory. Click on the vtt.bat or invoke vtt via the command line with the command

java -jar vtt-jar-with-dependencies.jar

Once the vtt window is up, select the output vtt file by selecting the vtt tab at the top left.



Navigate to the location of the output vtt file. One is located at



One is located at

.\600\_BodyFunction\60\_01\_data\examples\_output\GATE\BodyFunctionExample001.vtt

Once opened, the VTT editor will be populated with the rendered vtt file:

