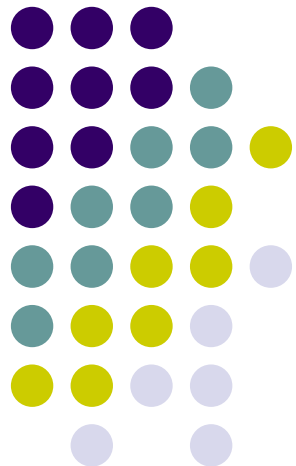
ASMURF CODER SOFTWARE TUTORIAL

Richard Golden

School of Behavioral and Brain Sciences

University of Texas at Dallas

golden@utdallas.edu

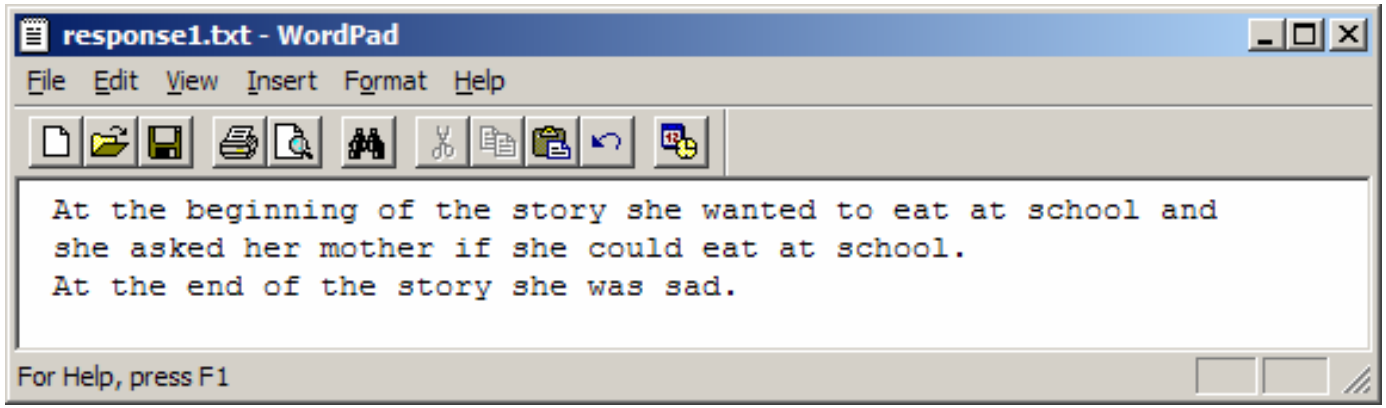


Revision Date: January 3, 2006

Supported in part by the NSF ITR Award Initiative through the Research on Learning and Education Program Award 0113369

Any opinions findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Overview of a Free Response Data Semantic Annotation Problem

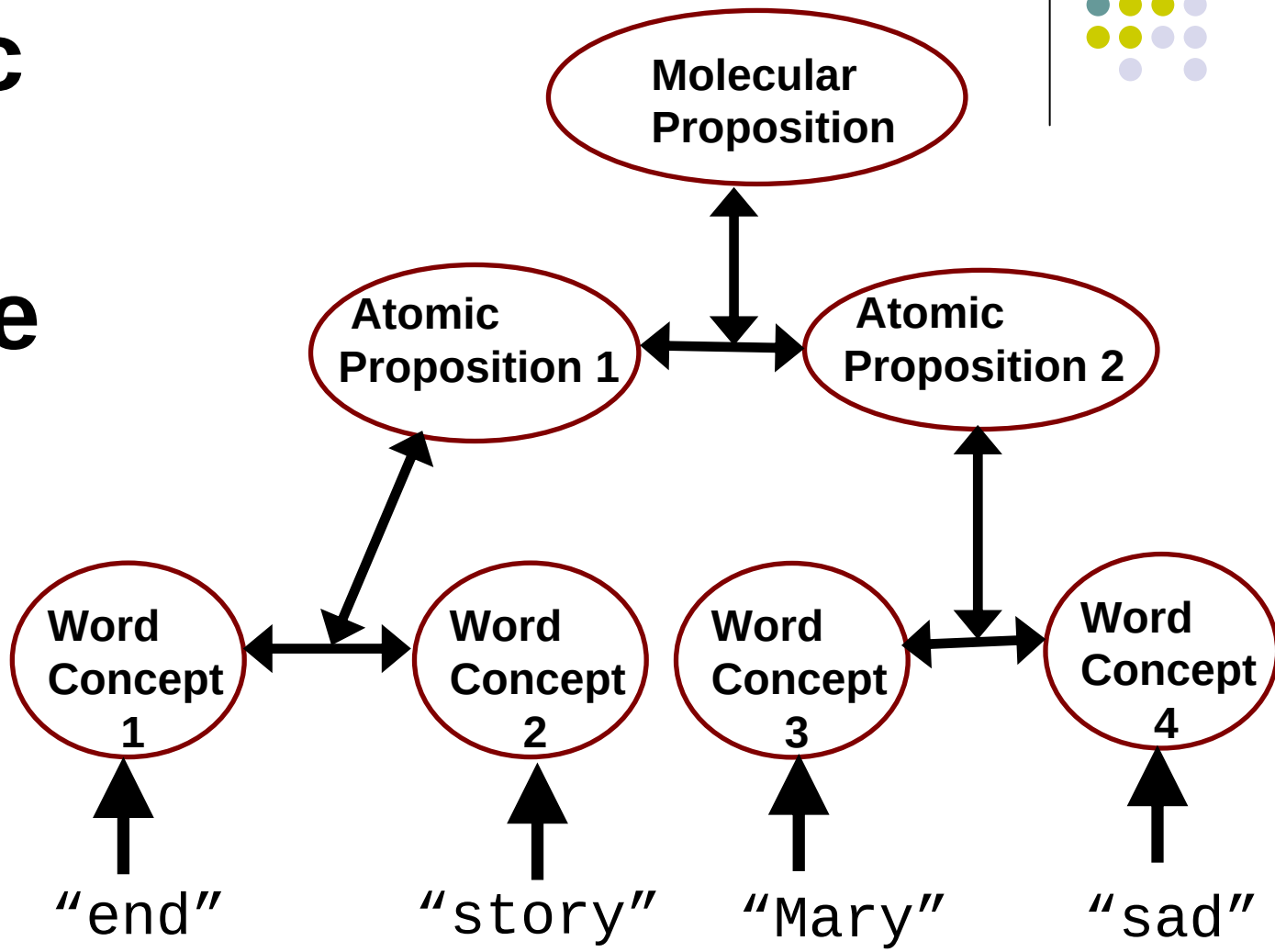


→ 2,3,5

SEMANTIC ANNOTATION TABLE

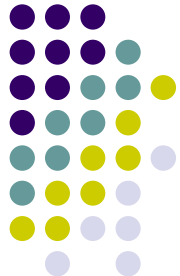
- WATCH(MARY,CRY(KIDS))
- WANT(MARY,EAT(MARY,SCHOOL),BEGINNING(STORY))
- REQUEST(MARY,MOTHER,EAT(MARY,SCHOOL))
- HAPPY(MARY,END(STORY))
- SAD(MARY,END(STORY))
- ANGRY(MOTHER)
- BRING(MARY,LUNCH,TO:SCHOOL)

Annotated Semantic Markov Utterance Random Field



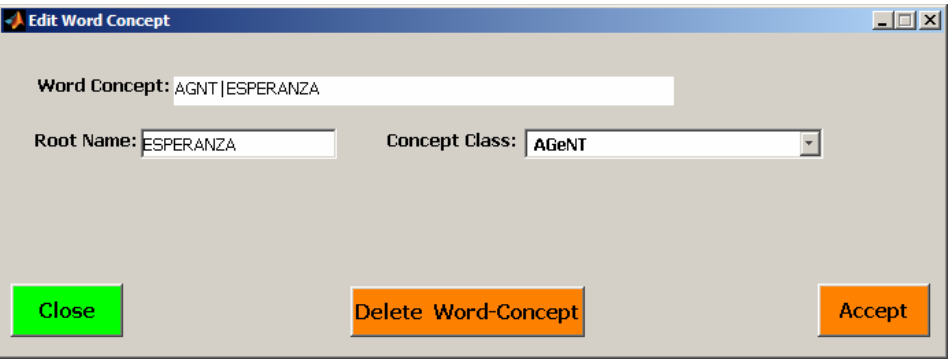
"At the end of the story, Mary was sad."

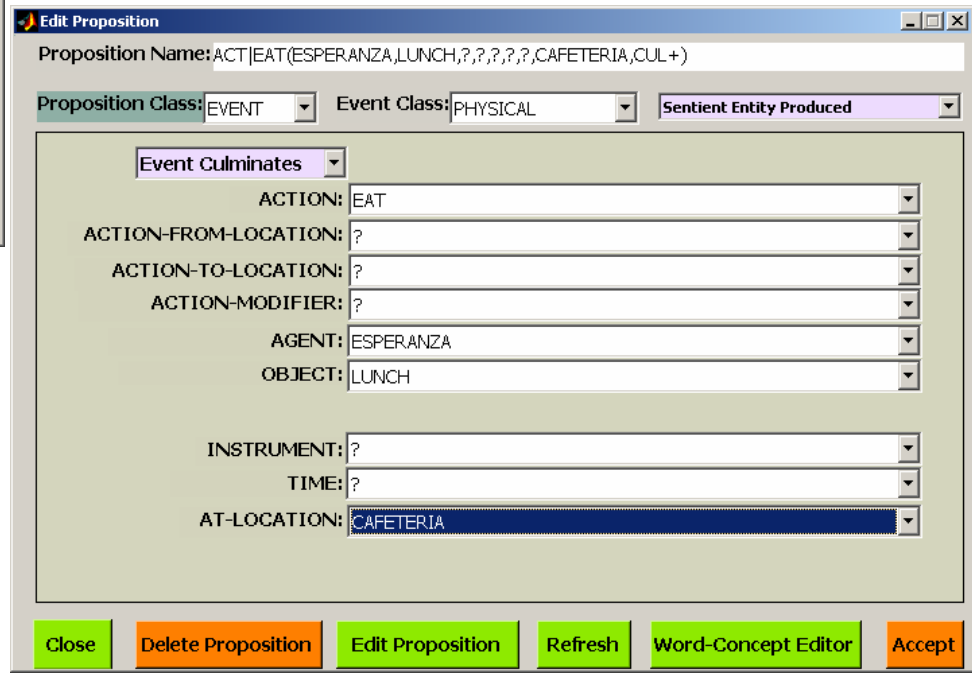
ASMURF Proposition Concepts



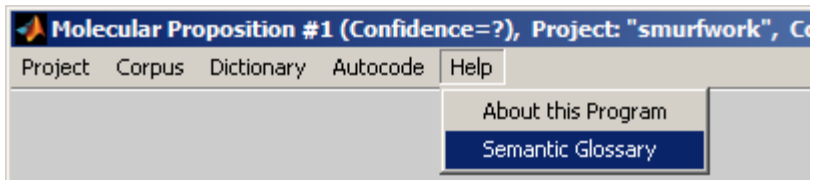
- Atomic Propositions
  - An *action atomic proposition* refers to exactly a single action. For example, the meaning of sentences such as "`Mary asked her mother`" or "`Mary is eating at school`" may be represented as action atomic propositions.
  - A *stative atomic proposition* refers to a particular situational state. For example, the meaning of sentences such as "`Mary was sad`", "`Mary was at the restaurant`", or "`Mary was sad at the restaurant`" may be represented as stative atomic propositions.
- Molecular Propositions
  - A molecular proposition is a composition of one or more atomic propositions. For example, the meaning of sentences such as "`She asked her mother if she could eat at school`" as "`At the end of the story she was sad`" may be represented as molecular propositions.
  - The objective of the "coding process" is to map free response data into an ordered sequence of molecular propositions.

ASMURF Semantic Representational System

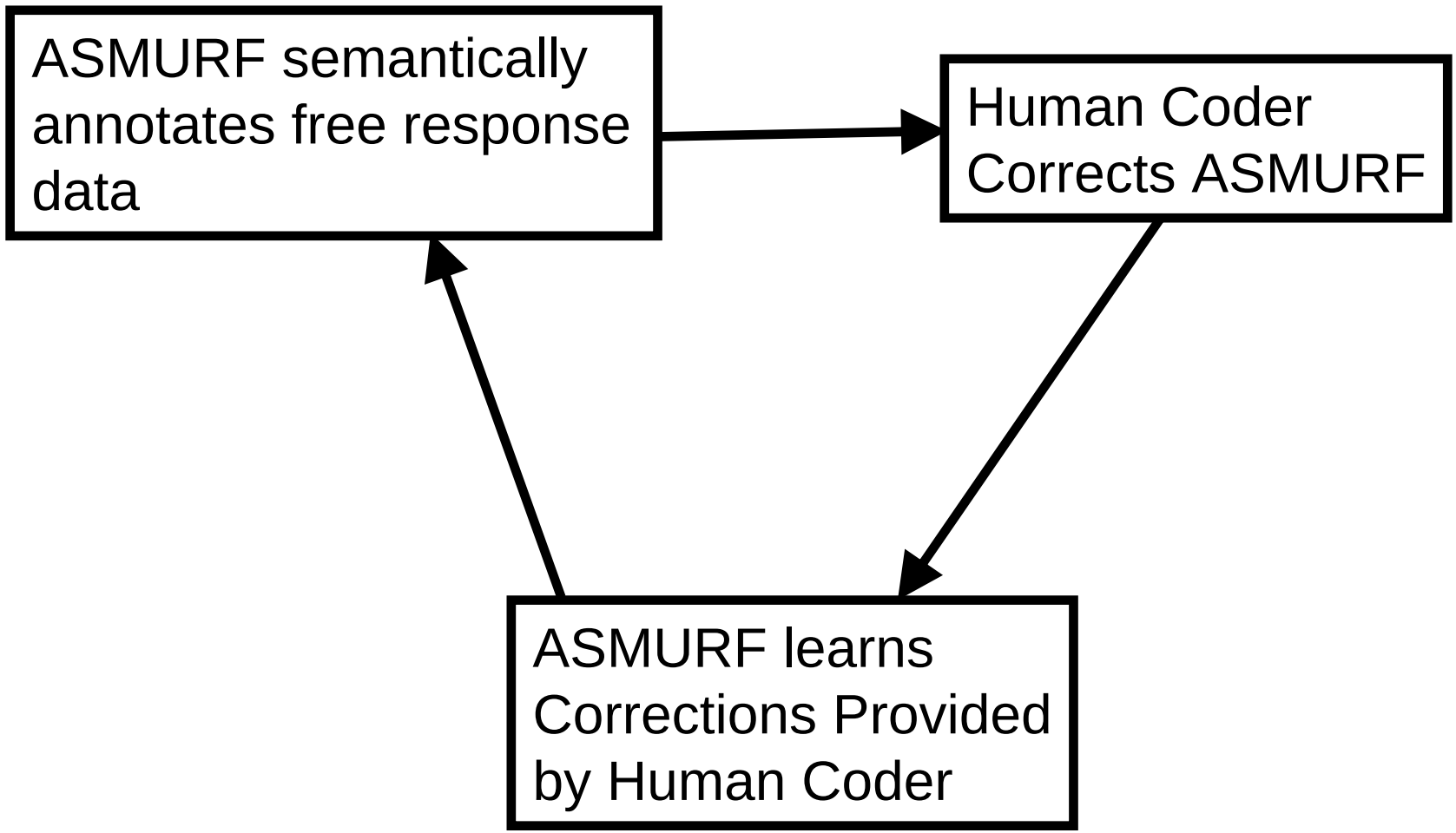




See *Help→Semantic Glossary* for additional details regarding the semantic representational system…

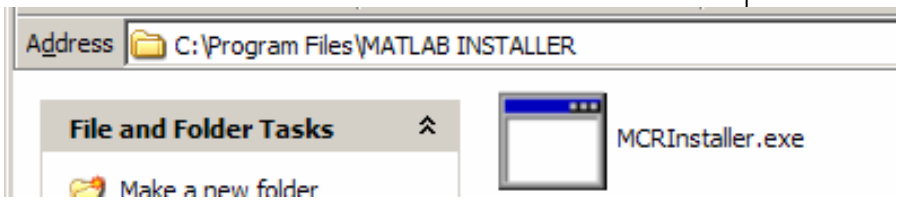


ASMURF CODER OVERVIEW

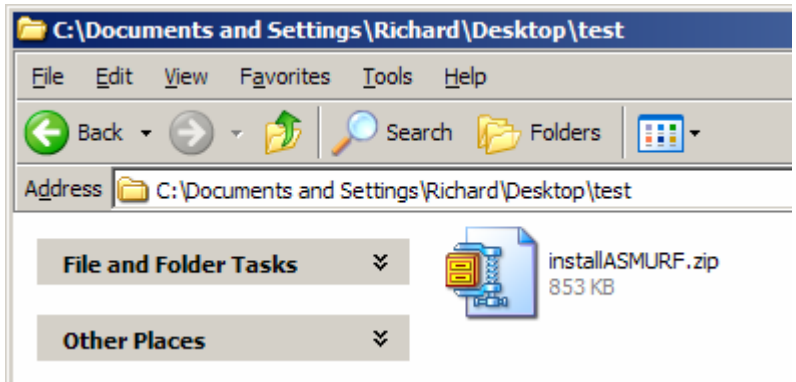


Installing ASMURF Coder on a Windows Operating System

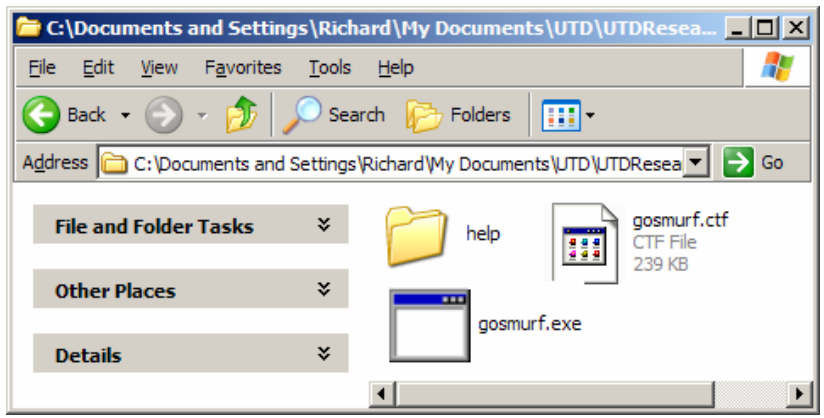
- **STEP 1:**
Move the file *MCRInstaller.exe* into the folder *MATLAB INSTALLER* in the *Program Files* folder. Then install the MATLAB Run-Time Component Library by clicking: *MCRInstaller.exe* and following the directions. Note that this step may be omitted if the MATLAB Run-Time Library has been previously installed.



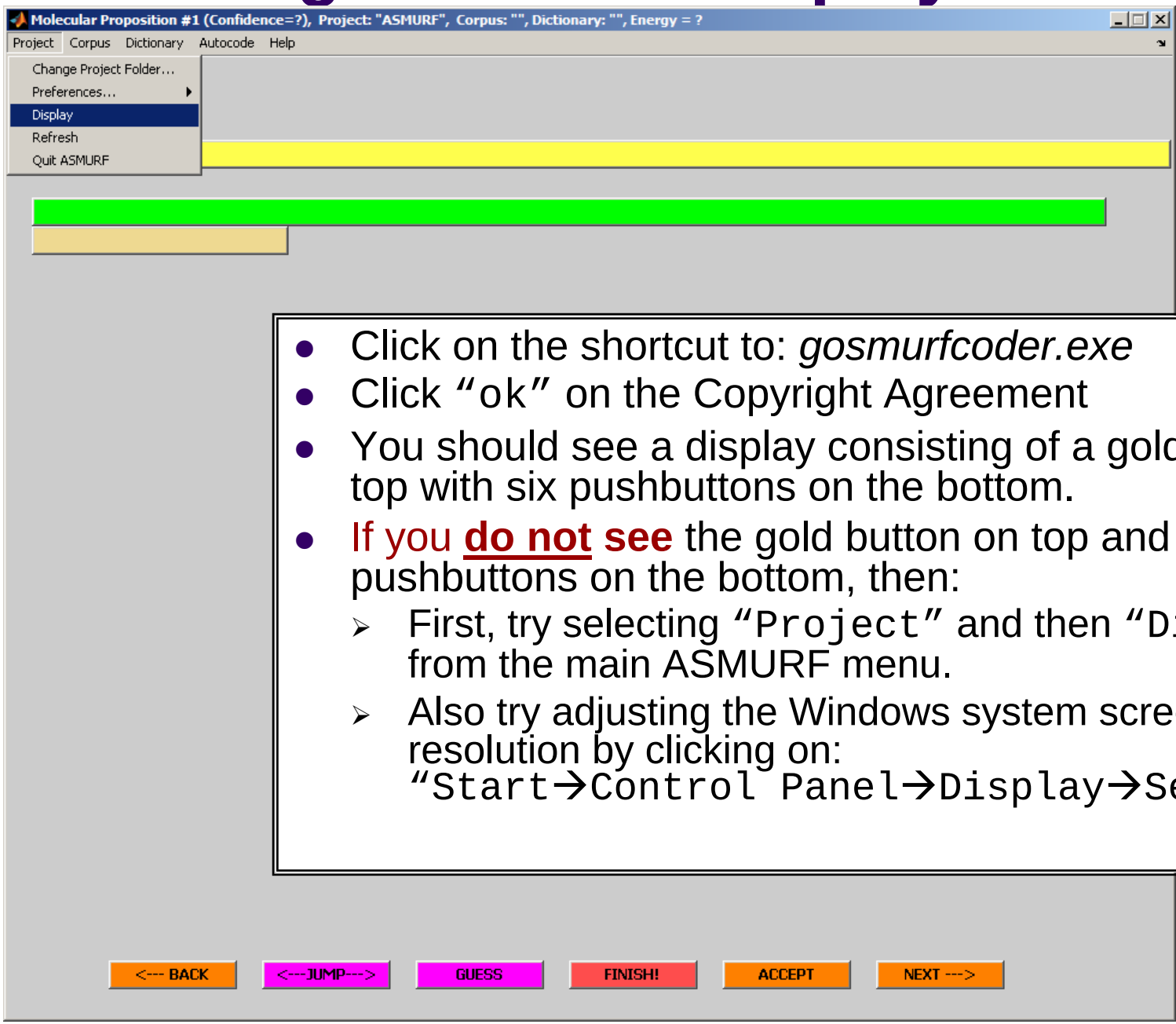
- **STEP 2:**
Unzip the file: *ASMURF.zip,* obtain the files *gosmurfcoder.exe* and *gosmurfcoder.ctf,* put both of these files in a folder called *ASMURF* located in your *Program Folder* with the *help folder. No other files or folders should be located in the folder ASMURF at this point in the installation process.*



- **STEP 3:**
Create a short-cut to *gosmurfcoder.exe* by highlighting *gosmurfcoder.exe* and right-clicking "`Create Shortcut`". You can copy and paste this short cut anywhere you wish to invoke the software. Alternatively you can click on "`Pin to Start Menu`" to access gosmurfcoder from the start ment of your system



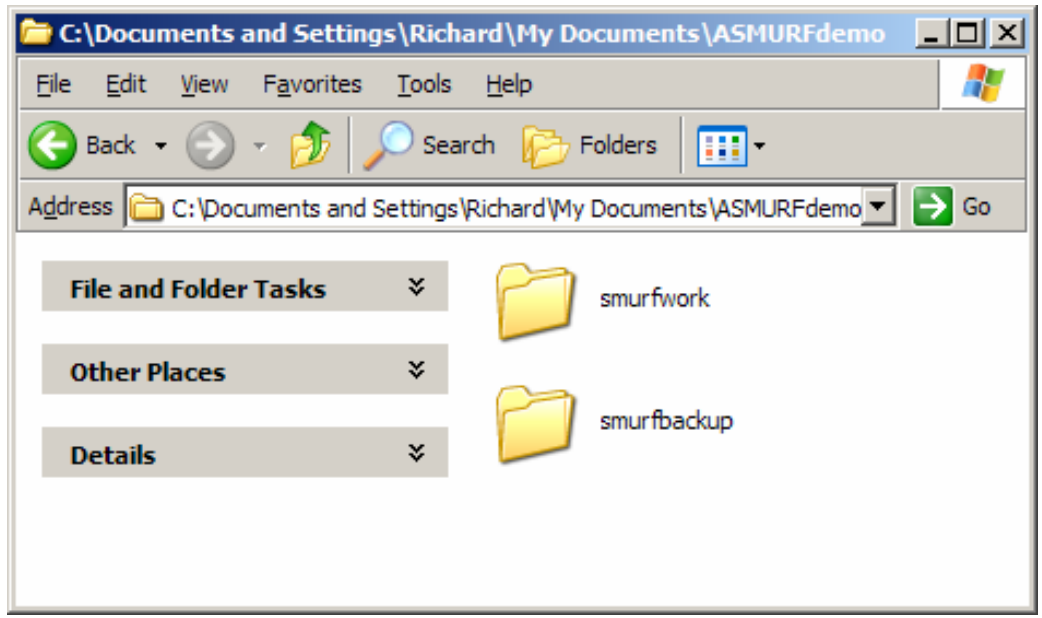
Checking ASMURF Display



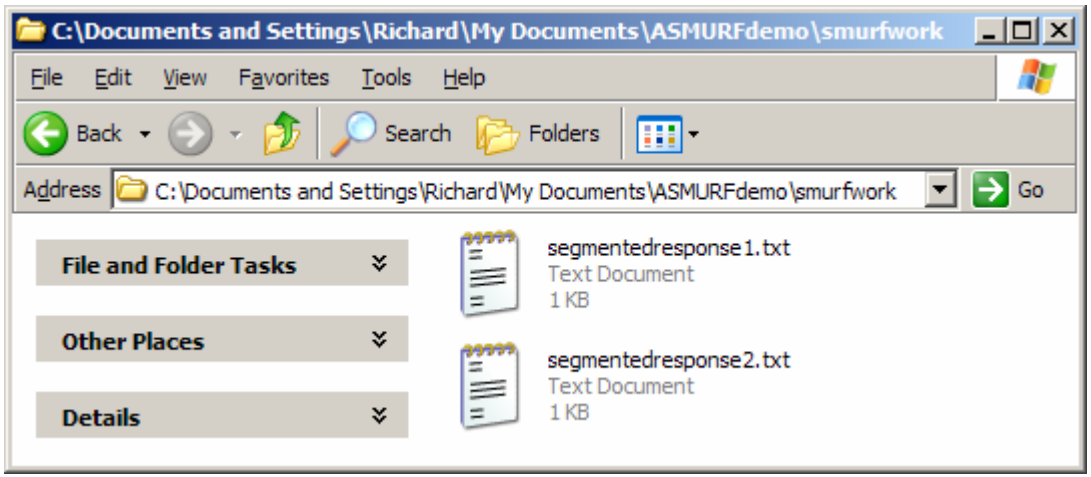
- Click on the shortcut to: *gosmurfcoder.exe*
- Click "ok" on the Copyright Agreement
- You should see a display consisting of a gold button on top with six pushbuttons on the bottom.
- If you **do not see** the gold button on top and the six pushbuttons on the bottom, then:
  - First, try selecting "Project" and then "Display" from the main ASMURF menu.
  - Also try adjusting the Windows system screen resolution by clicking on: "Start→Control Panel→Display→Settings"

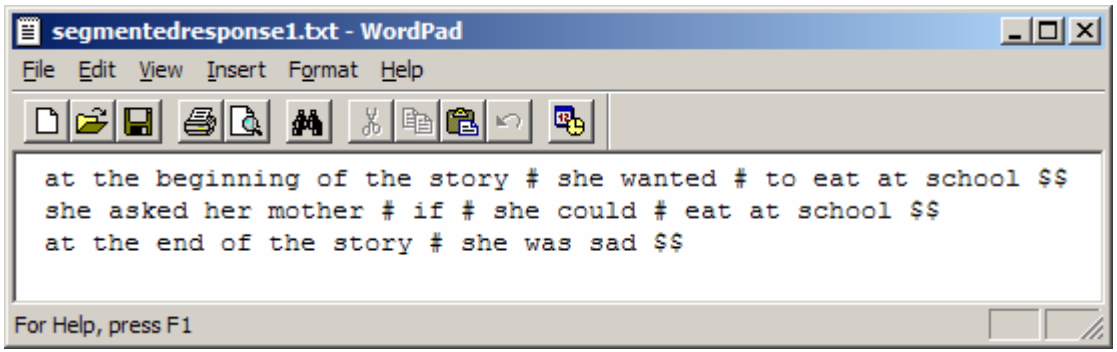
Getting Organized: Folders

- Create a folder called: *smurfwork* which contains a text file for each student's response
- Create a folder called: *smurfbackup*
- Note that the names "smurfwork" and "smurfbackup" are arbitrary. Any other folder name choice is also ok.



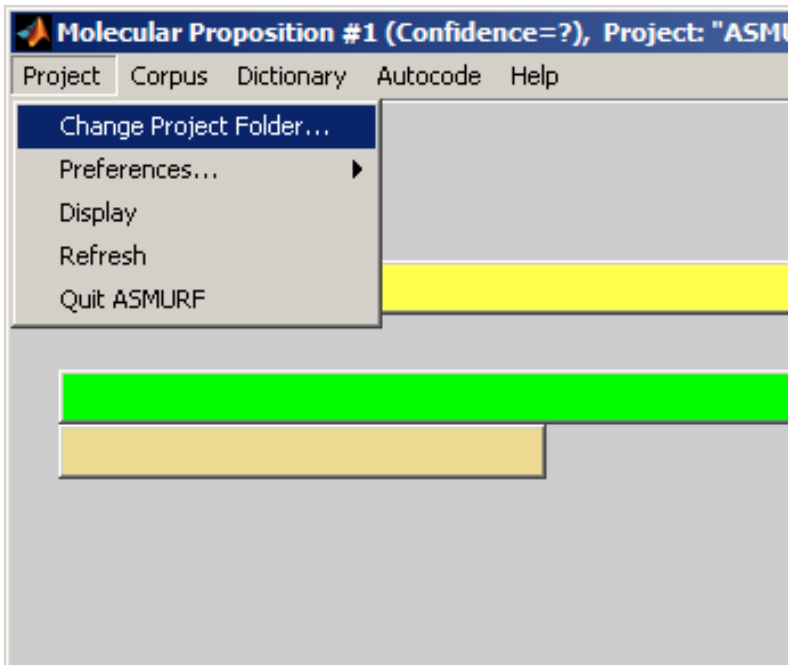
Getting Organized: Data Files

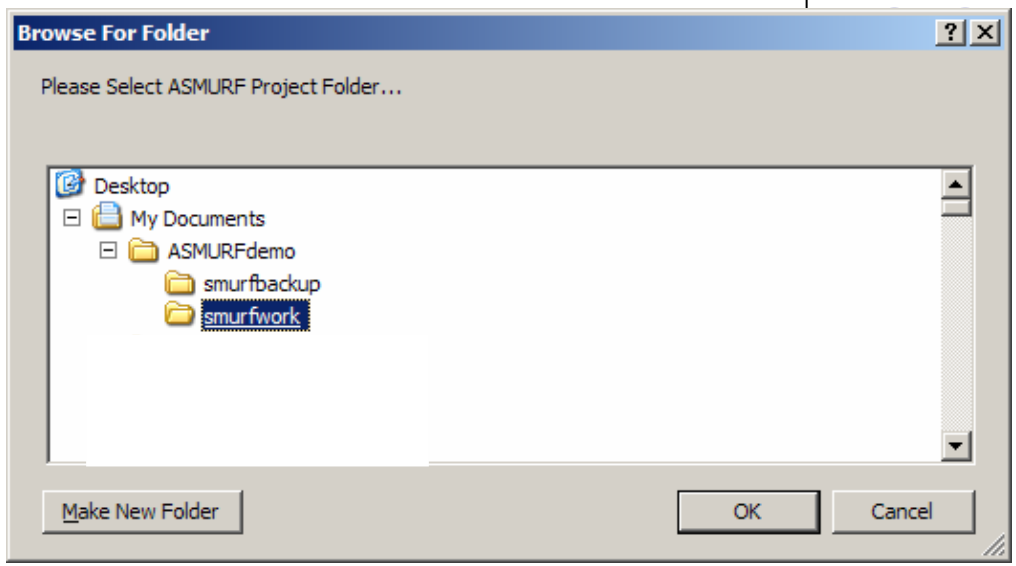




- The folder *smurfwork* contains segmented data files of student responses which will be used to "train" ASMURF
- These text files may be created/edited using *"wordpad"* in Windows OS
- # symbol separates atomic propositions within a molecular proposition
- *$$* symbol separates molecular propositions

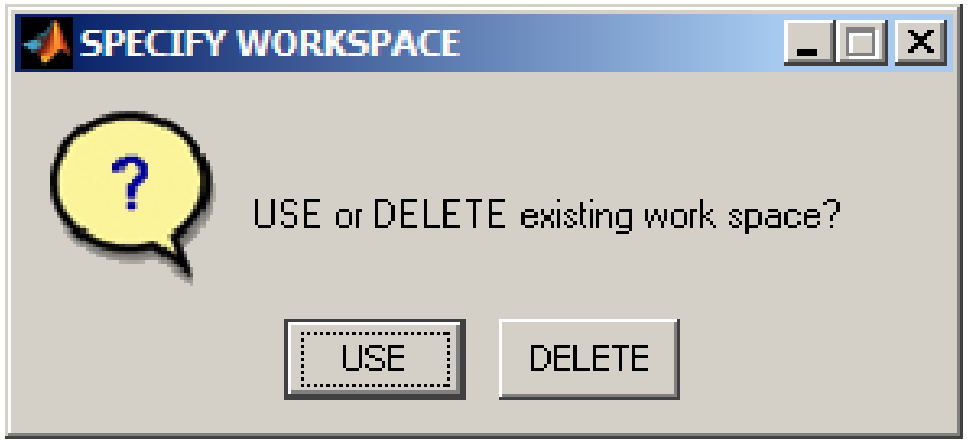
Selecting the Project Folder…





- Use the ASMURF menu to set the Project Folder to the folder containing your segmented response data files using the command sequence: “`Project→Change Project Folder…`”

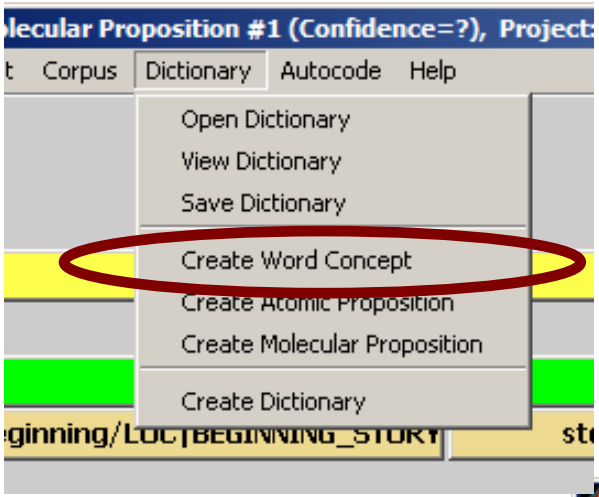
The “Specify Workspace” Option
(appears after using: `Project→Change Project Folder`)



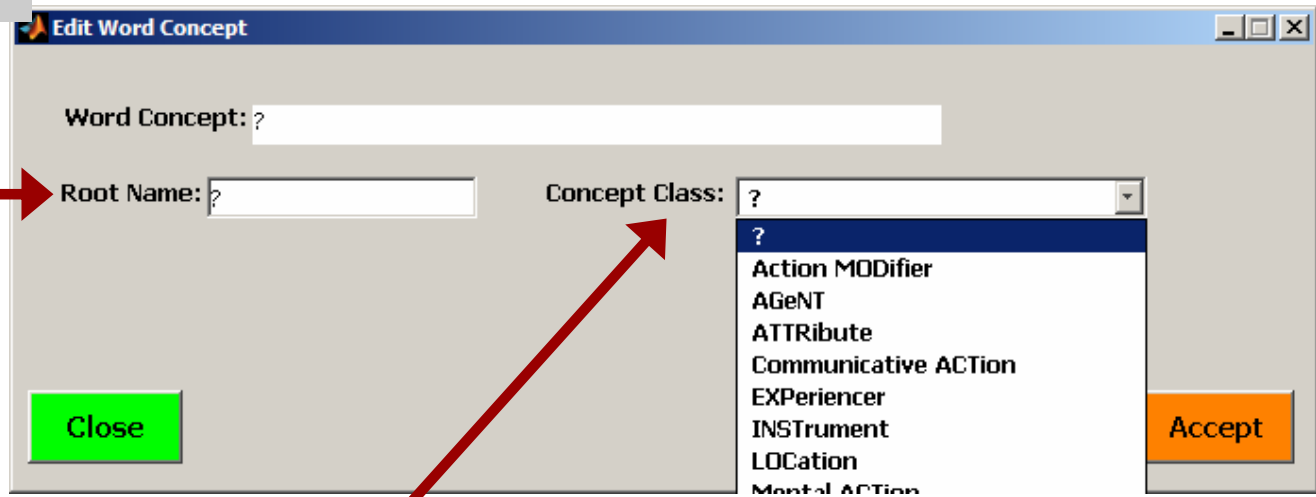
- If you choose “use” then the current workspace can be used to make inferences but you risk modifying the “ecoded files” in that workspace. This is a good option to choose during the coding process if you are interrupted from your work.

- If you choose “delete” then you will need to load a new dictionary (workspace) using:
`Dictionary→Open Dictionary`

Adding Word Concepts...



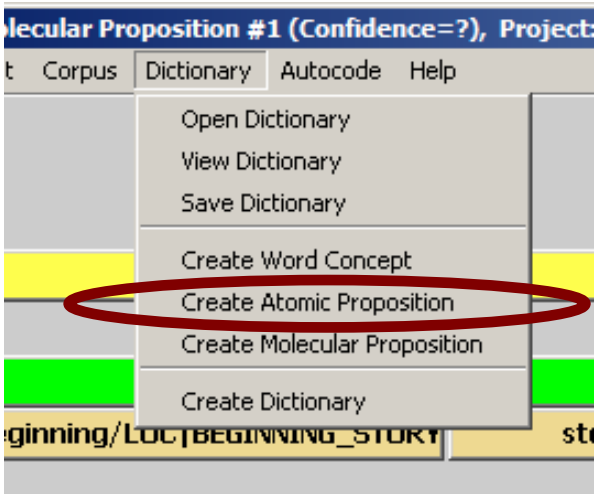
Use `Dictionary→ Create Word Concept` to add a word concept

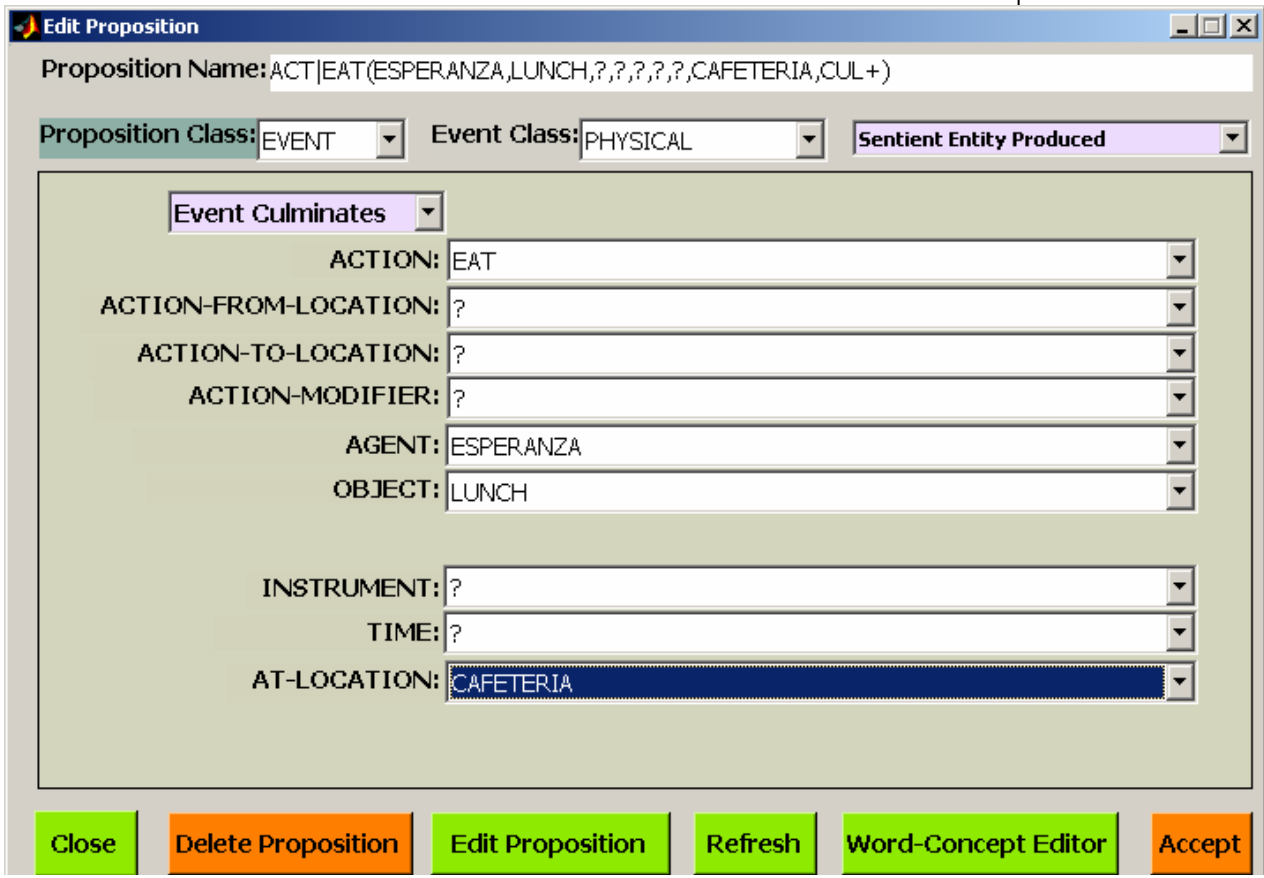


`"Root Name"` Is the concept Name without Its class designator

Assign the "rootname" a concept class (see `Help→Semantic Glossary`)

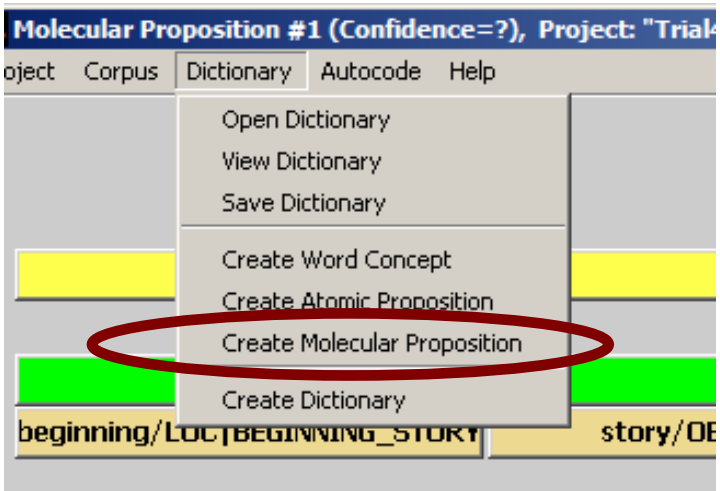
Adding Atomic Propositions…

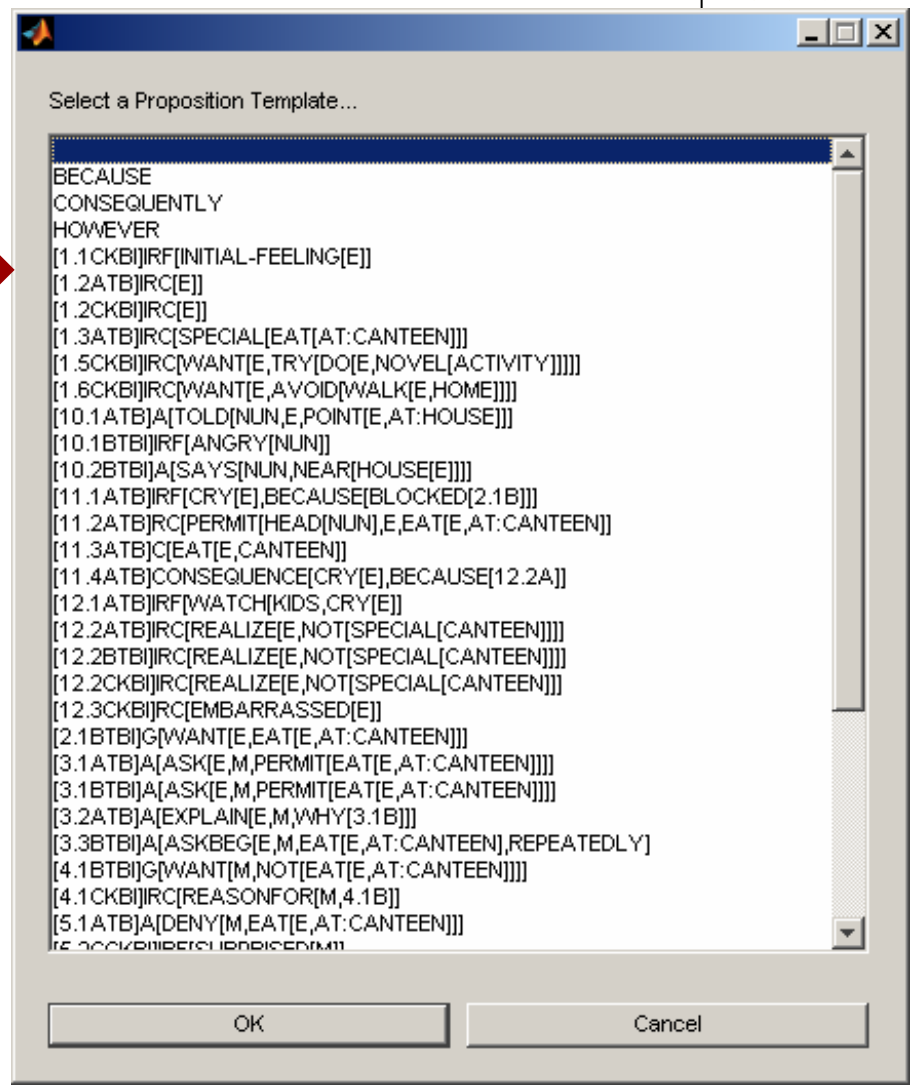




- Use `Dictionary→Create Atomic Proposition` to add atomic propositions
- Use `Help→ Semantic Glossary` to learn more about semantic representations

Creating Molecular Propositions...





- Each molecular proposition is placed on a separate line.
- Molecular propositions are typically created from a discourse analysis and are not created by the coders.

Using ASMURF to Assist in Coding Free Response Data

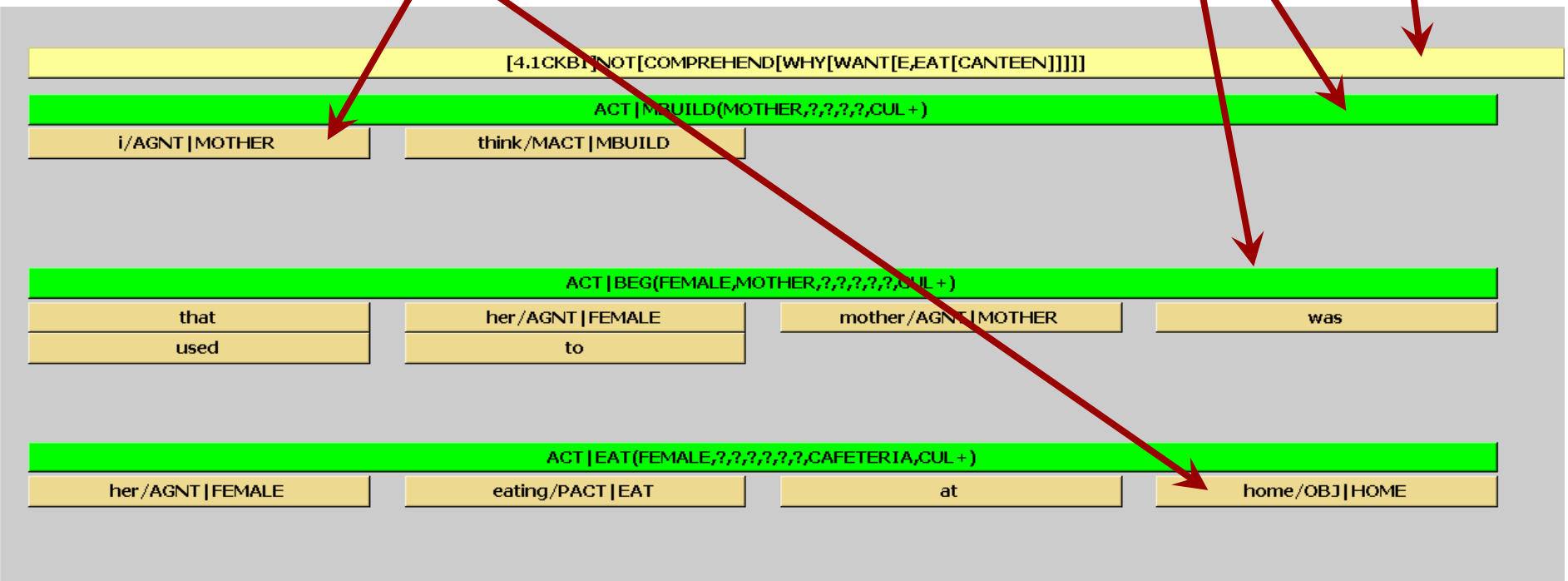
- Guess: Request ASMURF to guess
- Accept: Request ASMURF to learn
- Jump: Go to a molecular proposition without guessing
- Next: Accept, Jump to Next, Guess
- Back: Accept, Jump Previous, Guess
- Finish: Use to Save your Work and Get Next Participant's Data for Coding

Annotating Free Response Data in ASMURF Environment

Click on *the* *molecular-concept button* to change the *molecular proposition*

Click on an *atomic proposition button* to change its *atomic proposition*

Click on a *word-concept button* to change its *word concept*



Using Accept Button to Teach ASMURF to Code...

The 

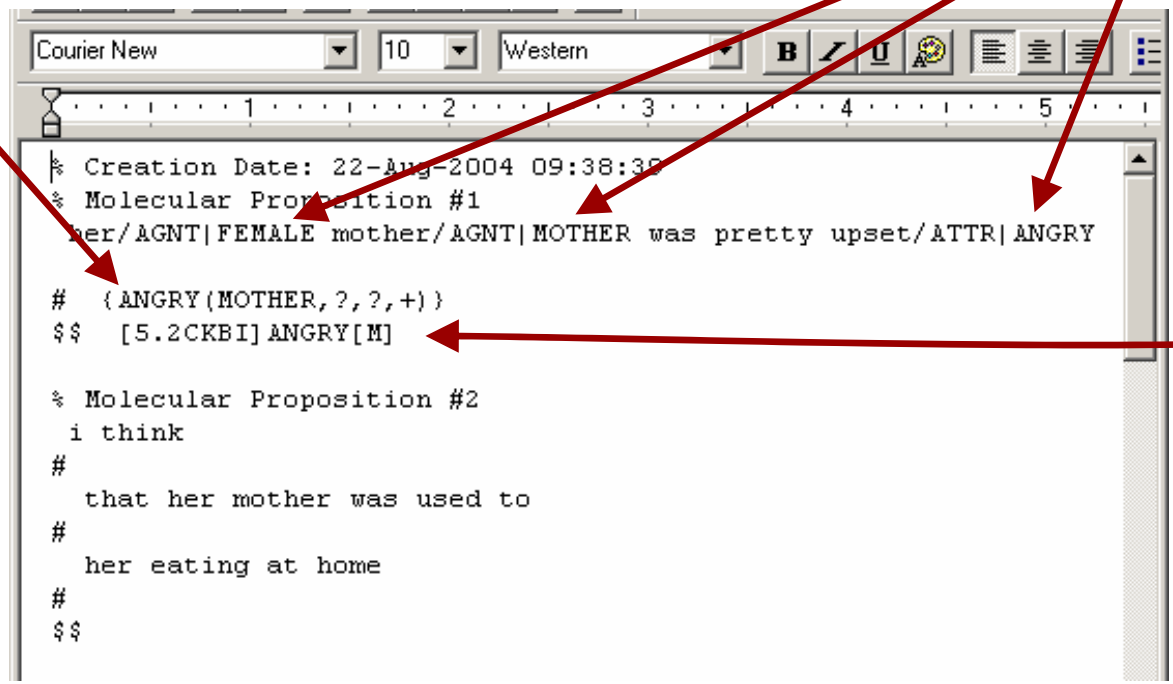
button tells ASMURF to update the system with the semantic annotations you have given ASMURF (updates the “ecoded” file)

To prevent loss of work, make sure to regularly make a backup copy of the *smurfwork* folder and put that backup copy with a timestamp into the *smurfbackup* folder

Interpreting the ECODED file...

The ecoded file with the *.ecoded* suffix contains the coding of the *.txt* file!

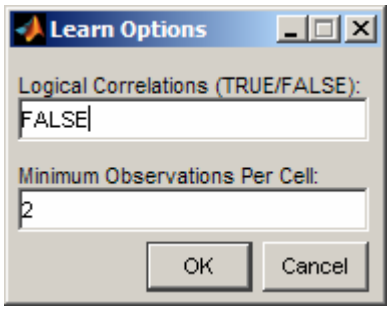
It provides explicit documentation of the assignment of *word-concepts*, *atomic propositions*, and *complex propositions.*



Always make backup copies of your ECODED files!!!!!!

Creating a Dictionary from Ecoded Files…

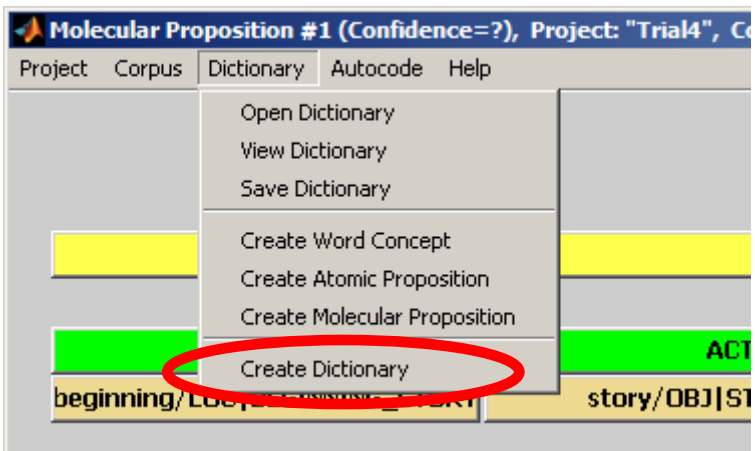
STEP 1: Set the `Learn Options` using: `Project→Preferences→Learn`



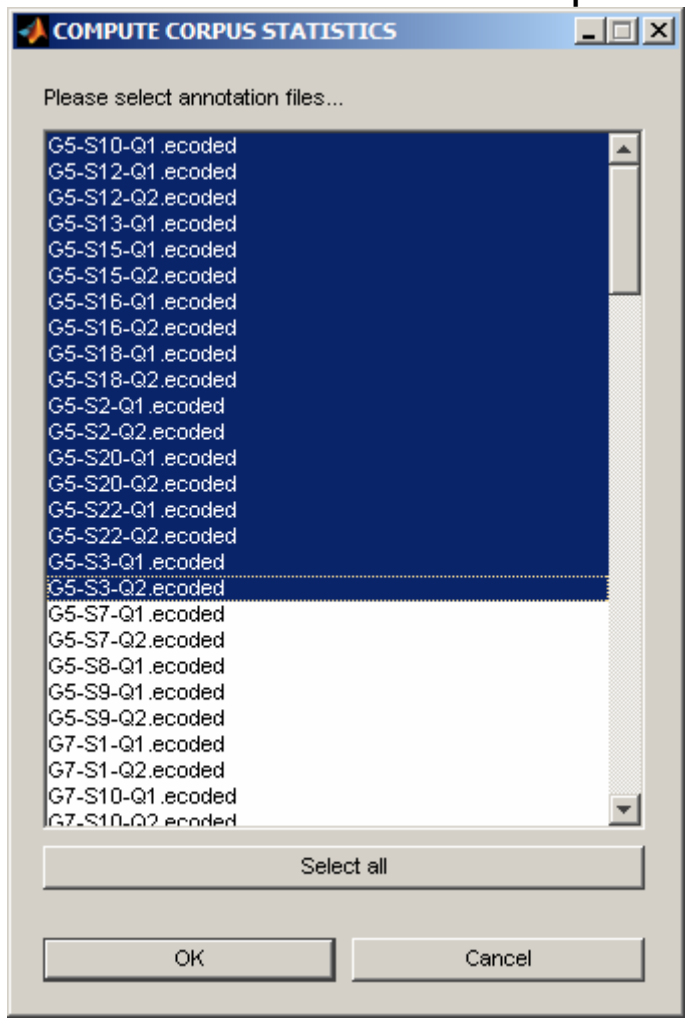
"FALSE" means look for Presence or Absence of Association and Ignore Associative Strength

"2" means look for at least 2 examples Before declaring an Association Exists

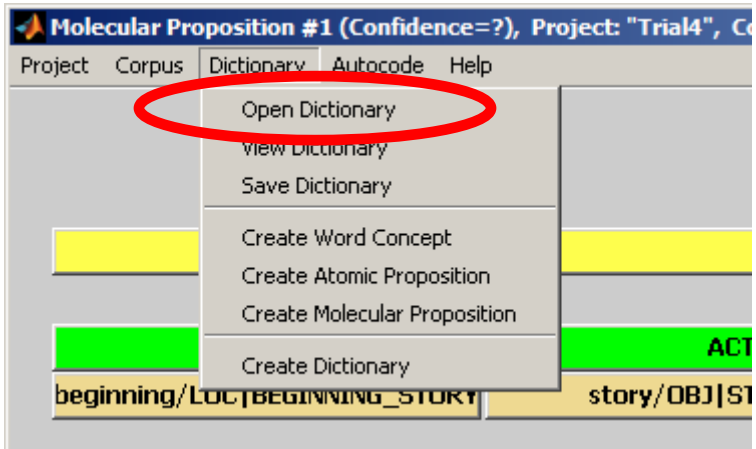
STEP 2: Compile dictionary using: `Dictionary→Create Dictionary`

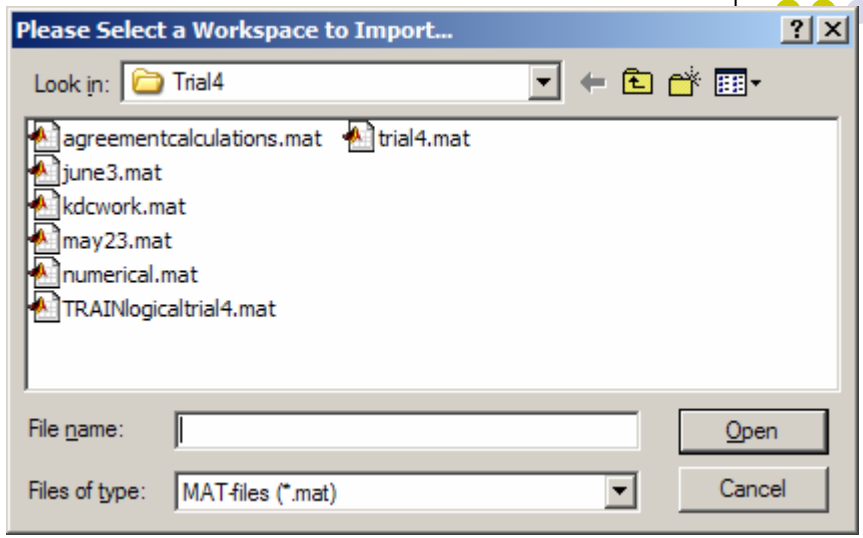


STEP 3: After naming dictionary, select ecoded files to be compiled…



Loading a Work Space (Dictionary)...

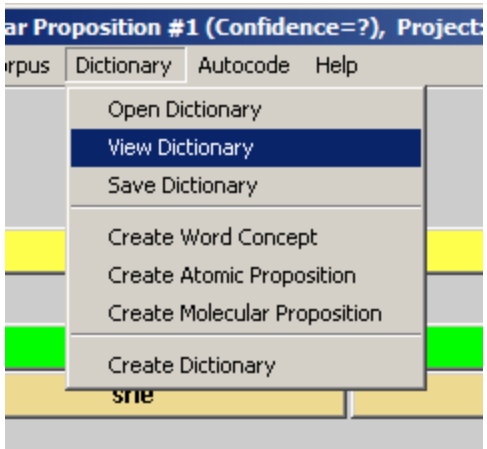




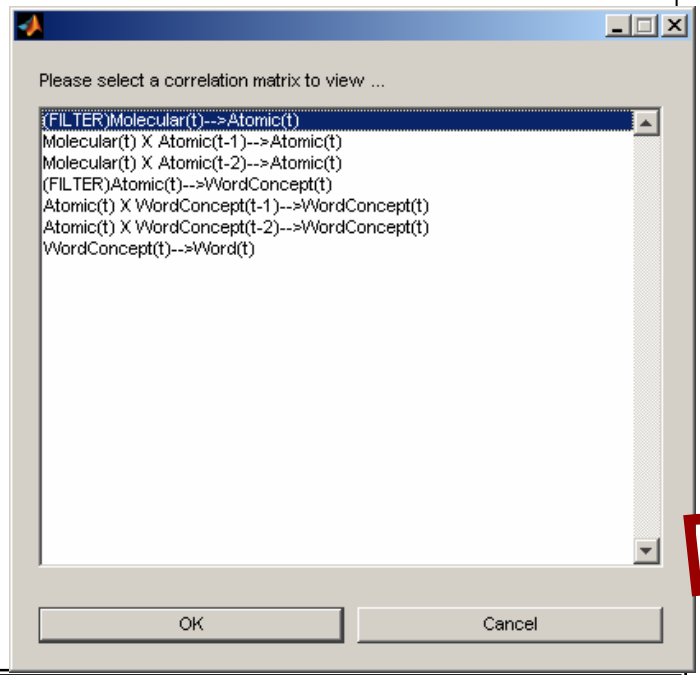
- The “workspace” contains the system’s knowledge of how to code response data. It is sometimes referred to as a “dictionary”.
- Your current work is always saved in the workspace called: *kdcwork.mat*
- When you save a “dictionary” it is stored as a “workspace” as well.
- Don’t worry too much about backing up “dictionaries” since they can always be reconstructed from ecoded files.

Viewing Correlations Learned by ASMURF

Step 1: Use:
`Dictionary→View Dictionary`



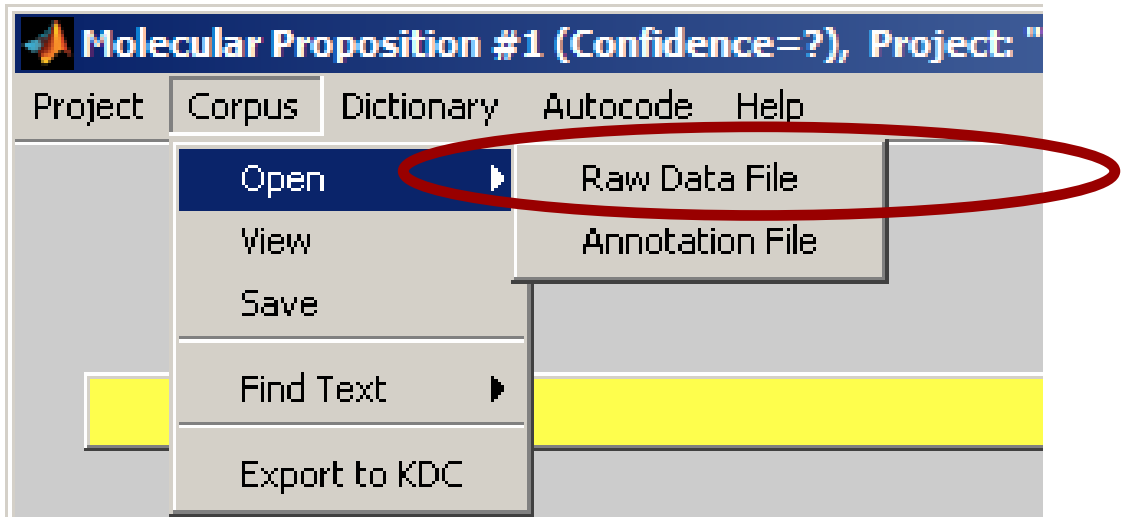
Step 2: Select Correlation Type




```
CORRELATION TYPE SPECIFICATION:
   Molecular(t) X Atomic(t-1)-->Atomic(t)

R = 0.95
{M5:[1.1CKBI]IRF[INITIAL-FEELING[E]]}--> {M2:BECAUSE}

R = 0.89
{M25:[3.1BTBI]A[ASK[E,M,PERMIT[EAT[E,AT:CANTEEN]]]]}-->
   {M2:BECAUSE}

R = 0.88
{M30:[5.1ATB]A[DENY[M,EAT[E,AT:CANTEEN]]]}--> {M2:BECAUSE}
```

"R" Is Correlation Value

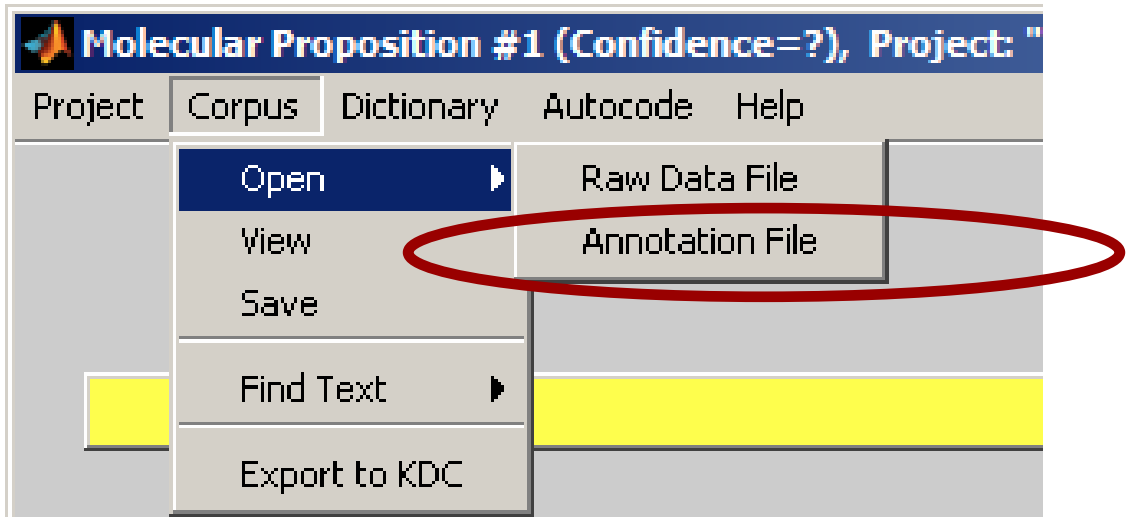
Loading a Segmented Text File



- After you have loaded the workspace, load in a text file of a participant's response data which has been segmented using the command: `Corpus→Open→Raw Data File`

- **Warning!** Choosing this option will *destroy* the corresponding "ecoded" file since it "clears" the annotations upon loading of the raw data file!

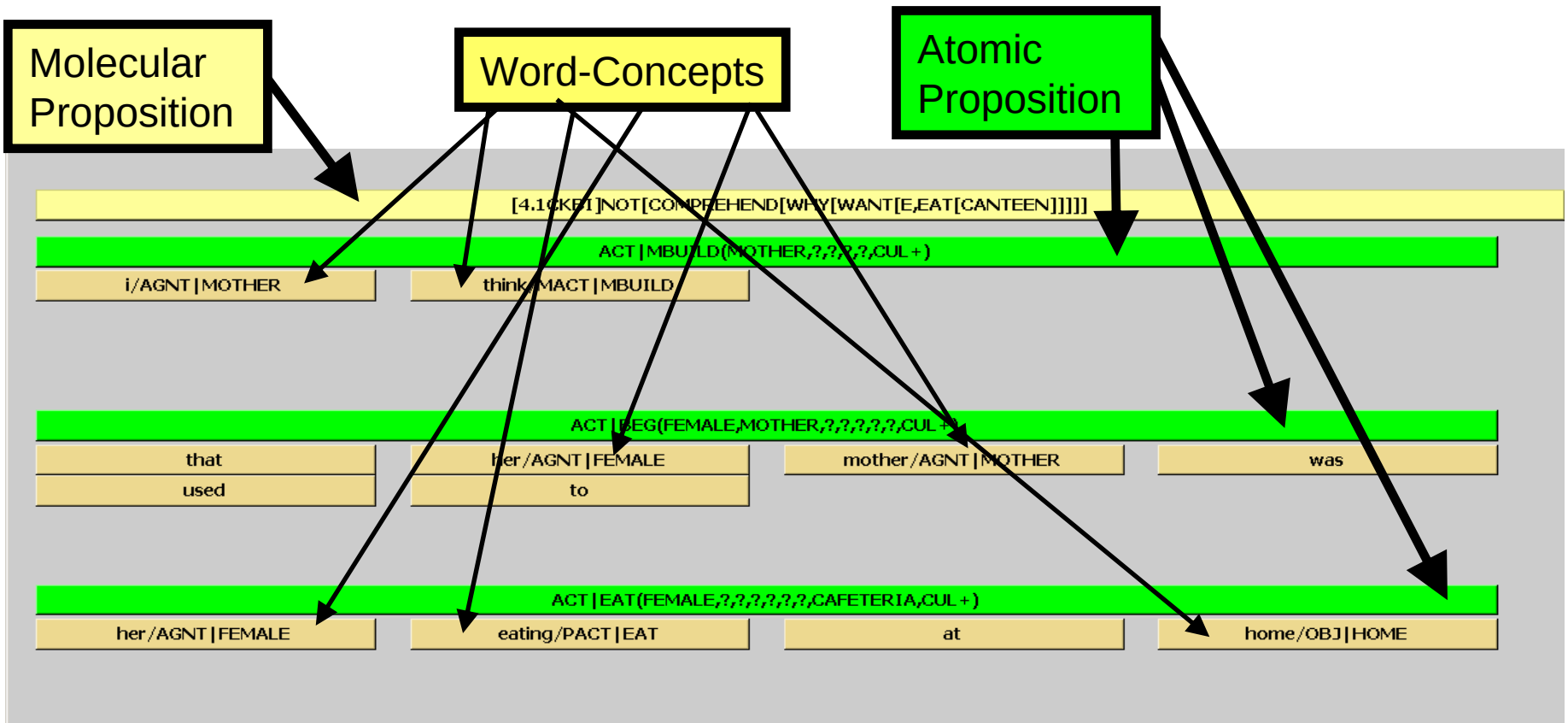
Loading an Ecoded File



After you have loaded the workspace, load in a text file of a participant's response data which has been segmented using the command:
`Corpus→Open→Annotation File`

Requesting/Interpreting an ASMURF Guess...

Select the GUESS button (also remember system guesses automatically if <--- BACK or NEXT ---> is used....use <---JUMP---> to move without guessing)

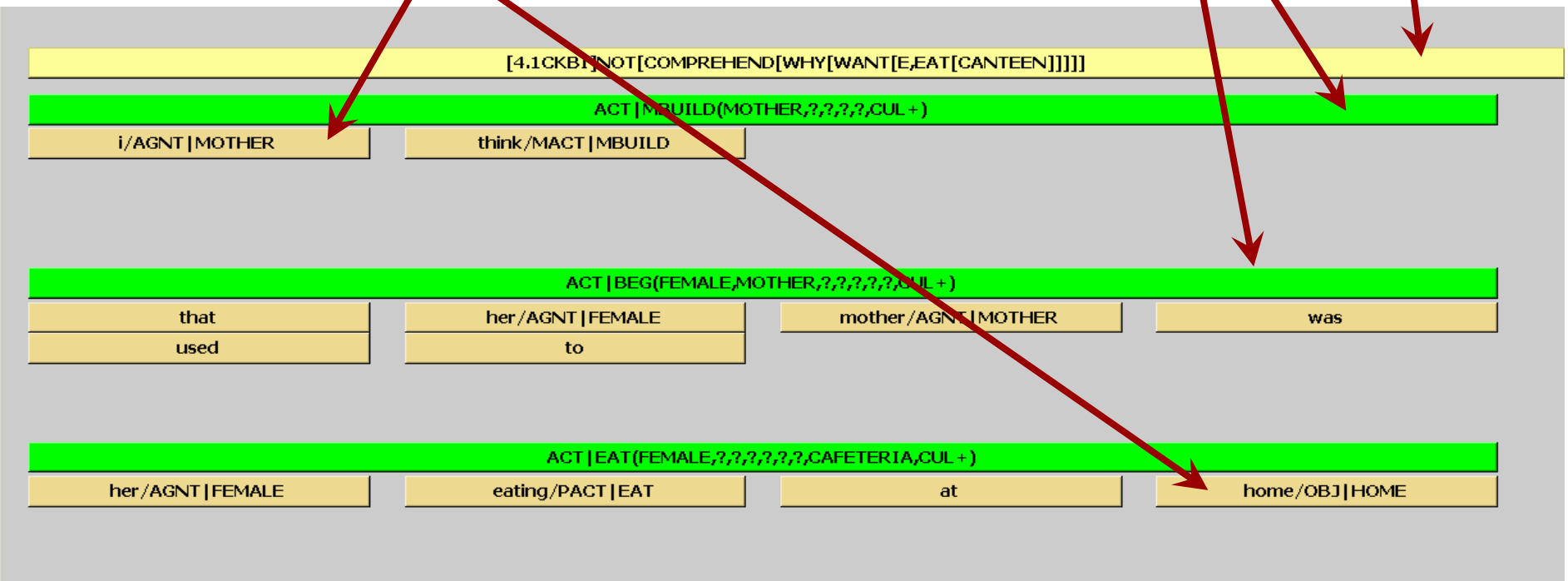


Correcting ASMURF's Guess…

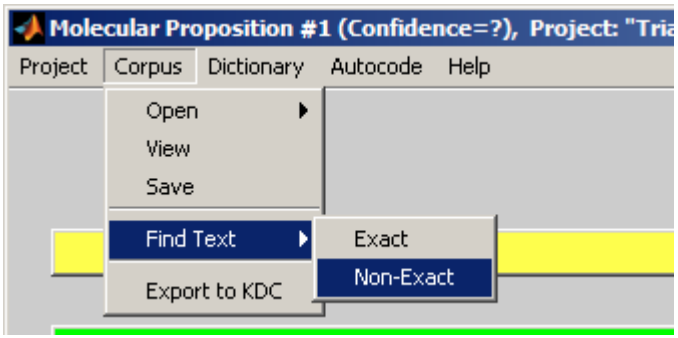
Click on *the* ***molecular-concept button*** to change the *molecular proposition*

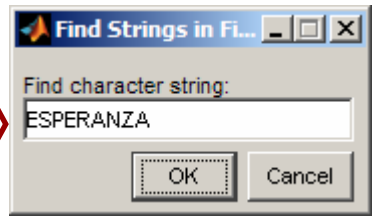
Click on an ***atomic proposition button*** to change its *atomic proposition*

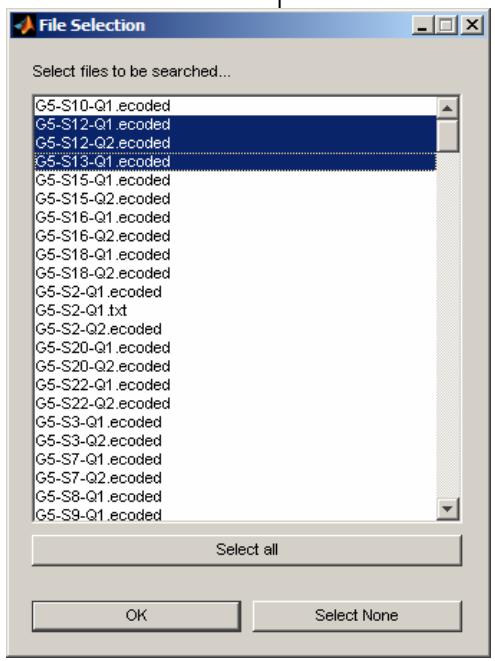
Click on a ***word-concept button*** to change its *word concept*



Searching a Folder of Ecoded Files using "Find" Command



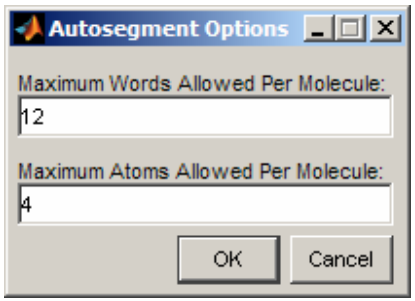




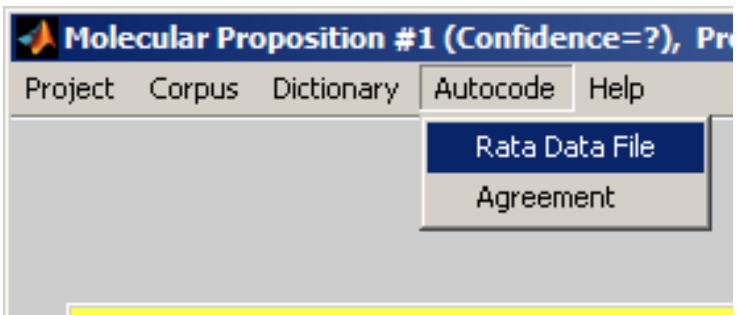
"G5-S12-Q1.ecoded" (Line #28): FOUND--> "ESPERANZA"
"G5-S12-Q2.ecoded" (Line #19): FOUND--> "ESPERANZA"
"G5-S13-Q1.ecoded" (Line #5): FOUND--> "ESPERANZA"
"G5-S13-Q1.ecoded" (Line #22): FOUND--> "ESPERANZA"
"G5-S13-Q1.ecoded" (Line #29): FOUND--> "ESPERANZA"
"G5-S13-Q1.ecoded" (Line #59): FOUND--> "ESPERANZA"
"G5-S13-Q1.ecoded" (Line #70): FOUND--> "ESPERANZA"
"G5-S13-Q1.ecoded" (Line #72): FOUND--> "ESPERANZA"
"G5-S13-Q1.ecoded" (Line #74): FOUND--> "ESPERANZA"

Autocoding Unsegmented Free Response Data

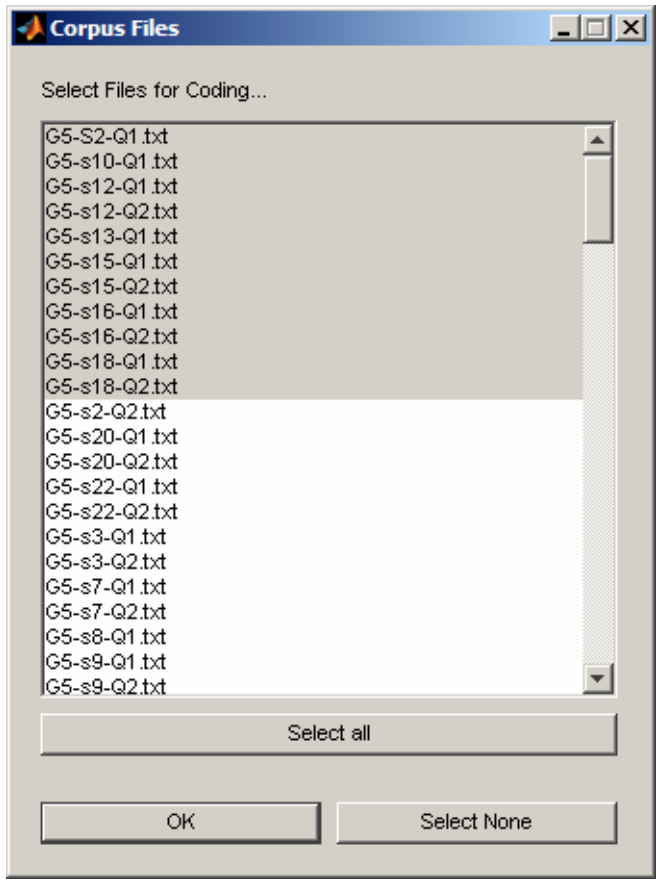
Step 1: Set Autosegment Options Using: `Project→Preferences→Autosegment`



Step 2: Initiate Automatic Autocoding: `Project→Autocoder→Raw Data File`

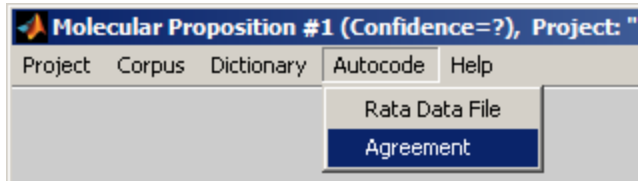


Step 3: Select unsegmented ".txt" files for autocoding. For each `".txt"` file, an "acoded file" (formatted like an "ecoded file") is generated.



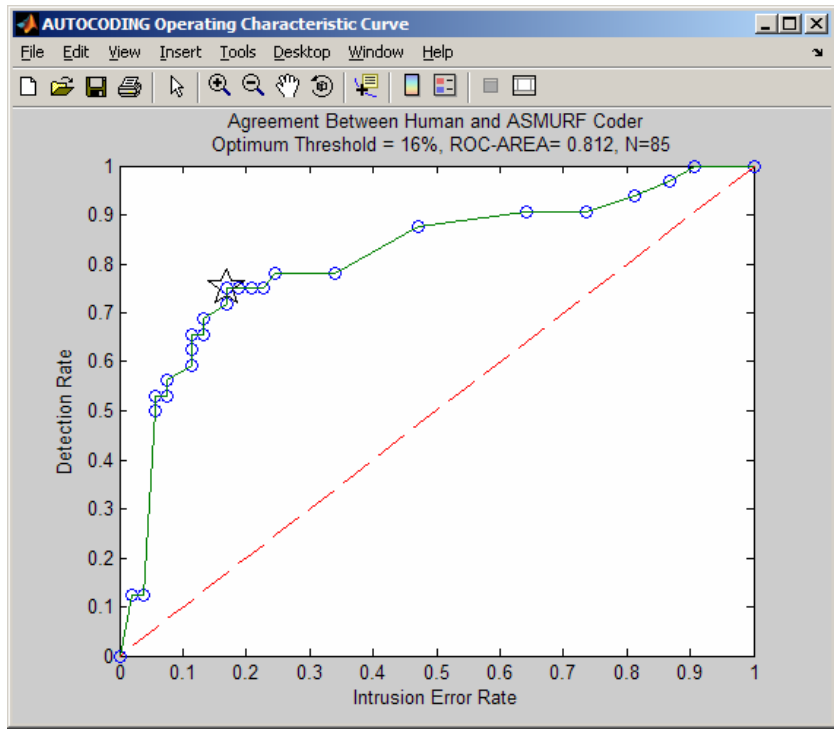
Comparing Coding Performance of ASMURF with Human Coders

Compare the performance of human coders ("ecoded files") with that of the performance of ASMURF ("acoded files") by selecting: `Autocode→Agreement`

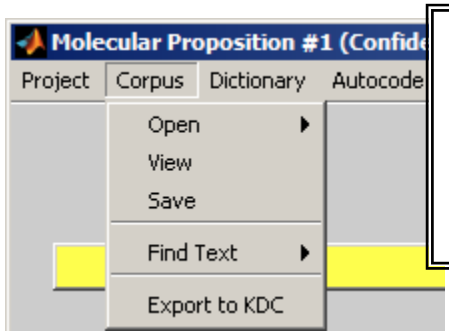


If the ASMURF confidence level that a molecular proposition is present in a particular `acoded` file exceeds threshold θ, then detection and intrusion error rates may be computed for that molecular proposition across pairs of `acoded` and `ecoded` files.

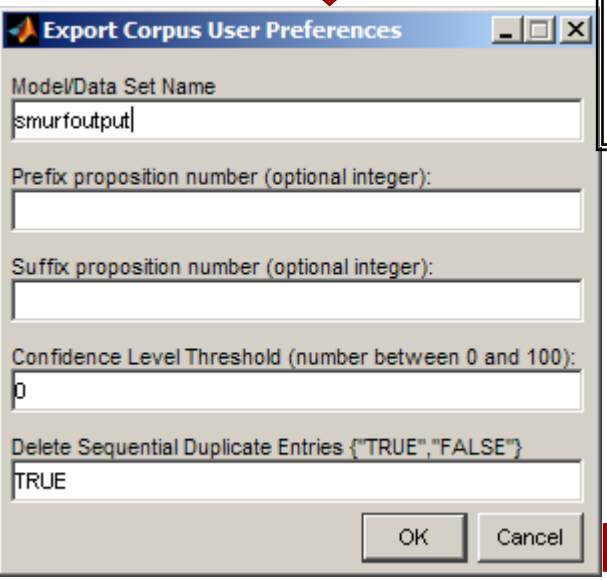
Varying the threshold θ, then generates an operating characteristic curve. The optimal classification threshold which maximizes detection and minimizes intrusion error is also computed as well. ROC (Receiver Operating Characteristic) curve area is also computed. Values of 0.80 and above are generally considered to indicate good classification performance.



Exporting Results for KDC Analysis



To export the coding of the free response data into a form suitable for KDC (Knowledge Digraph Contribution) analysis, use: `Corpus→Export to KDC`



After selecting which "ecoded files" should be exported, the ASMURF software will ask for the name of the "Model/Data Set" ("`smurfoutput`" in this example)

A ***model file*** with the suffix ".model" which maps integers into molecular propositions is generated (e.g.,smurfoutput.model)

A ***data file*** with the suffix ".data" which contains a sequence of integers for each participant which Indicates the sequence of molecular propositions for that participant's response data.