

which was nearly as old as	the	Ashburton	district
	The	Associated	Press quoted Hadithah police Captain
to fly	the	Atlantic	Ocean solo
Like Ashburton,	the	Auckland	central business district last night also
e black and white poster had been named in honor of	the	Aussie	Rules football hero of the same name.
the report, a six-page cable from	the	Australian	Embassy in Copenhagen titled Den
	The	Australian	Embassy notes that the Danish Govern
er ($p > 0.05$) with echo-tracking (0.023 mm) than with	the	B-mode	(0.036 mm) or M-mode (0.074 mm) me
ning spline slightly reduced the standard deviation of	the	B-mode	and M-mode distension amplitudes, but
And in London, a London	the	B-13	reported senior British government sou
was taught	the	B-13	
air strikes blast palaces, government buildings and	the	Baath	Party headquarters across the river from
to supply lines behind	the	Baghdad	front
a friend at	the	Baghdad	neighborhood struck Wednesday by tv
Soldiers were also filmed in	the	Baghdad	parade ground, which is marked by a c
burnt out cars and injured people from	the	Baghdad	bombing, saying the Iraqis
Before	the	Baghdad	market incident, Iraq had reported 78 ci
	the	Baghdad	market incident will be investigated

UAM CorpusTool Version 3.0

Tutorial Introduction (June, 2013)

Mick O'Donnell
michael.odonnell@uam.es

About this Document

This document provides a tutorial introduction to UAM CorpusTool 3.0 (henceforth: UAMCT3). For more detailed information about the options in each screen and menu of UAMCT3, please see the *UAMCT3 User Manual*.

About UAM CorpusTool 3.0

UAM CorpusTool is a set of tools for the linguistic annotation of text. Core concepts include:

- The user defines a project, which is: a set of files, and a set of analyses which are applied to each of these files.
- All the files of a project are stored in a single folder: the original texts (the 'corpus'), the annotations on this text and the coding schemes (the tags applied to the texts).
- Each 'analysis' can be seen as a 'layer' of annotation. CorpusTool currently allows two types of annotation:
 1. **Document Coding**: where the text as a whole is assigned features. For instance, these features could represent the register of the document (field, tenor, mode), or text-type.
 2. **Segment Coding**: The user can select segments within a file, and assign features to each of these segments. Segments are specified by dragging the mouse over a span of text, and the user is then prompted to specify the features of this segment.
- Annotation can be 'manual' (the user swipes text and chooses categories for it) or 'automatic' (the program does the annotation for you). Sometimes annotation is mixed, for instance, you can have the program recognise clause or noun-phase segments, but it is up to the you to code them.:

CorpusTool is available from:

<http://www.wagsoft.com/CorpusTool/>

See that site for instructions on how to install CorpusTool on your machine.

Tutorial 1:

Starting a new project

1 Launch UAM CorpusTool

Once UAM CorpusTool is installed on your machine, you can begin working with it. The first thing to do is to create a new “project”:

Windows:

- When installing CorpusTool, you had the option to place an icon on the desktop. Click on this icon to launch CorpusTool.
- Alternatively, there should be a UAM CorpusTool icon in the Programs menu in the Start menu on Windows Toolbar. Select this to launch CorpusTool.

Macintosh:

- The installation of CorpusTool placed the application in your Applications folder. Double-click on the application to launch it.
- You might find it useful to place the application in the Dock for easy access.

If you have already created a project, you can open it simply by double-clicking the .cp3 file in the Project folder. This file has an icon as below:

MacOSX:



Windows:



The Opening Window

A window should appear as in Figure 1.1. This window provides, amongst other information, the version number you are using (useful if you need to communicate bugs).

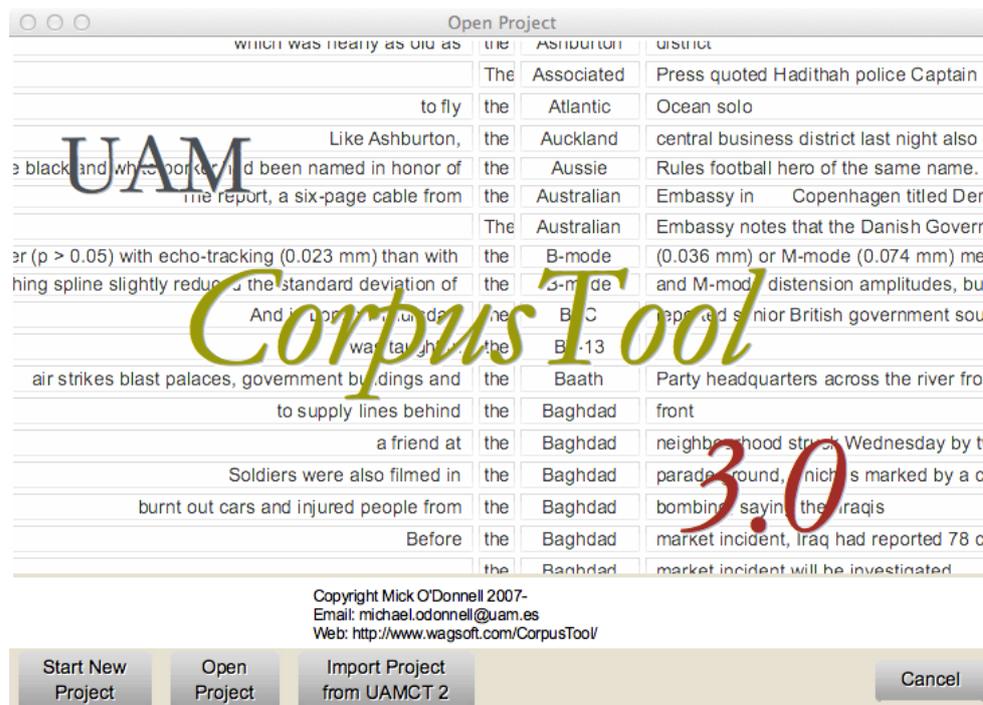


Figure 1.1: The Opening Window

The Window offers several options,

- *Start New Project*: create a new project from scratch.
- *Open Project*, to continue with a project you have already started, you will be prompted to select one.
- *Import Project from UAMCT 2*: If you have a project from UAMCT 2, you can use the “Import Project from UAMCT 2” button to make a copy of your project in the UAMCT 3 format.
- *Open SomeProjectName*: If you have opened a project previously on this machine, there will also be a button to open the last project opened.

2 Click on the “Start New Project” button.

After clicking this button, a “Create Project Wizard” will appear, which will lead you through the steps needed to create your project:

1. Providing a name for a new project
2. Specify the folder where your new project’s folder is to be stored. For instance, choose the Desktop folder on your machine.

When you click the “Finalise” button, CorpusTool will create your project, which is a folder containing all the details related to your project, including the corpus, and the annotation files. It also contains an icon which can be used to launch your project directly (the .ct3 file).

Once you have finished with the *Create Project Wizard*, the *CorpusTool Main Window* will open, showing the *File pane*. See Figure 1.2. This pane is where you add or remove files to your project, or open a file for annotation.

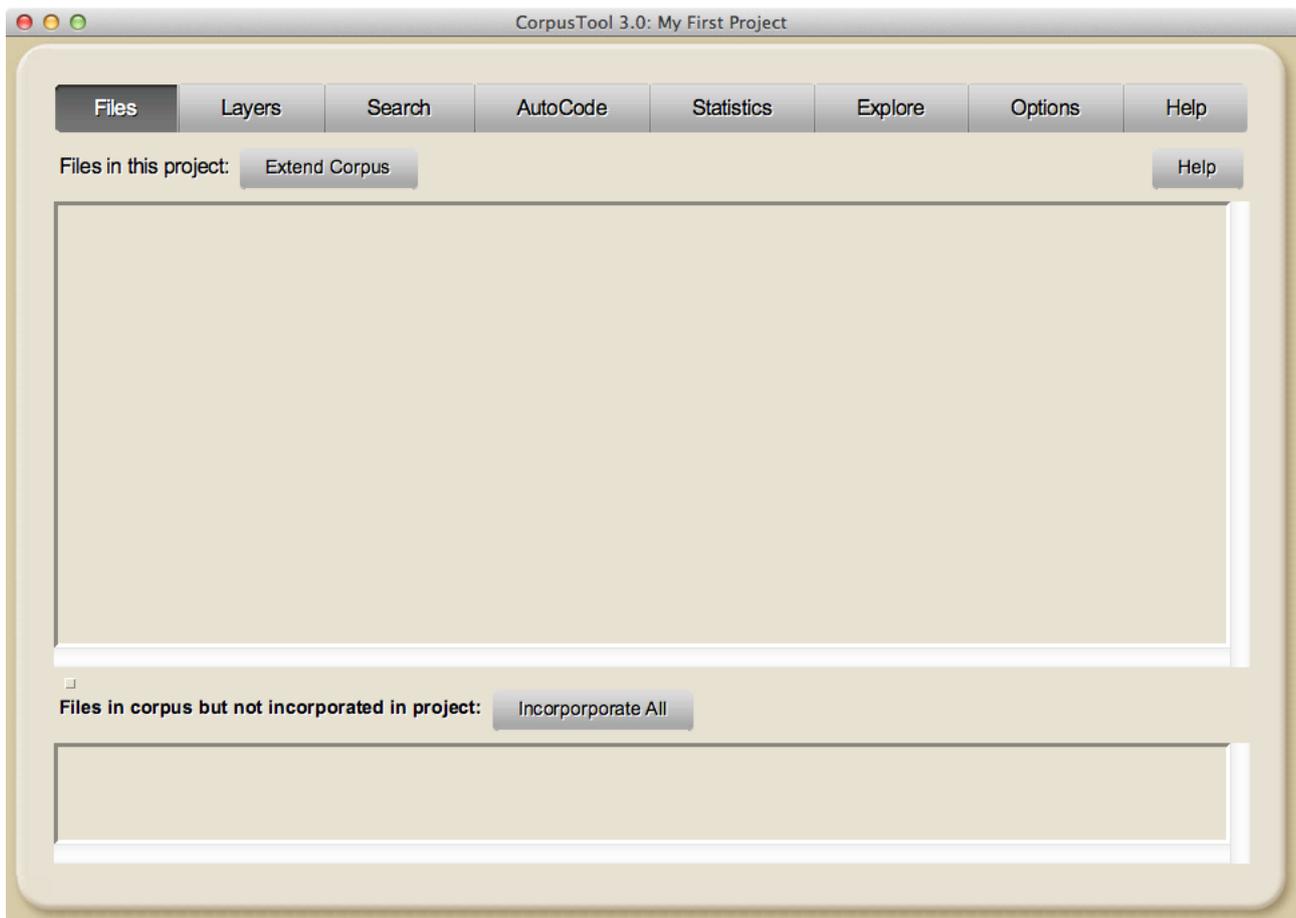


Figure 1.2: The File Management pane

The buttons at the top of the pane allow you to switch between the different panes of CorpusTool: **Files** (Tutorial 2), **Layers** (Tutorial 3) **Search** (Tutorial 5), **Autocode** (Tutorial 6), **Statistics** (Tutorial 7), **Explore** (Tutorial 8), **Options** and **Help**.

We will assume for now that the “File” pane is selected. The name of your project is shown in the title bar of the Project window. In the space below is a box showing all the files in the project (initially empty), and for each file, one button for each of the possible analyses of that file.

This ends the first tutorial. The next tutorial will show how to add content to your project.

Tutorial 2:

Adding text files to your project

The next step is to add some files to the project.

1 Save Documents as plain text

UAMCT 3 deals only with plain text files. If your files are in MS Word format or PDF, you need to save them as plain text.

If you are on Windows, and your texts are in languages with non-western characters (e.g., Cyrillic, Chinese, Korean, etc.), then it is better to open your .docx document with WordPad, and use the “Save as...” option there, as it can save as a Unicode file.

2 Click on “Extend Corpus”

Click on the *Extend Corpus* button in UAMCT. A window will appear to guide you through the process of adding files. You are given a choice between:

- **Add a single file:** You will be asked to select a file to add to the corpus. Additionally, you will be asked to specify a “subcorpus” for the text file. Texts in UAMCT are stored within subcorpora (folders within the Corpus folder). For instance, you might have one subcorpus for native texts, and another for learner texts.
- **Add a folder of files:** You will be asked to select a folder to add to the corpus. This folder could be:
 - A folder of plain text files: the folder will be added as a “subcorpus” of the project.
 - A folder of folders of plain text files: each folder will be added as a “subcorpus” of the project.
- **Paste from the Clipboard:** you will be given a space in which to copy/paste text into. This is a useful way to take texts from the internet into UAMCT.

In the first two cases, the files you select will be copied from where they are into the Corpus folder of your project. The originals are left untouched.

For this tutorial, let's use the “Paste From Clipboard” option. Copy the following paragraphs of text and follow the instructions below:

Obama is like Apple, Google and Facebook: a once hip brand tainted by Prism
Among the guests at the fabled Bilderberg meeting, held this weekend just outside London, are the top brass of Google, Amazon and Microsoft. How appropriate they should be there, alongside luminaries of the US political and military establishment. For this was the week that seemed to confirm all the old bug-eyed conspiracy theories about governments and corporations colluding to enslave the rest of us.

The Guardian revealed that the US National Security Agency has cracked open our online lives, that it can rifle through your emails, listen to your calls on Skype, watching "your ideas form as you type", as a US intelligence officer put it – apparently in cahoots with the corporate titans of the web.

1. Select “I want to paste from the clipboard (Figure 2.1) then press “Next”.

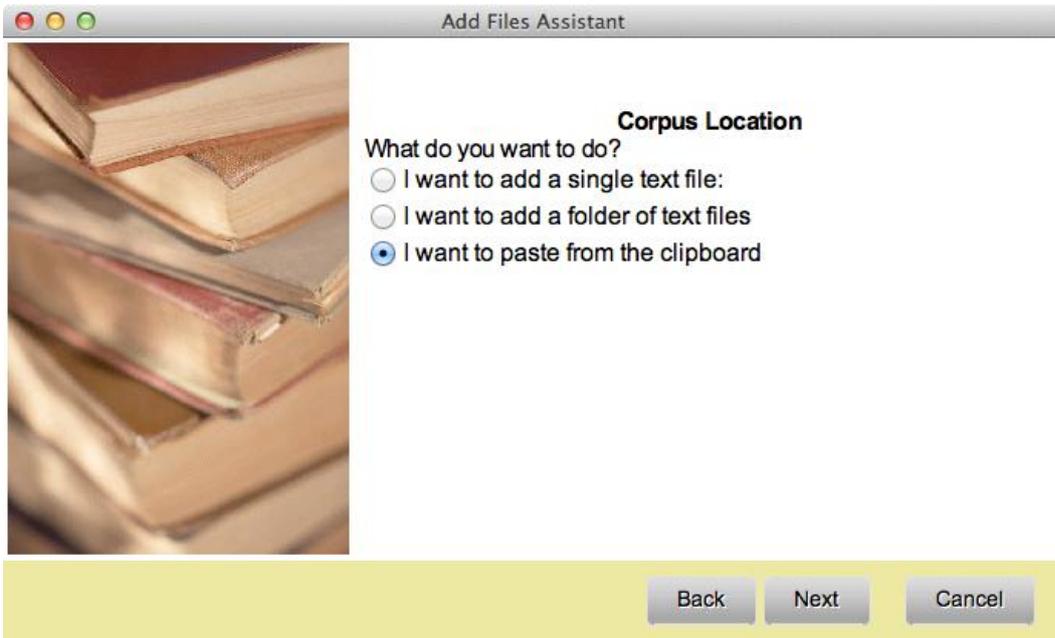


Figure 2.1

2. Paste the text into the space (edit it here if you want).

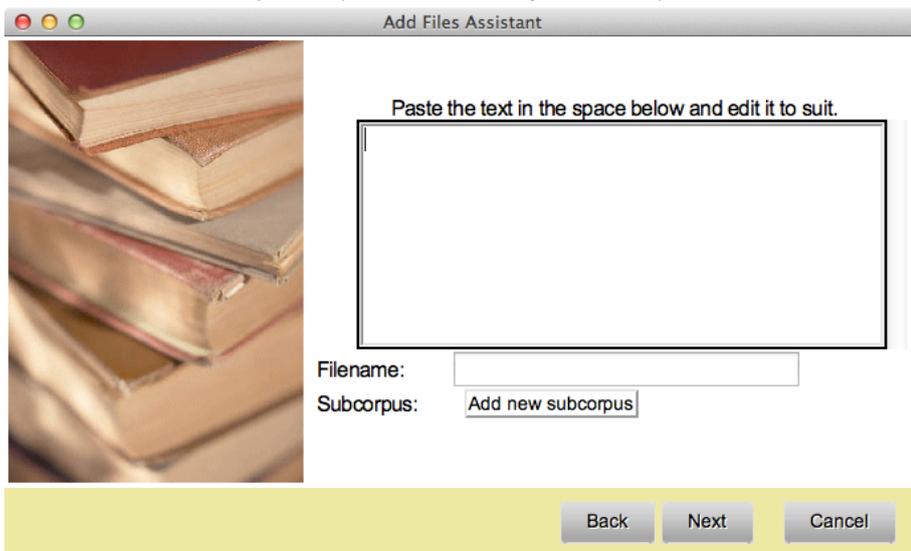


Figure 2.2

3. Type in a filename for the file (e.g., “Obama1.txt”).
4. Leave “Subcorpus” set to “Add new subcorpus”.
5. Press “Next”. You will be prompted for the name of the subcorpus to add the file to. Type “News” and then press OK
6. Press “Finalise”.

The file you added should not be displayed in the Project window.(see Figure 2.3).

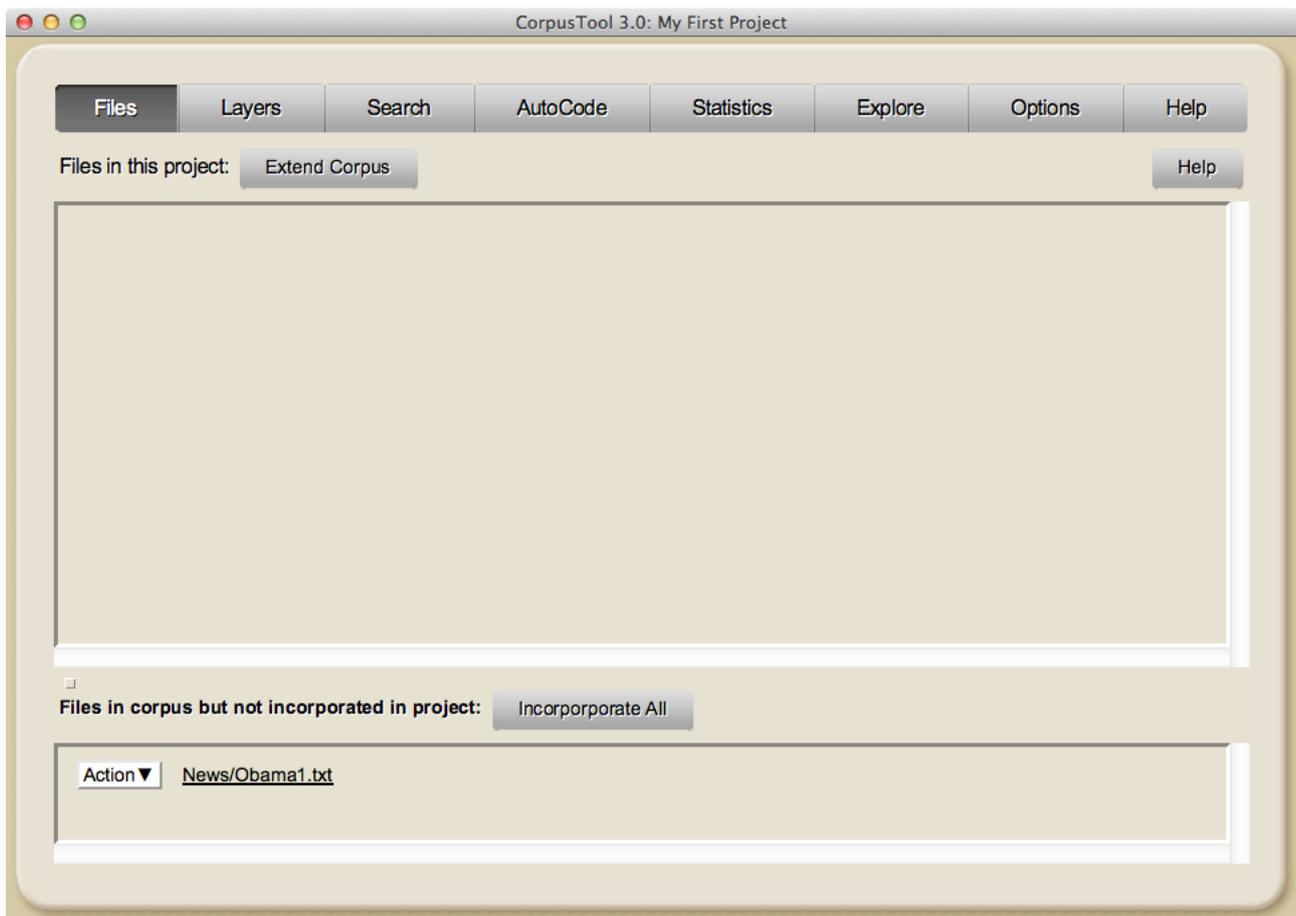


Figure 2.3: Files window after adding a text file

The newly added files are under the caption “Files in corpus but not incorporated in project”. UAMCT makes a distinction between “**incorporated**” files, which have buttons to annotate at all available levels, and “**unincorporated**” files, which are in the corpus but not yet opened for annotation.

This distinction is made to make it easy to keep track of those files which you have started editing, distinct from those you may wish to add later. If you have 100 files in the corpus, but have only annotated five, then you want the five with annotations to be clearly indicated. This allows for a gradual expansion of your corpus over time, but let’s you get results at each point.

3 Incorporating Files

To incorporate a file into the project, making it available for annotation, you can either:

- Click on the “Incorporate All” button to incorporate all unincorporated files, or
- Click on the “Action” button next to a file and select “Incorporate file” from the menu. This will incorporate just the single file.

If you do either of the above, you will be presented with a window asking for some metadata regarding the file or files (See Figure 2.4). This includes:

- **Language:** which language the text written in? This field is used to determine which language resources to use for the document. These resources include lexicons (for concordance searching, calculation of

lexical density, etc.), parsers (for automatic segmentation) and taggers. Currently, only English is really supported, but soon lexical resources for other languages will be provided.

- **Encoding:** text files are stored in a particular text encoding. You can tell CorpusTool what encoding your file is in by selecting from this field. The default option offered by UAMCT is a guess of what it should be, but if the text does not display properly, you may need to change it. To find out what encoding the document is in, try right clicking on the document and select “Open with...” (or the MacOSX equivalent) and open the text with MS Word, which may help you choose the best encoding. Otherwise, using ‘Open with...’, select a browser, and look for the “Encoding” or “Character Encoding” menu item, and see which encoding this program gave the text.
- **Display Font:** Choose here the font family and size you want to use to display your text in the annotation windows. Some fonts will better cope with non-western writing systems, e.g., some fonts are designed to display Chinese, etc. However, many modern fonts should display any writing system.

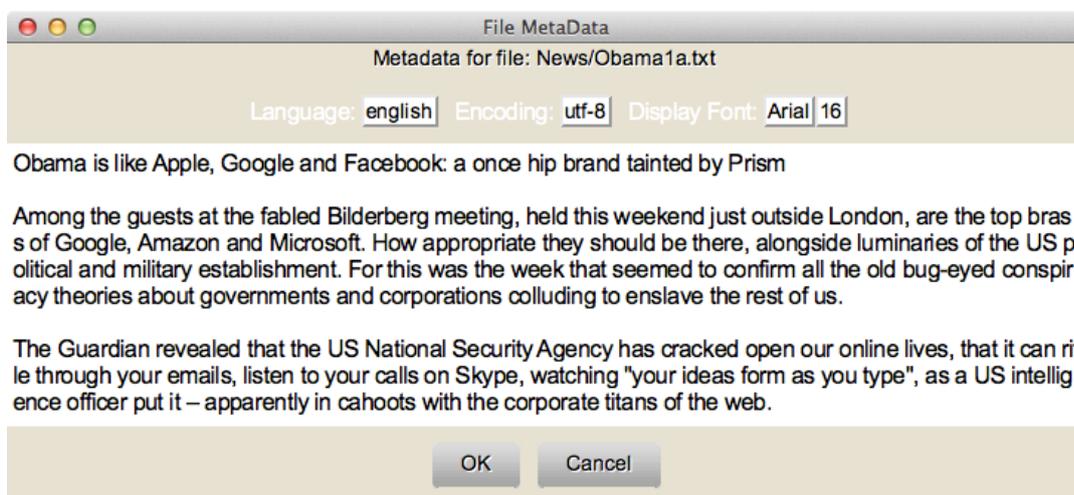


Figure 2.4: File Metadata Window

After incorporating the file, the Project Window appears as in Figure 2.5.

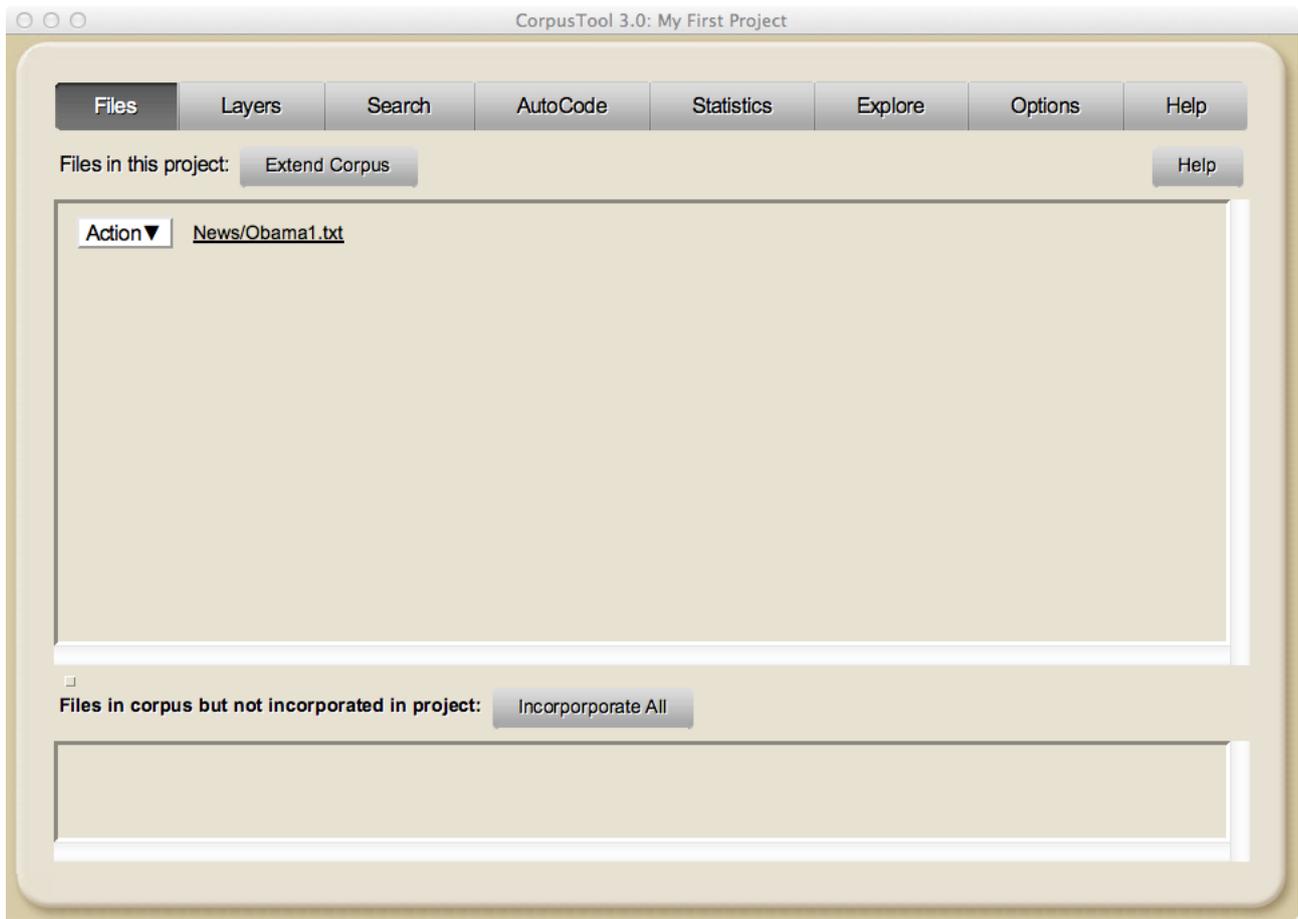


Figure 2.5: The Files Window after incorporating files.

Tutorial 3:

Adding a manual annotation layer to your project

The next thing we need to do is to specify what analyses you want in the project. Let's start by adding just one layer. A "Layer" is a type of analysis of the text files. We can add layers for coding clauses, for coding groups, for the register of the whole text, for appraisal analysis, etc. For this example, we will assume we are adding a layer for analysing noun phrases (NPs) in terms of both their content (what they express), and their form (proper, common, pronominal).

1 Change to the Layers pane

Click on the "Layers" button at the top of the window.

2 Click on the "Add Layer" button.

When you click on "Add Layer", a window will pop up asking several questions. Use the Next button to move between questions:

1. Layer Name: the name given to the layer. Put "Entity".
2. Automatic or Manual Annotation: choose 'Manual'.
3. Scheme: choose "Design Your own". The other options allow you to use one of the schemes supplied with UAMCT3, or to use a scheme from another project you have created.
4. Kind of Segment: here you specify whether you want to assign features to a text as a whole (e.g., its register or text type) (*Whole Document*), or whether you want to assign features to subsegments in the text (e.g., clauses). Let's assume that we are interested in the second, so click on "*Segments within a Document*".
5. Special Layer: This window offers options for special kinds of annotation. Error annotation layers provide a special slot on the coding interface for you to provide the correction of the error. RST annotation provides a special interface for annotating the "rhetorical structure" of the text. For now, just select "No".
6. Automatic Segmentation: here you can specify whether the text should be segmented for you, recognising paragraphs, sentences, or words. For English texts, automatic recognition of clauses and NPs is also possible. For this tutorial, select "No".
7. After following these steps, you will see a final window displaying your choices, as in Figure 3.1. If any of the settings vary from yours, use the Back button to go back and change it. Then press "Create Layer" to return to the main window.

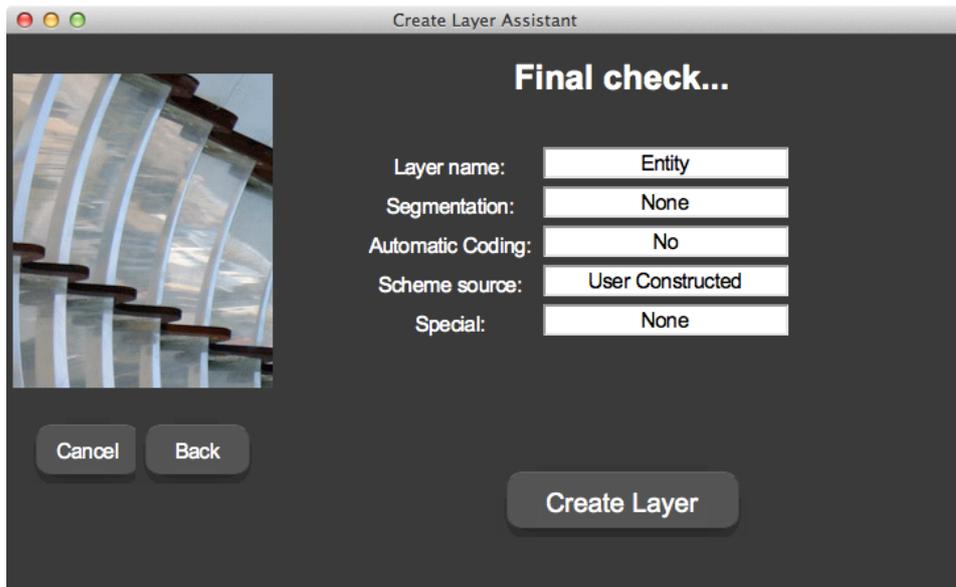


Figure 3.1: Last pane of the Add Layer Assistant

Figure 3.2 shows the Layers window with one layer added. The Layer space provides some information about the layer.

There are two buttons on the Layer control panel:

- **Edit Scheme:** this button will open a window to allow you to edit the coding scheme. We will come back to this in the next tutorial.
- **Delete Layer:** this will delete the layer, and all analyses of text files performed on this layer. Press this only before you begin coding of the layer, or if you really want to delete the layer.
- **Edit Details:** this button is currently disabled, but will in the future allow you to change the characteristics of the layer (e.g., manual/automatic, auto-segment, etc.). Currently, you need to delete the layer and add it again to change the characteristics.

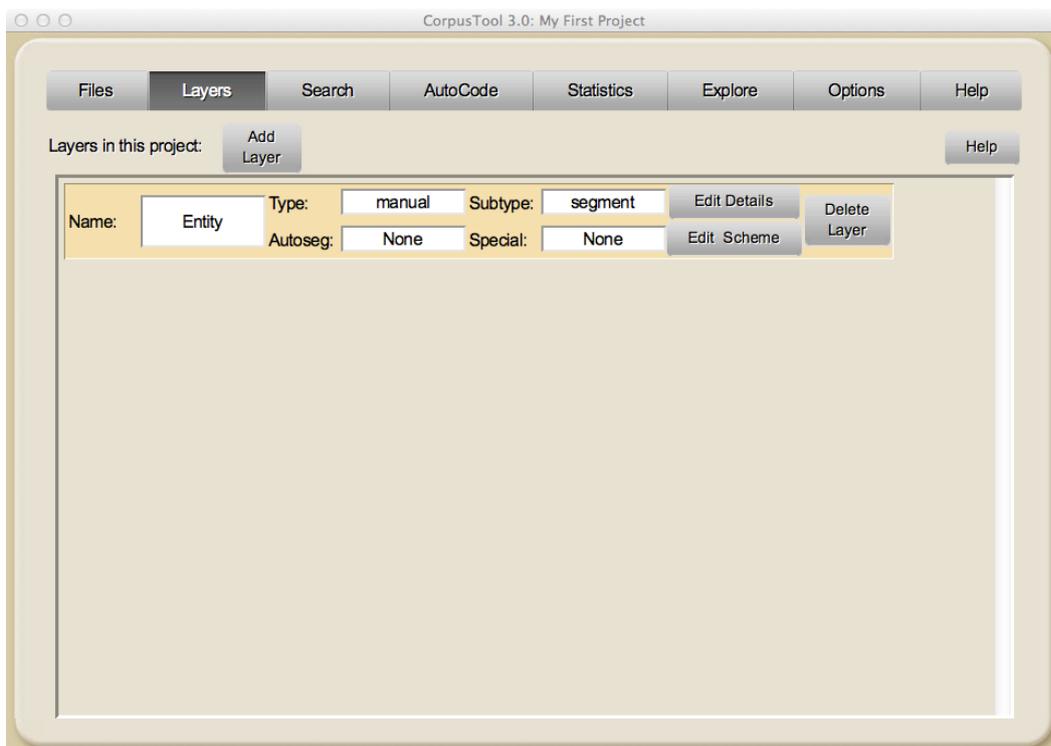


Figure 3.2: The Layers Window with one Layer added

2.1 Return to the Files pane

If you click on the Files button, you will see that the display has changed slightly. The entry for the “Obama1.txt” file now has a button next to it “Entity”. You can click on this button to edit this file at this layer. The colour of the annotation buttons are colour coded to indicate their degree of completeness:

- Light: Not yet coded
- Medium: Partially Coded
- Dark: Coded to a high degree

Don't open the annotation window just yet, though. First, we need to specify the coding scheme for the Entity layer. The next tutorial will deal with this process.

Tutorial 4:

Editing the Coding Scheme

1 Opening the Scheme Editor

Before annotating files for a given layer, you need to define the annotation scheme for the layer. The first step here is to open the scheme editor. Change to the Layers pane, and click on the “Edit Scheme” button for the layer. This tutorial assumes we are working on the “Entity” layer defined in the previous tutorial.

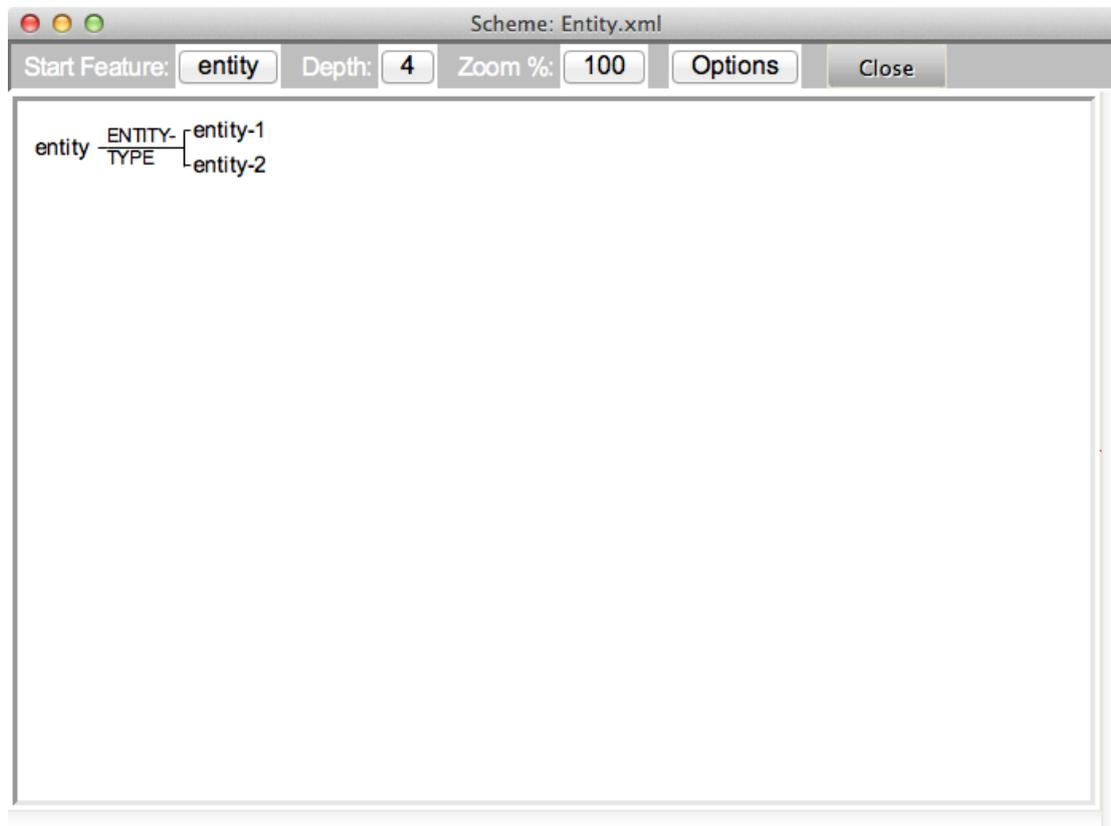


Figure 4.1: The Entity Scheme before editing

A window like Figure 3.2 will pop up. It shows a small “system network” (a hierarchy of features), with “entity” as the most basic concept, and a choice between entity-1 and entity-2.

2 Editing the Scheme

These features have been automatically generated, and we will change them to more informative names.

- **Click on “entity-1”**, and a menu will appear with options, as in Figure 4.2.

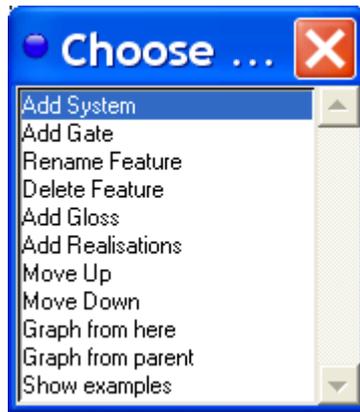


Figure 4.2: Options for Features

For more information on these options, please see the manual. For now, we want to change “entity-1” to something more plausible. Let’s assume that we want to code our entities as either ‘human’ or ‘organisation’ (we will ignore NPs that refer to other sorts of entities).

- **Select “Rename Feature”**. A window will appear allowing you to edit the feature name.
- **Change the feature text to “human”** (in UAMCT, all features are in lower case, and spaces are not allowed, so if you put capitals, they will be changed to lower case, and spaces will be substituted for “-“).
- **Repeat this process to change “entity-2” to “organisation”**.

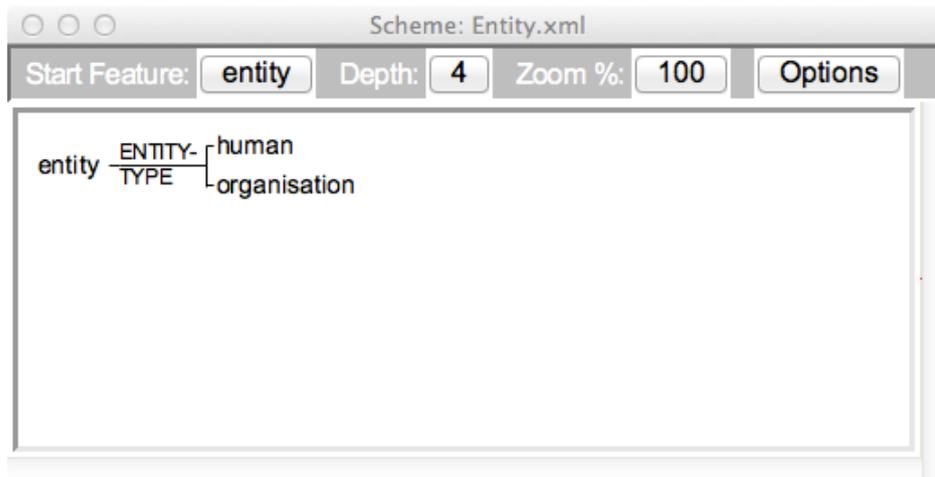


Figure 4.3: Start of the Entity Scheme

After this, you should have the scheme as shown in Figure 4.3. Notice also that the choice between human and organisation has a name, automatically provided as “ENTITY-TYPE”. If you want to rename this choice, you could click on “ENTITY-TYPE”, and select “Rename System” from the menu which appears. Rename the system to “SEMANTIC-TYPE”. You can type in lowercase, but UAMCT will always display system names in upper case.

Lets now add a more delicate distinction between types of organisations. We want to sub-classify organisations as either company, government or media.

- **Click on ‘organisation’ and select ‘Add System’**: a new system (set of choices) will appear under ‘organisation’, as shown in Figure 4.4.

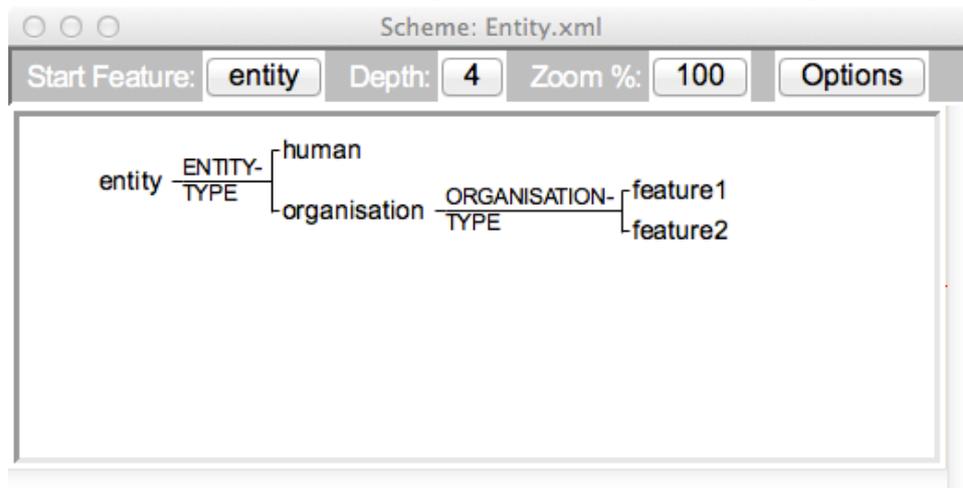


Figure 4.4: The Entity scheme with subsystem

- **Rename ‘feature1’ to ‘company’**
- **Rename ‘feature2’ to ‘gouvernement-body’**
- **Click on ‘ORGANISATION-TYPE’ and select ‘Add Feature’**, calling this feature ‘media’

The resulting scheme should look like the scheme in Figure 4.5.

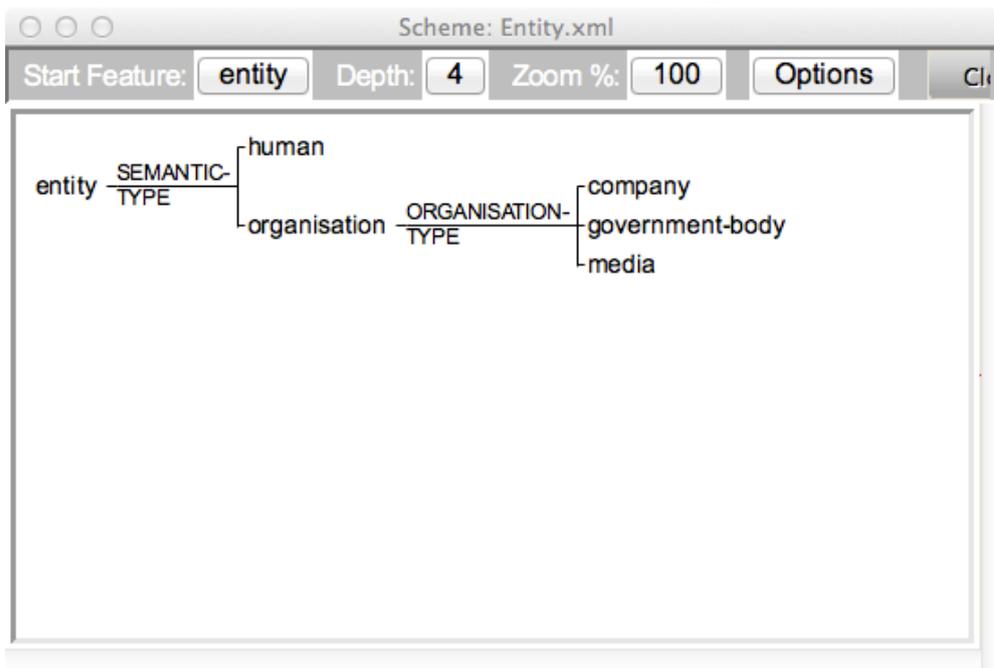


Figure 4.5: Scheme with subsystem renamed

As a next step, we want to code each entity not only by semantic type, but also in terms of form. We can do this by providing a sub-network in parallel to the content network.

- **Click on ‘entity’ and select ‘Add System’**: this will add a system underneath the original one. The curly bracket indicates that, during coding, you have to select from both the system above and the system below. See Figure 4.6.

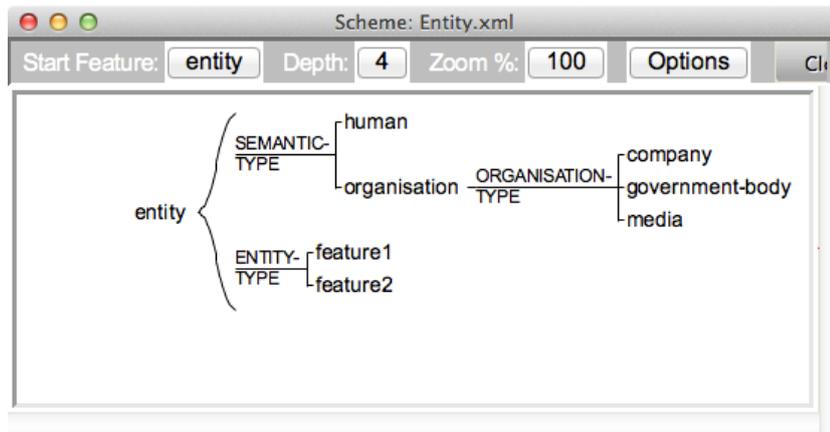


Figure 4.6: Network with parallel system

We now need to edit this new system:

- Click on “ENTITY-TYPE” and select “Rename System”: call the system “FORM”.
- Click on ‘feature1’ and rename it ‘common’
- Click on ‘feature2’ and rename it ‘proper’
- Click on ‘FORM’ and choose ‘Add Feature’, and call the feature ‘pronoun’.

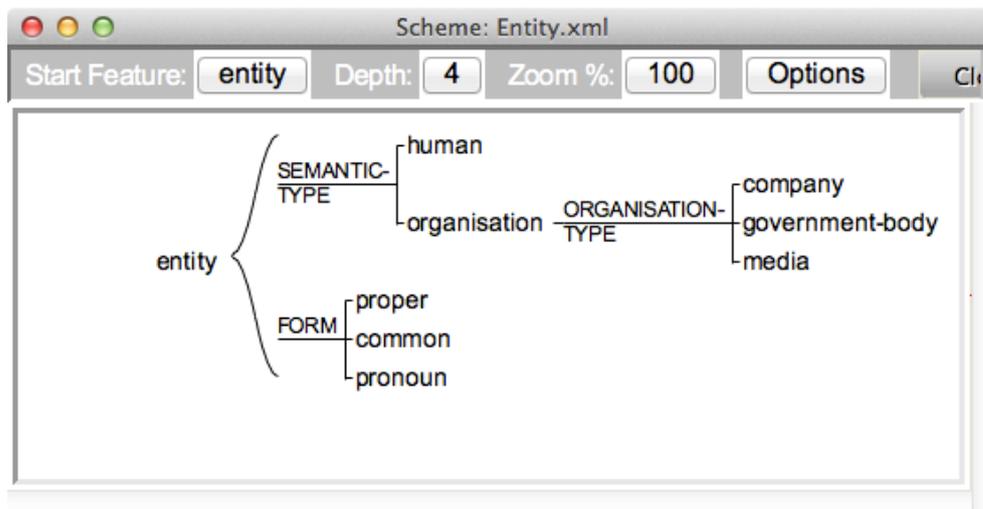


Figure 4.7: The finished scheme

The resulting network should look like Figure 4.7. Coding Schemes can get quite complex. They can grow to contain hundreds of choices, with lots of parallel systems and sub-classifications. However, the smaller the scheme, the quicker the coding.

This is all we need for now. For more details on editing coding schemes, and including the schemes in your publications, see the UAMCT3 Manual.

Tutorial 5:

Manual Annotation

THE REST OF THIS DOCUMENT STILL NEEDS
TO BE UPDATED FROM VERSION 2.8

1 Annotation Types

CorpusTool currently supports two types of annotation:

1. *Code-document*: the document as a whole is assigned features. Useful for defining document language, text-type, register, etc. Also can be used to code features of the writer (e.g., language proficiency).
2. *Code-segments*: the user defines segments in the document, and assigns features to each segment. For instance, clauses, NPs, words, speaker turns, etc.

Below we will explain how to annotate in both manners.

2 Annotating Code-Document files

Each text file incorporated into your project has a button for each layer of analysis. If you click on a layer button where the layer was specified as “Code document”, then a window like that in Figure 4.1 will appear.

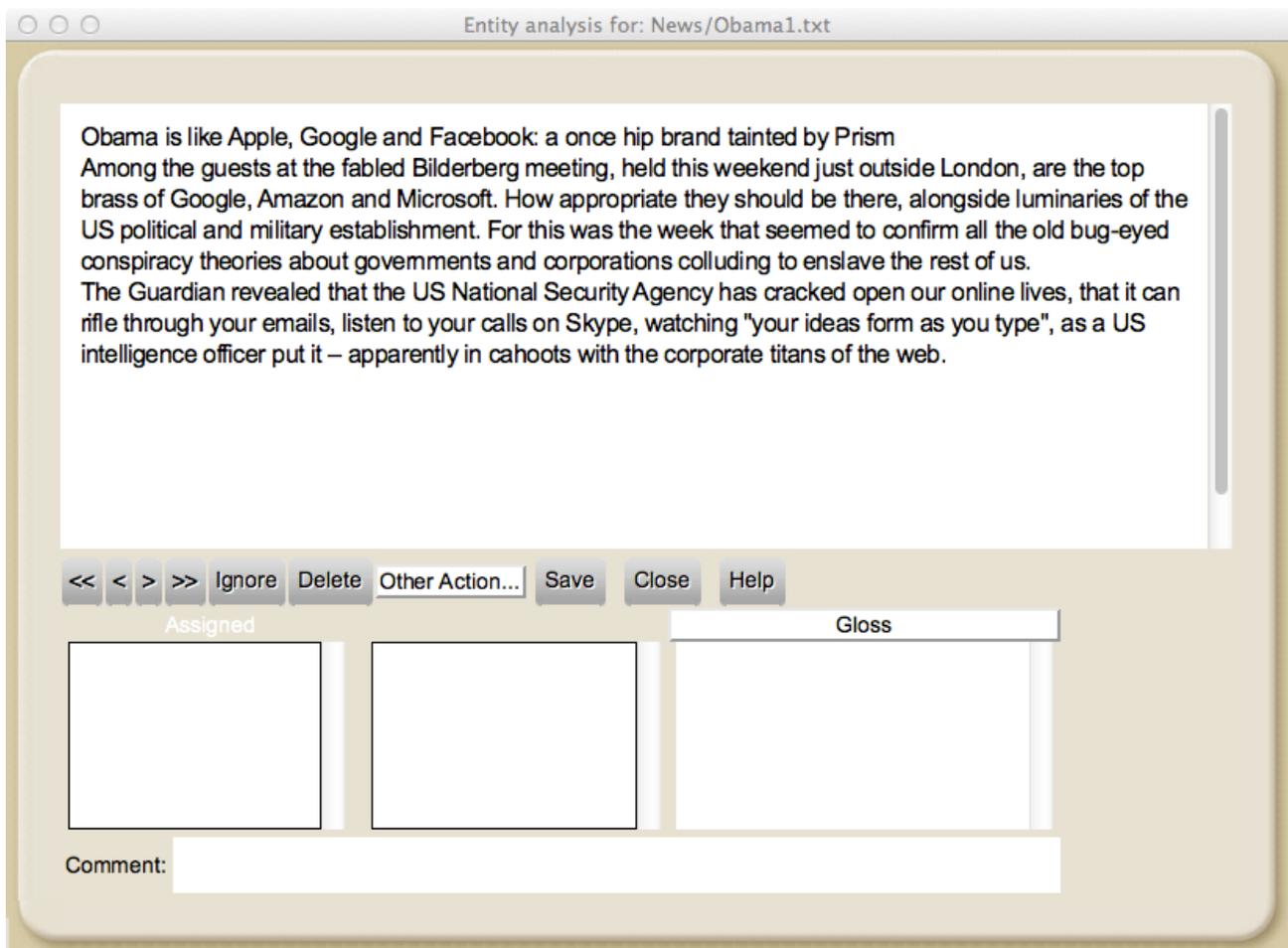


Figure 5.1: An Annotation window

The code-document window has 4 parts:

1. The **Text Frame** shows the text file. You can scroll to see the whole text.
2. The **ToolBar**: giving various actions, such as Save, Close and Help (see below).
1. The **Coding Frame** contains three boxes:
 - a. *Selected Features* (labelled 'Assigned'): the features already assigned to the text. Initially, this will contain one feature, the leftmost ('root') feature of the coding scheme for this layer. As other features are assigned, they will appear here. You can delete features by double-clicking on the features in the Selected Features box. The root feature cannot be deleted, since it applies by default to all documents.
 - b. *Current Choice*: the middle box is a choice which needs to be made for this document. Double-click on one of the options. That choice will be moved to the Selected Features box. If there are more choices in the coding scheme, the next choice will then be displayed.
 - c. *Gloss Box*: If you introduced a gloss for a feature in the scheme (see Section 3.3 above) then, if you (single-)click on a feature in the Current Choice box, the gloss will be displayed in this space. This is useful when you have forgotten what exactly is the coding criteria for this feature.

2. The **Comment Frame**: In this box, you can type comments about the current segment, either to remind yourself of some problem, or to communicate with other people working with the same project. For instance, one might write: “Is this a material or behavioural clause? Check with IFG.”

In summary, to code a whole document:

1. Select from the options shown in the Current Choice box until no options remain.
2. If you make a mistake, double click on features in the Selected Features box to undo the selection.
3. Close the window and your codings will be saved.

3 Annotating Code-Segment files

When annotating a document at a layer specified as “Code Segments”, the process is slightly more complex.

Firstly, for the sake of this tutorial, let’s add a new layer to our study.

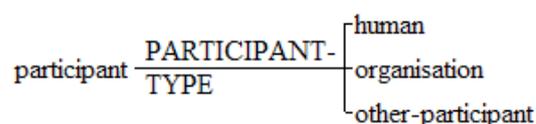
1. Bring the Project Window to the front
2. Click on the Add Layer button on the right of the screen
3. Call the layer “Participant”
4. Select “Annotate Segments”
5. Select “Do not automatically segment”
6. Select “Create New Scheme”
7. Press the “Finalise” button

Note that this adds a new Layer in the layer space, and also adds a new button for each incorporated file.

Now, let’s define the scheme for this layer:

1. Click on the Edit button in the space for the Participant Layer.
2. When the scheme window opens, change *participant-1* to *human* and *participant-2* to *organisation*.
3. Click on “PARTICIPANT-TYPE” and select the option “add feature”. Type in “other-participant”.

Your network should look like that shown below:



Now, close this window, returning to the Project Window.

Click on the “Participant” button for one of your text files.

This will open an annotation window for the document at this layer. See Figure 4.2.

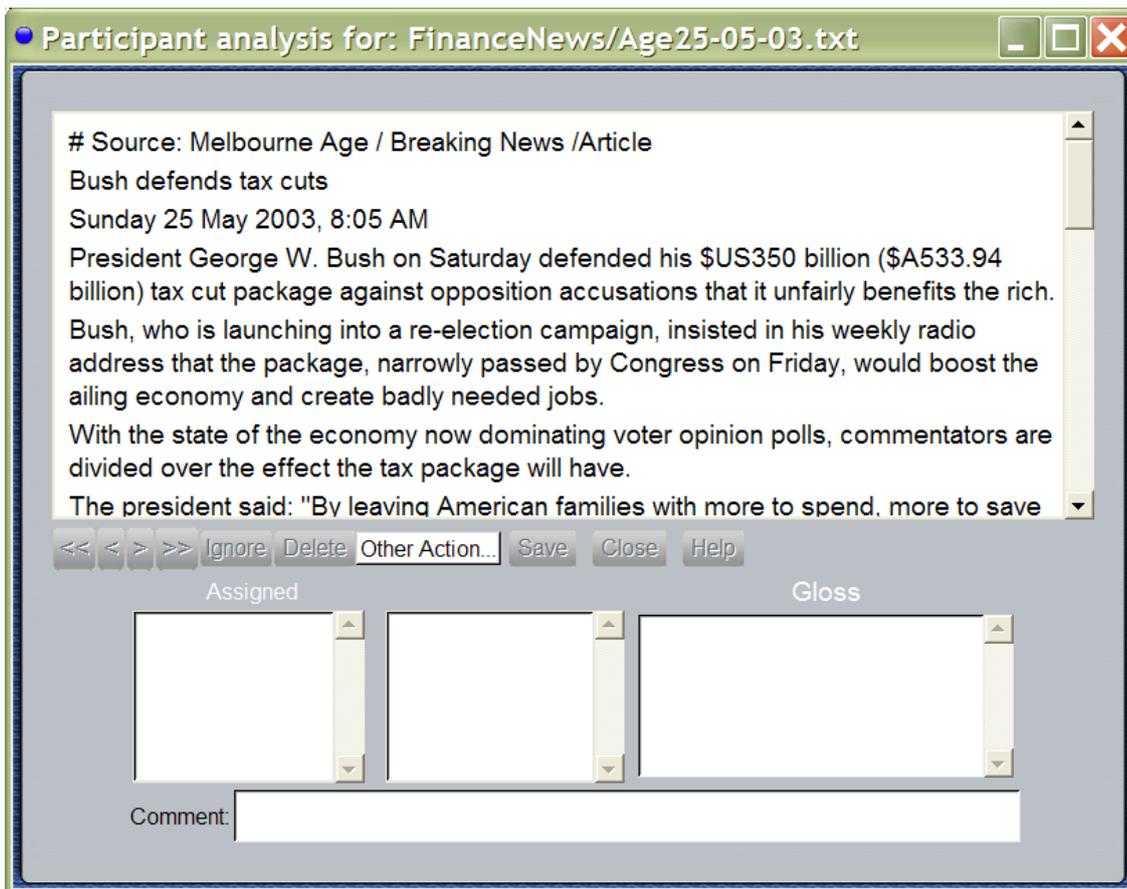


Figure 4.2: Code-segments window

This display differs from that for coding a whole document in that there are more buttons in the toolbar in the middle. These buttons basically allow you to move through the segments.

3.1 Making, Moving and Selecting Segments

- **Make segments** by 'swiping' text: clicking down at one point in the text and dragging to the place you want to end the segment, then releasing the mouse.
- **Select segment:** you can select a segment by clicking on the segment line which runs under each segment. You can tell which segment the mouse is over, as the line of the segment is highlighted.
- **Select next/previous segment:** use the < and > buttons in the toolbar to move around between segments.
- **Select next/previous incomplete segment:** use the << and >> buttons in the toolbar to move to the next or previous segment which is not totally coded yet.
- **Resizing Segments:** Select the border of a segment by moving the cursor over the small border marker (a vertical line) until it goes red to indicate you are over it. Then click down and drag it where you want to go.
- **Delete segments:** if you create a segment erroneously, you can delete it by selecting the segment then clicking on the delete button in the toolbar. Alternatively, hit the Delete key.

3.2 Ignoring Segments

Click the Ignore button when a segment is selected, and this segment will not be used in statistical analyses. Ignore segments are shown in grey in the text window. The same button can be used to unignore a segment.

4 The “Other Actions” Menu

This menu displays some extra options, depending on the kind of annotation (whole-document, segments) that you are annotating:

- **Edit Scheme:** Opens the scheme window for this annotation layer, so that you can edit the scheme, or add/change the glosses associated with features.
- **Add New Feature:** Prompts you to type in the name of a new feature, which is added to the currently displayed set of choices, and assigned to this segment.
- **Copy Features:** Copy the features so far assigned to this segment into memory.
- **Paste Features:** Assigns the features previously copied to this segment.
- **Resegment Document:** Wipes all segmentation of this layer for this document. Note: this deletes all annotation of the document at this layer.
- **Show XML:** Displays how the currently open file is stored on disk, in XML format.
- **Show Structure:** Switches to an alternative display of the segmentation interface, which approximates more to the standard structural display of Functional Linguistics. See figure 4.3.

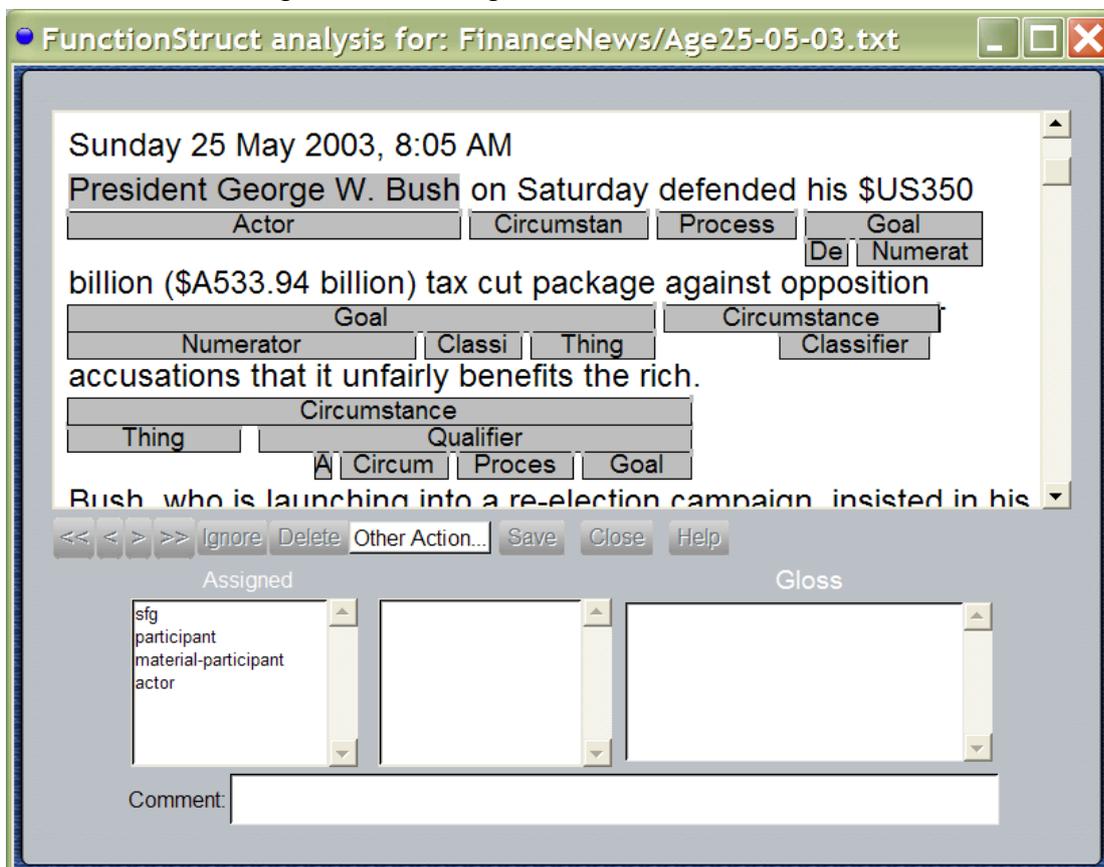


Figure 4.3: Function Structure Display Mode

- **Show Text Stream:** brings up a new window which allows you to view how choices made change throughout this text. Use the “System to Graph” menu to select a system to view. Use the “Smoothing” menu to change the degree of smoothing. With 0 smoothing, each choice is shown in the sequence it occurs in. Use higher levels of smoothing to better view how choices are distributed over phases of the text. For instance, in Figure 4.4, the text stream shows that passive clauses occur more strongly at the beginning of the text, and to lesser degrees later in the text.

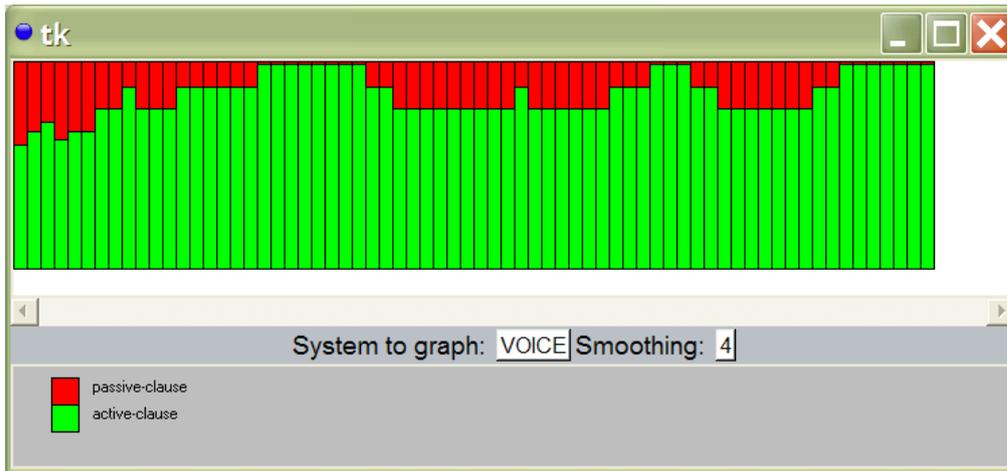


Figure 4.4: Text Stream Window

Tutorial 4:

Adding a “document” layer to your project

The first thing to do in a new project is to specify what analyses you want in the project. Let's start by adding just one layer. For

5 Click on the “Add Layer” button.

A “Layer” is a type of analysis of the text files. We can add layers for coding clauses, for coding groups, for the register of the whole text, for appraisal analysis, etc.

Let's start by adding a Layer for the Register (features which belong to the document as a whole).

When you click on “Add Layer”, a window will pop up asking several questions, and use the Next button to move between questions:

- Layer Name: the name given to the layer. Put “Register”.
- Automatic or Manual Annotation: choose ‘Manual’.
- Coding Object: here you specify whether you want to assign features to a text as a whole (e.g., its register or text type) (*Annotate Document*), or whether you want to assign features to subsegments in the text (e.g., clauses). Let's assume that we are interested in the first, and select on “*Annotate Document*”.
- Coding Scheme: the coding scheme is a description of the features you want to annotate the text with. You have two options here:
 - i. *Create New Scheme*: In most cases, the user is interested in making their own coding scheme, representing the features that they themselves are interested in, organised in the way they feel they should be. CorpusTool includes an easy to use interface for creating and modifying these schemes (see section 3).
 - ii. *Copy Existing Scheme*: In some cases, you might reuse a coding scheme that you developed before, or which was produced by someone else. CorpusTool ships with a few schemes predefined, which you could use. One of these is Peter White's Appraisal network. Another is based on Granger's error annotation scheme.

8. For this tutorial, select “Create new scheme”. Then click on the Finalise button, and your new layer will be added to the Project Window.

Figure 2.3 shows the Project window with one layer added. The Layer space provides some information about the layer: it's name (Register), its type ('code-document'), and the name of the scheme associated with the layer ('Register.xml').

There are two buttons on the Layer control panel:

- **Delete:** this will delete the layer, and all analyses of text files performed on this layer. Press this only before you begin coding of the layer, or if you really want to delete the layer.
- **Edit:** this button will open a window to allow you to edit the coding scheme. We will come back to this in the next section.

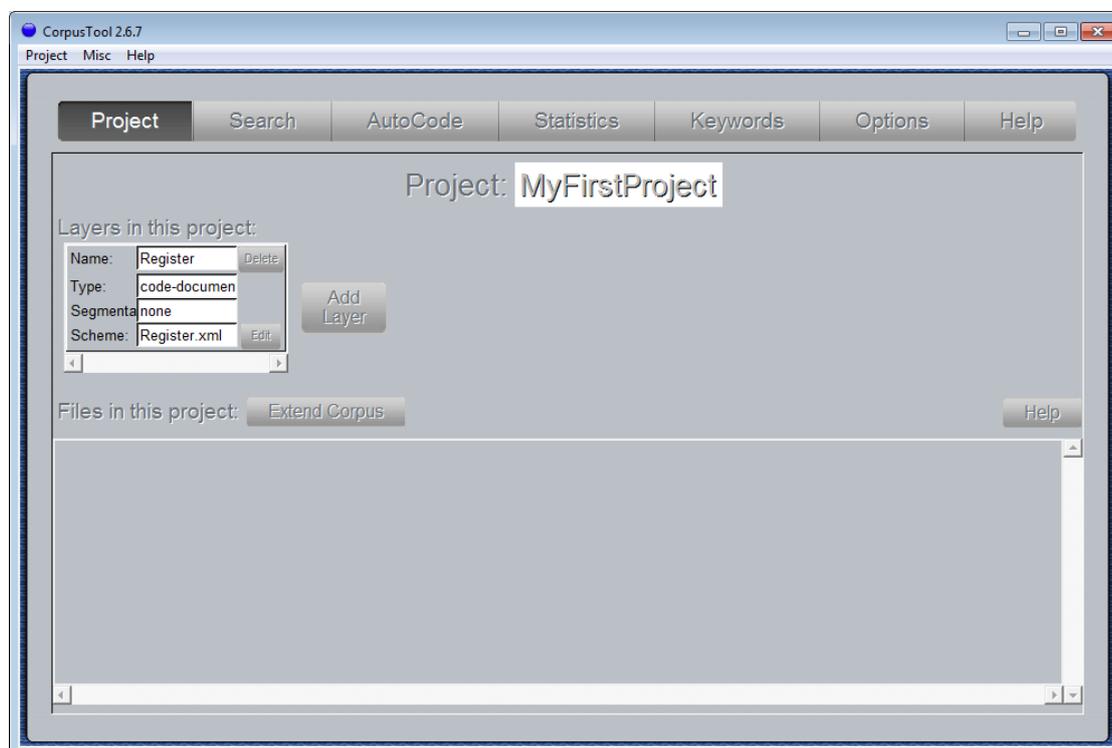


Figure 2.3: The Project Window with one Layer added

9.

You can also select 'Import layer' from the Project menu to add a layer using a *Systemic Coder* study (.cd3 files). See Appendix I for more details.

5.1 Opening an Annotation Window

The remaining buttons on each row each correspond to an annotation layer defined in your project. Click on the button to open an annotation window for this file at the specified layer.

Button Colours: The buttons for each layer of a document are colour coded to indicate their degree of completeness:

- White: totally coded
- Light Blue: Partially Coded
- Dark Blue: Coded to a high degree

Note that these colours are indicative only.

6 Quitting CorpusTool

Note that all changes to a project are automatically saved. If you quit the Project Management Window (using the X in the top right corner), you quit CorpusTool, all changes saved.

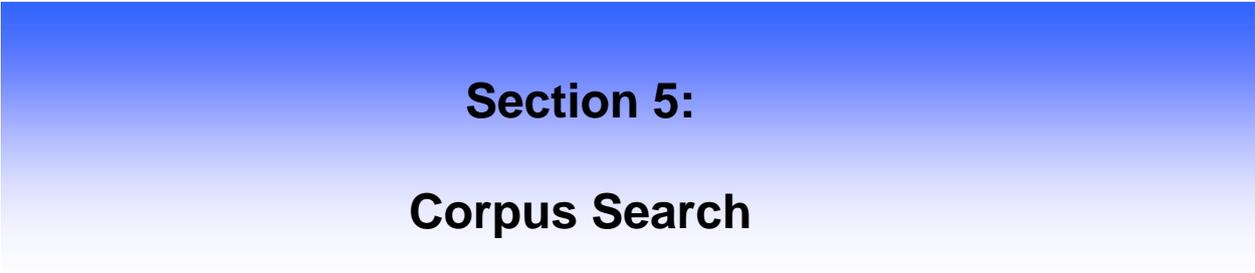
7 Continuing a Project

Once your project is created, the easiest way to open CorpusTool to work on your project is:

1. Open your project folder on the desktop
2. double-click on the .cptr file (which has a blue globe icon).

CorpusTool will open directly with your Project Window.

UNDO: No undo is currently supported. It will be supported in a later version.



Section 5: Corpus Search

1 Introduction

The Search Interface is opened by clicking on the *Corpus Search* button on the Project Window. Figure 5.1 shows this window.

NOTE: You can also open the Search Window from:

- a Scheme window. Click on a feature and select “Show Examples”. CorpusTool will open the Search window with all segments marked with that feature displayed.
- Descriptive or Comparative Feature Statistics: Click on the count field of any set and the instances which make up the count will be displayed.

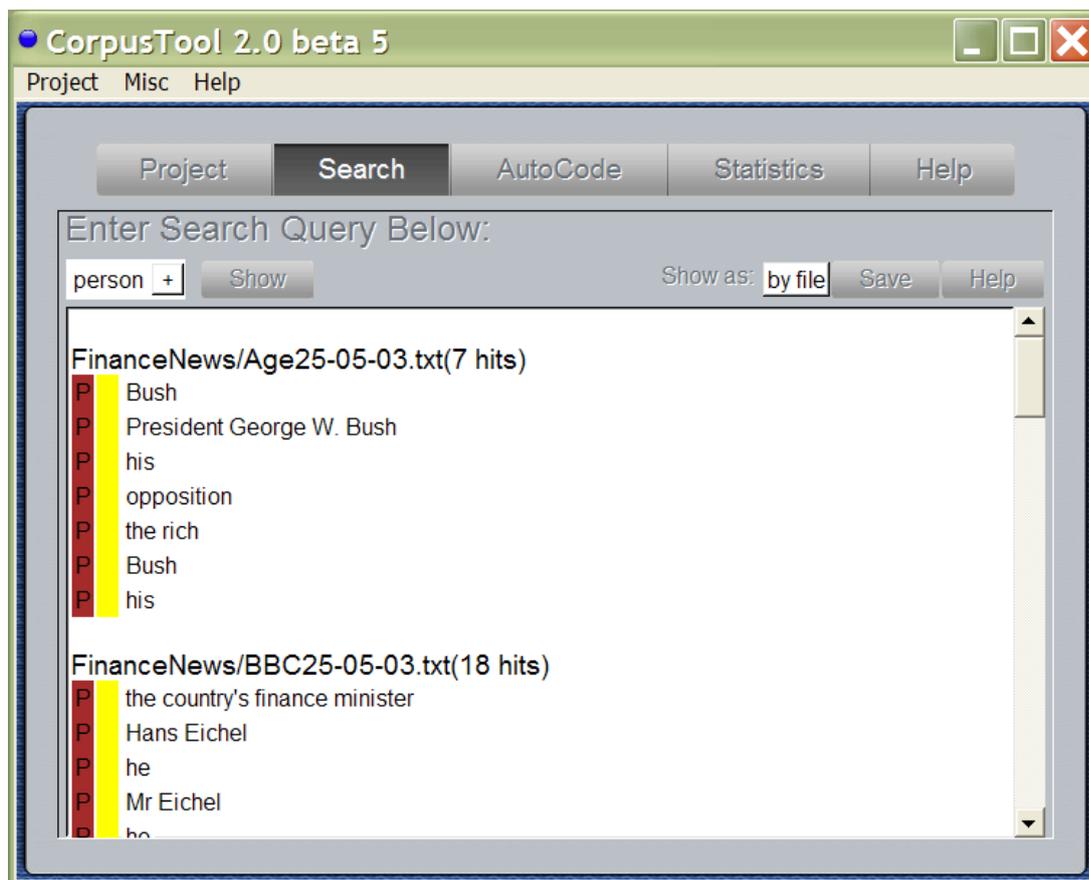


Figure 5.1: The Corpus Search Window

2 Specifying Search Queries

At the top of the window is a menu-driven widget to define your search query.

For this tutorial, we will use a small project called “Finance”, which can be downloaded from the CorpusTool website, on the Download page.

1. **Simple Feature Search:** To search for all segments containing a given feature, click on the widget at the top left (in Figure 5.1, “person”), and select a feature from one of your layers. Then press “Show” to see all instances. Press “Save” to save the search results to a file.
2. **More Complex Searching:** Click on the small “+” next to the feature selector to extend your query.
 - **‘and’:** Allows you to add another feature, and the search will return all segments containing *both* the nominated features.
 - **‘or:** Allows you to add another feature, and the search will return all segments containing *either* of the nominated features.

NOTE: *and* and *or* cannot be mixed!
 - **‘and not’:** Allows you to add another feature which should be excluded, and the search will return all segments containing the first feature but *not the second feature*.
 - **‘containing segment’:** this allows search across layers: it returns all units tagged with the first feature which contain segments at another layer tagged with the second feature. For instance, one might search for ‘finite-clause containing person&subject’, to find all finite clauses where the segment boundaries totally include a segment at the participant layer which is coded both person and subject.

- **‘containing string’**: this will allow you to find all segments with the nominated feature which contain a given string. Matching is not case sensitive.

NOTE: this feature is also used for **concordance searching** (searching based on lexical features, wildcard matching, etc. See below for more details).

- **‘in segment’**: this allows search across layers, specifying that segments should match only if they are contained within segments at the second specified layer. For instance, one might search for ‘person in editorial’ to find segments tagged as person in editorials.

Immediate containment: NOTE: for search queries including ‘containing segment’, ‘containing string’ or ‘in segment’, you can choose between “immediate” and “anywhere”. The difference is as follows:

- *anywhere*: if the containing segment contains the specified segment or string, it will match.
- *immediately*: Sometimes users allow units embedded within others at the same layer, for instance, clauses can be embedded within other clauses. If you specify 'immediately', then if the contained segment or string falls within such an embedded unit, it will not match the units in which the unit is embedded. For instance, with "[They left because [she was tired]]", a search for: `clause containing immediately 'was'` would only match the inner clause.

3. **Combining Complex Searches:** One can combine complex searches, e.g.,

person containing immediately "bush" in finite-clause in editorial&english

3 Concordance Searching

CorpusTool lets you search for lexical patterns (English only currently for most features).

3.1 Specifying the Search Query

If you specify “containing string” (see above), you can specify a lexical pattern instead of a simple string. For example, to find passive clauses, "be% @participle" will match all segments containing any form of 'be' followed by a participle verb (-en verb).

Note that the corpus is NOT tagged in terms of part of speech (POS). Rather, CorpusTool includes a large dictionary of English, and looks up each word in the dictionary. Because of this, a word will match all POS classes to which it belongs. For instance “be%” will match all occurrences of “being”, even in the context where the word is not a verb, e.g., “the being”.

Matching occurs as follows:

Case Insensitive: all searching is case insensitive. Thus ‘Birch’ will match ‘Birch’ and ‘birch’ and “BIRCH”.

The search string consists of a sequence of search tokens separated by a space. Each search token can be of the following format:

- 1) *Literal token*: a token not containing *, #, @ or % will match the token itself only.
- 2) *Wildcard token*: if the query token includes an "*", the "*" will match any number of chars. Thus
 - ca* matches 'cat', 'carburettor', etc.
 - *ed matches 'weed', 'lived', etc.
 - bro*en matches 'broken', 'Brollerglen', etc.
 -
- 3) *Match any*: a '#' by itself matches any single token.

(The above 3 cases should work for any language where words are divided by space characters or punctuation)

- 4) *Constraining by class*: a wildcard form can be followed with '@' and then a lexical feature, and the form will match only tokens which, according to the system's lexicon, can take that lexical class. E.g.,
 - ca*@noun matches nouns starting with 'ca'.
 - *ing@mental-projecting matches mental-projecting verbs ending with 'ing'.

An asterisk cannot appear by itself, it must have text either before or after it.

A full list of the lexical features that can be used are in Appendix II, and can be seen within the tool by selecting "Show Wordclass Network" from the Misc menu of CorpusTool.

- 5) *General class matching*: If no token string is provided before the '@', then the query form matches all tokens which could represent the specified class. E.g.,
 - @noun matches any noun form
 - @verb matches any verb form
 - @adverb matches any adverb form
 - @mental-projecting matches any verb which is classified as mental
 - @human-noun matches any noun classified as a human-noun
- 6) *Inflection matching*: '%', at the end of a token indicates that all inflection forms of the token, which should be a root form, should be matched. Thus,
 - break% matches 'break', 'broken', 'broke', 'breaking', 'breaks'
 - red% matches 'red', 'reds' (noun), 'redder' (adj), 'reddest'
 - be% matches 'be', 'is', 'are', 'was', 'were', 'been', 'being'
 - is% matches nothing (only roots can be used)

To constrain the inflection matching to a limited set of inflections, one can add 'noun', 'verb', 'adjective' or 'pronoun' after the '%'. E.g.,

- red%noun matches 'red', 'reds'
- red%adjective matches 'red', 'redder', 'reddest'

Note that wildcards cannot be used within % forms. Nor can the string before the % be blank.

4 Running a Query

After entering your query, you can hit the “Show” button. If your cursor is in a text field (Containing String), you can hit the Return Key.

5 Modifying a Query

To change a feature selection, just click on the feature to change it.

To delete any of your search extensions, click on the keyword (“&”, “/”, “containing”, “in”) and click on “remove”.

6 The Result Space

The white space below the Query space displays the results. Click on a result and the annotation file containing this segment will be opened at the right place.

The three columns at the left indicate the state of each coding:

- P/- Whether or not the segment is totally coded (P=partial)
- */- whether the segment has a comment associated. Click on the segment to see the comment.

Section 6:

Automating Coding

1. Introduction

The Autocode window allows you to assign features to existing segments using search patterns. For instance, we can identify passive clauses in English using a pattern like:

'clause' containing 'be% @participle'

Using the Rule Editor, we define a rule like:

Rule: select passive-clause if clauses containing immediately 'be% @participle'

(**Note:** as with Search, lexical-based search patterns currently work for English)

We can then press the "Show" button, and all instances matching the search query are shown, with a check-box next to each. We can uncheck any item which is a false match (not truly a passive). Clicking on "Code Selected" will then assign the "passive" feature to each of the selected segments.

In this way we can quickly code many of the more common grammatical patterns. To see a sample of such autocode rules, add a new layer to your project, and use the scheme included with the system "clauses.xml". This includes rules for process type (mental, verbal, etc.), voice (active, passive), modality, nonfinite clauses, etc.

1. **Opening Autocoder:** Click on the Autocode button on the main window of CorpusTool.
2. **Adding a new rule:** To add a new rule, click the "Add" button in the list of buttons at the top of the Autocode window. A window like the following will appear:



Select a feature which you want to code automatically. Then specify a search pattern to use (see section on "Corpus Search" for how to specify a search query). Then press "Save" to keep this rule in memory.

3. **Editing a rule:** Click on the Edit button to edit the currently displayed rule.

4. **Deleting a rule:** Click on the Delete button to delete the currently displayed rule.
5. **Coding with a Rule:** When you have a rule selected, press the Show button to see all segments which match the search pattern component. A new toolbar appears with three widgets:
6. **Display All/Agreements/Conflicting/Nonconflicting:** selecting from this list allows you to filter out some of the matches:
 - *All:* shows all of the matches
 - *Agreements:* shows all segments already coded with the specified feature.
 - *Conflicting:* shows those segments which are already coded with a feature which conflicts with the feature you are autocoding. For instance, if autocoding as 'passive', this would show all segments already coded as 'active'.
 - *Nonconflicting:* shows all segments which are neither agreements nor conflicting.
7. **Select All/None:** selecting one of these options will select/deselect the check boxes next to each segment.
8. **Code Selected:** Clicking on this button will automatically code all displayed segments which are selected.

Hints

- For some grammatical phenomena, you can provide a pair of rules like:
- Select passive if contains 'be% @participle'
- Select active if clauses and not passive
- Use the first rule to code passives and then use the second rule to put everything else as active.
- Provide one rule such as the passive rule above. Code these. Then edit the rule, inserting a # between the search terms, e.g.,
- Select passive if contains 'be% # @participle'
- This will find some instances where 'not' or an adverb falls between the verbs.

Section 7:

Corpus Statistics

1 Introduction

The Corpus Statistics pane allows various statistics to be derived from your tagged corpus. Press the “Statistics” tab on the main window’s toolbar to see the Statistics pane (as in Figure 7.1).

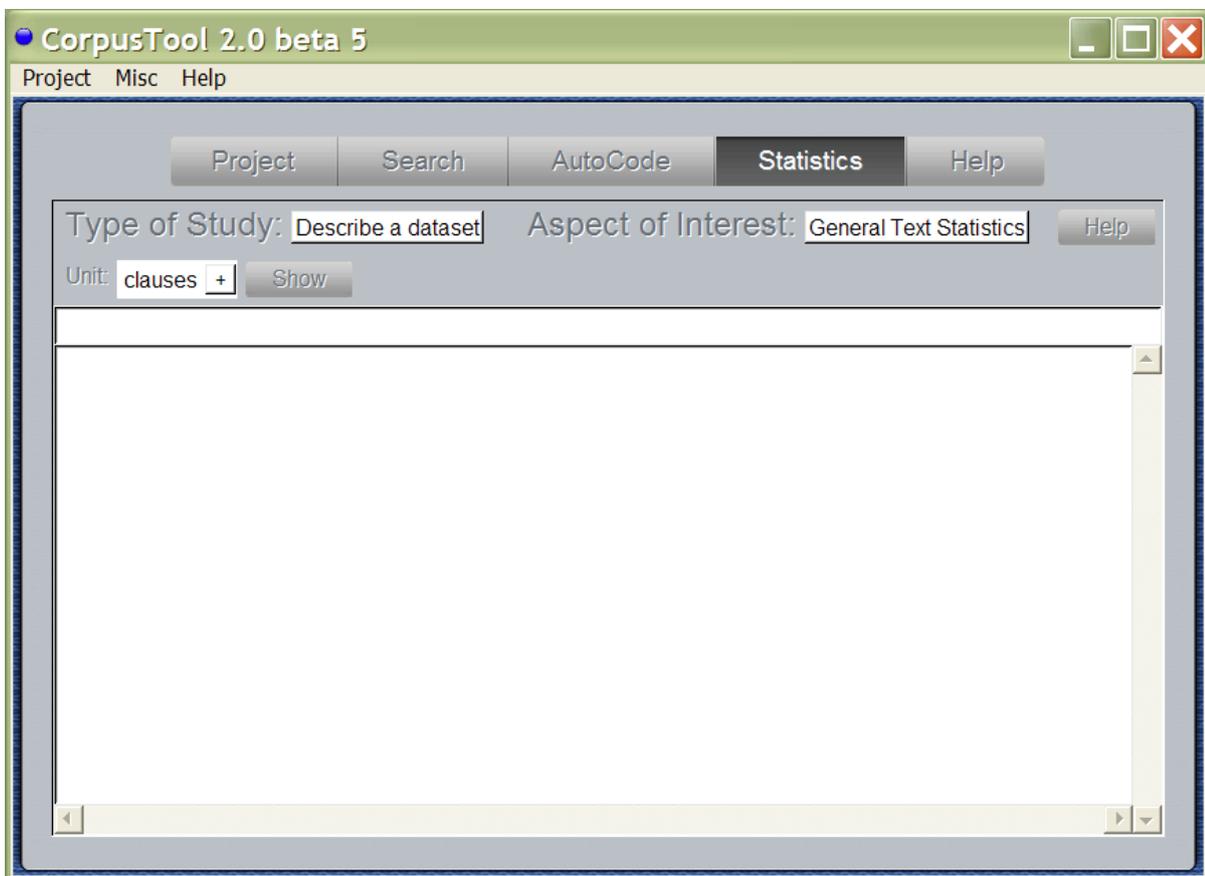


Figure 7.1: The Statistics Pane

You can use this interface to perform two kinds of studies on your corpus:

1. **General Text Statistics:** offers general statistics of the corpus, such as total number of segments, number of words per segment, lexical density in the corpus, pronominal usage, etc.
2. **Feature usage:** you specify a feature in a layer (most typically, the root feature of the layer), and the program describes the usage of features in the corpus at that layer (counts, mean, and standard deviation).

These studies can be done for a single dataset (descriptive statistics), two datasets (comparative statistics), or showing results for each document individually:

1. **Describe a dataset:** offers descriptions of your corpus, or a specified subcorpus.
2. **Compare two datasets:** provides a comparison of two subsets of your corpus (e.g., english vs. spanish). When Feature is selected, the two sets are contrasted in terms of the occurrence of presence of the features in the codings at the layer specified. Levels of significance of the differences between the sets are displayed, both in terms of Students T-test and Chi-Squared (see below).
3. **Compare Multiple Files:** provides details of each file in your corpus, one column per file.

2 A Contrastive Feature Study

Figure 7.2 shows a sample Comparative study done using the “Finance” project. Note very little of the text has been annotated, so the results are for small numbers only. We would need to tag a thousand or more participants from a range of editorials and fpn (front page news) articles before we could start to trust the results. This preliminary study shows two significant results (more reference to people rather than organisations at a 98% level of significance; and significant differences in the types of organisations discussed), but the numbers are too low to trust.

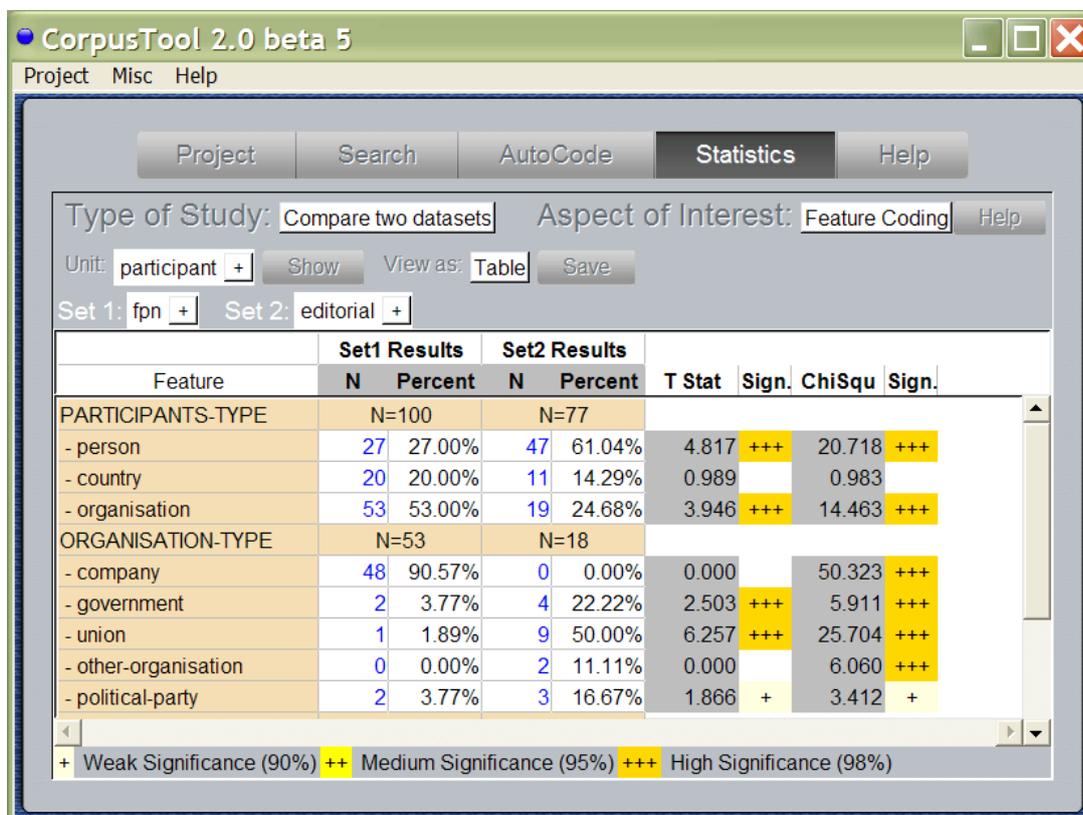


Figure 7.2: A Contrastive Stats Study

3 Performing a Study

To perform one of the studies outlined above:

1. **Choose one of the options from the “Type of Study” menu:** ‘describe a dataset’, ‘compare two datasets’ or ‘compare multiple files’.
2. **Choose from the “Aspect of Interest” menu:** choose either ‘Feature Coding’ or ‘General Text Statistics’.
3. **Specify the unit that you are interested in** (see section 5, part 2: *Specifying Search Queries*). This should be the unit which you wish to explore differences in. It could be the root feature in a network (as in the case in Figure 7.2), or a more delicate one.
4. If you are selected “Compare two datasets”, then **enter a feature in the Set 1 space and another in the Set 2 space**. This should be a unit which CONTAINS the unit of interest. In this case, we specify units of the Register layer, *fjn* and *editorial*. Since these features apply to whole texts, they do contain the segments with feature “participants”.
5. **Press Show.**

4 Interpreting the Results: Feature-based Studies

Only systems which are relevant are shown. For instance, if we had specified the unit of interest as “person” above, then the study would involve only those segments with feature “person”. For this reason, the results for this system are not shown, as “person” would score 100%, and the other features in that system 0%.

Counts and Percentages: The results for each feature are shown with both raw counts (how often that feature occurred in the dataset) and also as a percent. The percent shows the proportion of segments which have this feature. Note that the percentages in a system (a given set of choices) always adds up to 100%, so really what it is measuring the propensity to select this particular feature as opposed to the other features in the same system.

Statistical Significance: when a comparative study is done, it is possible to measure whether the differences between the two datasets is statistically significant (does it represent a real difference or is it possibly due to randomness in the data).

CorpusTool uses two measures of statistical significance, and presents them both in the results:

- **T-Statistic:** T-Stats are the numbers on which the level of significance of your result can be derived. The bigger it is, the higher the level of significance, but this also depends on how much data you have. In some more scientific papers, you might be requested to provide T-Stats, but it is quite rare in linguistics.
- **Chi Squared:** in recent years, particularly in linguistics, chi squared statistics are becoming the preferred means of testing significance. CorpusTool provides the Chi Squared statistics for each comparison, and the level of significance that corresponds to this.

At the end of each entry there will be between 0 and 3 "+" signs. These indicate how statistically significant is the difference of this features mean from that of the mean of the other set:

- (none) Not significantly different.
- + Significant at the 90% level (10% chance of error).

++ Significant at the 95% level (5% chance of error).

+++ Significant at the 98% level (2% chance of error).

The level of significance is important to establish how repeatable your results are. Results without significance may be accidents, and if we repeat the study with other texts, the result may be different. If results are highly significant they are likely to be repeatable if we apply the analysis to a totally different set of texts. To understand this, a single + means that of any 10 such results, you can expect one to be a false result (90% significance, or 10% chance of error).

5 Presenting Results as a Network

When performing a feature-based study, you can now view the results in a system network, instead of in table format. See Figure 7.3. After a study is presented in table form, a new menu is presented, labelled “View As”. Select “Network” to switch to Network view.

This way of displaying statistics has been copied from a similar feature in SysFan¹. I thank Canzhong Wu, the author of SysFan, for allowing me to use this feature.

¹ Available from <http://minerva.ling.mq.edu.au/units/tools/index.htm>

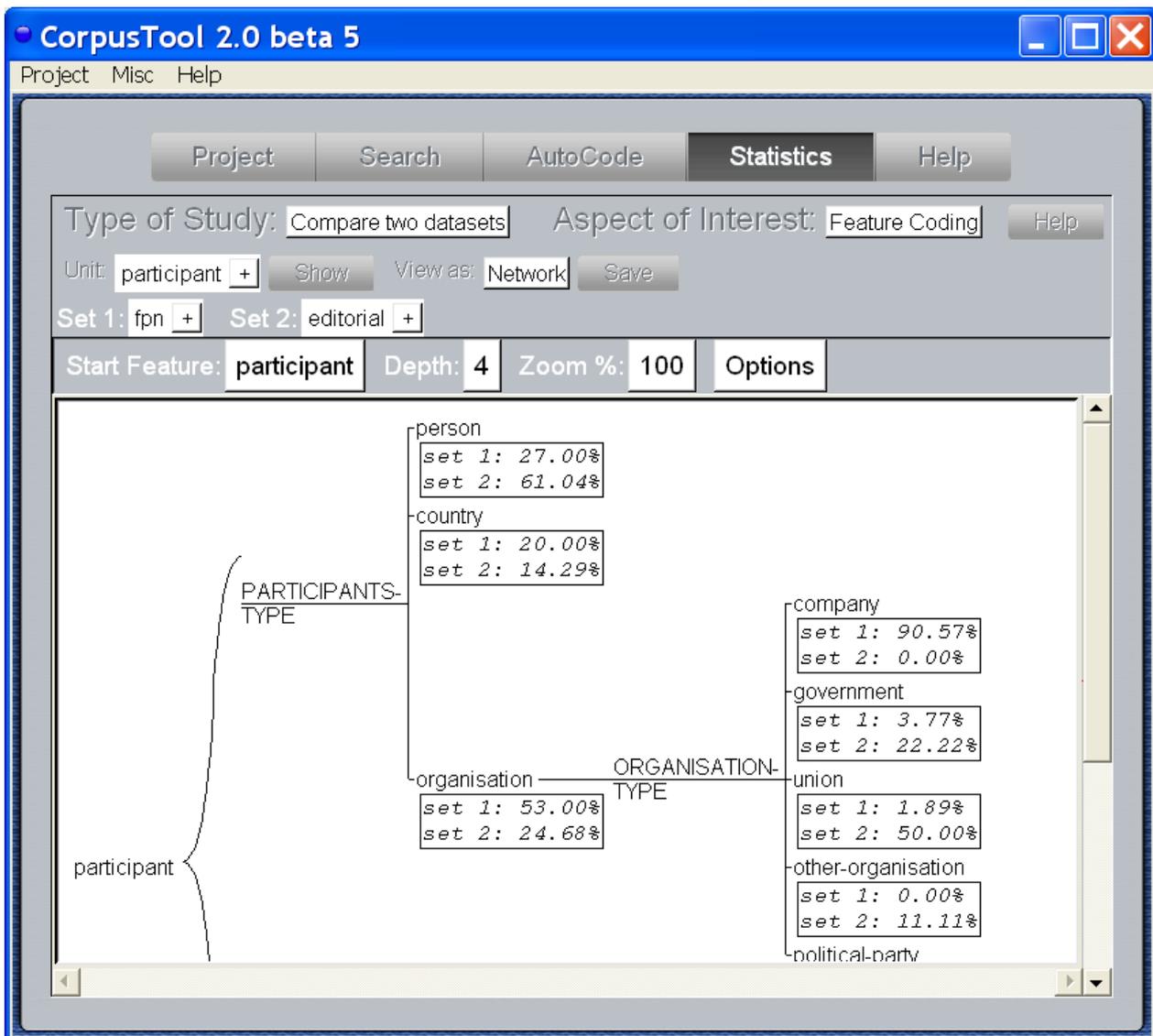


Figure 7.3: Network View of Statistics

6 Saving Statistics

Each Statistics window offers a “Save” button which allows you to save the results to file, in HTML format, tabbed delimited, or plain text.

Results saved in HTML can be opened in MS Word, and then cut/pasted into your publications.

Results saved tab-delimited can be opened in MS Excel (on Windows, right-click on the .txt file and specify Open with... Excel.) These files may also be useful for programs such as SPSS.

Section 8: Keywords

1 Keywords

The top words in any frequency list for English will be words such as “the”, “of” and “a”. A more informative listing works out how important each word is for a particular corpus, when compared with a more general corpus.

For instance, the keywords from a corpus split over three fields are shown below. The words are ordered in terms of their “specialness” for this corpus (relative frequency in this corpus when compared to the relative frequency in the general corpus). A value of 100 indicates the word appears 100 times more in this corpus than in other corpora.

NOTE: for this to work, one needs to select only a sub-corpus. If you select the whole corpus, then nothing will happen.

Military		Economics		Crime	
troops	100.0	economy	121.38	crime	142.85
weapons	100.0	companies	116.52	detective	50.0
engine	100.0	stock	100.0	police	49.16
mountains	100.0	tax	100.0	disappearance	40.0
smoke	90.0	cuts	85.0	criminal	39.86
gulf	85.0	profits	80.0	court	34.88
enemy	85.0	investment	75.0	justice	30.23
aircraft	80.0	billion	75.0	driver	30.23
force	70.0	returns	70.0	boy	29.06
civilians	70.0	sales	70.0	victims	18.6
civilian	70.0	earnings	65.0	family	17.56
guys	65.0	investors	65.0	child	13.95
military	62.47	jobs	65.0	car	12.81
squadron	60.0	package	65.0	lived	11.96
suicide	55.0	assets	65.0	officers	11.96
tanks	55.0	prices	60.0	legal	11.51
soldier	55.0	bill	60.0	children	10.57
jungle	55.0	corporate	60.0	kids	9.3
altitude	55.0	stocks	58.26	mercy	9.3
strikes	55.0	markets	55.0	investigators	9.3
trees	55.0	budget	55.0	woman	9.01
lieutenant	55.0	finance	50.0	murder	8.52
withdrawal	55.0	volatility	50.0	boys	8.52
missile	55.0	reforms	45.0	age	7.77
bomber	50.0	commercial	40.0	victim	6.64
invasion	50.0	temporary	40.0	street	6.27
combat	50.0	cent	37.87	body	6.22
rounds	50.0	analysts	32.04	incident	5.98
missions	45.0	growth	32.04		

2 Phrases

Rather than looking at single-words, n-gram analysis looks for sequences of words which are common in the corpus. For instance, a list of the frequent 3-

grams (sequence of 3 words) that occur in a small corpus of introductions to academic papers are shown below:

in terms of	12	ad hoc networks	6
a set of	11	we believe that	6
in this paper	10	of this paper	6
the performance of	7	terms of a	5
of the two	7	in section 4	5
be able to	7	some of the	5
a number of	7	in order to	5
the design of	7	large number of	5
which can be	7	that can be	5
the problem of	6	ad hoc network	5

According to Biber (e.g., Biber and Barbieri 2007), as the corpus grows to a reasonable size (millions of words), the kinds of phrases that raise to the top don't contain lexical content as such (e.g., 'ad hoc networks'). Rather, they are phrases which are used to frame such meanings. We see here: "in terms of", "a set of", etc.

While keywords tell us which words we should teach in a text, n-grams can tell us which phrasings are usefully taught. For instance, assuming we were teaching students how to write introduction sections to academic papers, we collect a corpus of such texts and produce the key n-grams for various lengths. From such a corpus, we can pick up frequent phrases such as "this paper reports on" or "this paper/article is organized as follows".

3 Key Features

'Key Features' does what 'keywords' does, except that, rather than looking at the words in the text, it looks at the features assigned to segments. The software thus shows which features are special to the focus corpus, as compared with the reference corpus.

A key value of '2.0' indicates that the feature is twice as common on the focus corpus as in the reference corpus.

Section 9: Text Styling

4 Text Styling

It is sometimes useful to view the coding of a text visually. CorpusTool allows you to view one of the text files of your project, specifying that particular segments (on whichever layer) should be showed in bold, italic, underline, larger font or coloured. See Figure 8.1 for the text style view of a file within the “Finance” project.

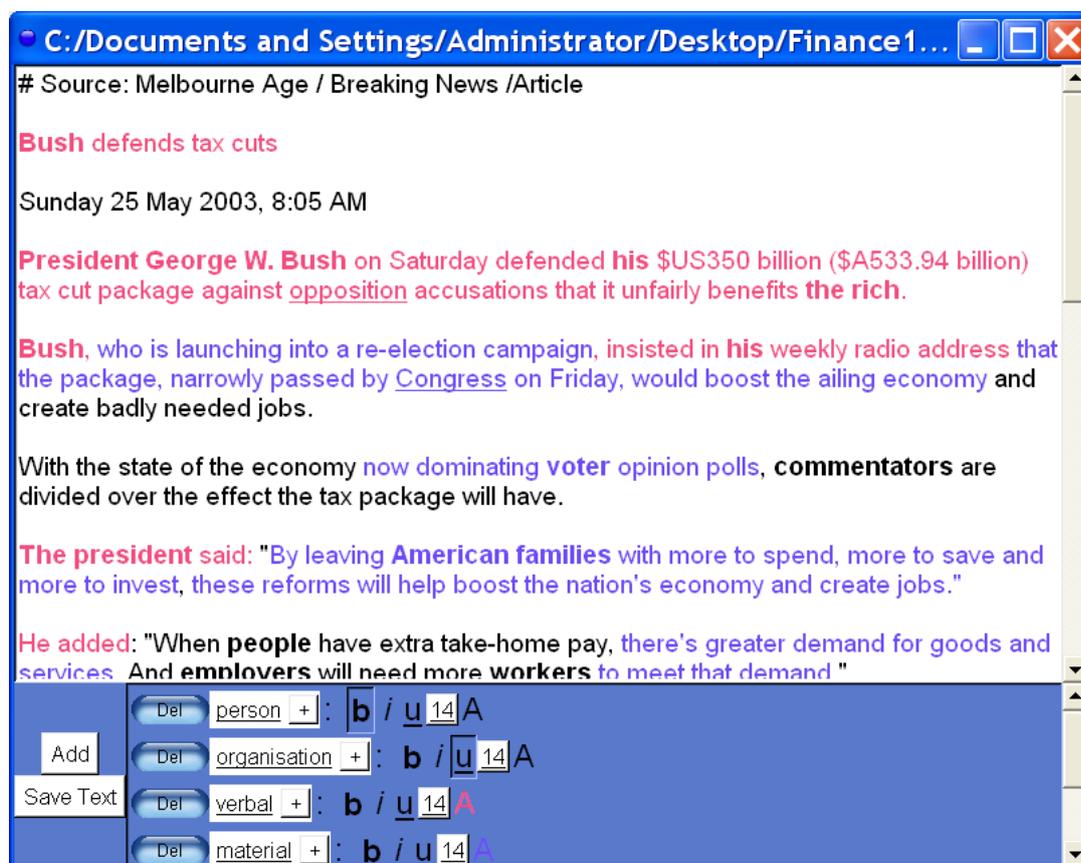


Figure 8.1: The Text Styler Window

5 Opening the Text Styler

From the Project window (the main window), click on the filename of one of your files. Note, this only works for files incorporated in your project. Also, your project needs to have at least one layer defined.

6 Styling the Text

You can assign colour and/or font effects (bold, italic, underline) to all text tagged with a given feature, or feature combination. This allows the patterns of selection throughout the text to be visible.

E.g., use bold/italic/underline for appraisal categories, and colour coding for clause type to see how appraisal is distributed in respect to clause types.

7 Saving Styled Text

You can save styled text to an HTML file. To include styled text in an MS Word document, open the HTML file in MS Word, and from there cut/pasted into your own document.

Section 10: The Menubar

1. Merge Projects

Up until Version 2.8.4, this function did not work correctly.

Before Merging: To ensure cleaner merger, delete any layers in either project which have no annotation associated with them. In all cases where the same files exist in both projects (e.g., schemes, corpus files and annotation files), the file in the current project will be preserved. Files will moved from the other project where they do not exist in the current project.

- Select “Merge projects” from the Project menu.
- You will be asked to select another project to merge with the current one.
- Results will be saved in a new folder with the same name as the current project, plus "-merged" added.

Appendix I:

Importing Systemic Coder Studies

1 How to Import Coder Studies

The analysis files in Systemic Coder can be imported into CorpusTool. To do so, follow the following instructions:

If you have a single file to import:

1. Ensure that the coding scheme is saved as an external file (master scheme). To do this, open the file in Coder, and select “Scheme Storing...” from the Options menu. Select “Save to Master” and specify a location to save the scheme.
2. Ensure the codings are saved as .cd3 not .cd2: if the file on disk has a .cd2 extension, you need to open the file and select “Save Codings As” from the File menu. The program will offer to save it as a .cd3 file.
3. Now, make a new folder and place within it the scheme file and the codings file.
4. Open CorpusTool and create a new project.
5. Select “Import Layer” from the Project Menu.
6. You will be asked to specify the folder created in (3) above.
7. The .cd3 file will be split into the raw text (to be put into your Corpus folder) and the analyses (placed in the Analyses folder). The next window asks in which subcorpus folder to place your text file.
8. The analysis scheme is imported as a new layer. The next window asks for the name of the layer.
9. In Coder, the only way not to code a bit of text was to ignore it. In CorpusTool, one selects only the bits of text one wants to code. You may thus want the ignored segments in your Coder study to disappear. The next window allows you to do this.
10. Press Finalise, and you have a new Layer added, and your cd3 file is imported.

If you have a set of files, all annotated with the same scheme:

1. Place all the Coder files in a folder.
2. Make sure ALL the files are in .cd3 format, not .cd2.
3. Follow step (1) for a single file, for at least ONE of the files (e.g., make sure there is a .scheme file in the folder)
4. Proceed from step (4) for the single file case above.

If you have one or more files, where the same text(s) have been coded with different networks (in a sense you have done multi-layered annotation using Coder):

1. For each set of files annotated with the same scheme, create a folder and place the coder files and the scheme file for that analysis. (ensure the files are in .cd3 format).
2. Ensure all files which are analyses of the same file have the same file name, e.g., if you have Text1-CLAUSE.cd3 analysed for clauses, and Text1-GROUP.cd3 analysed for groups, rename both files to Text1.cd3. (CorpusTool can only tell two files are analyses of the same text by having the same filename).
3. Open a new project and use the Import Layer option as described above for one of the folders.
4. Repeat (3) for the other folders.

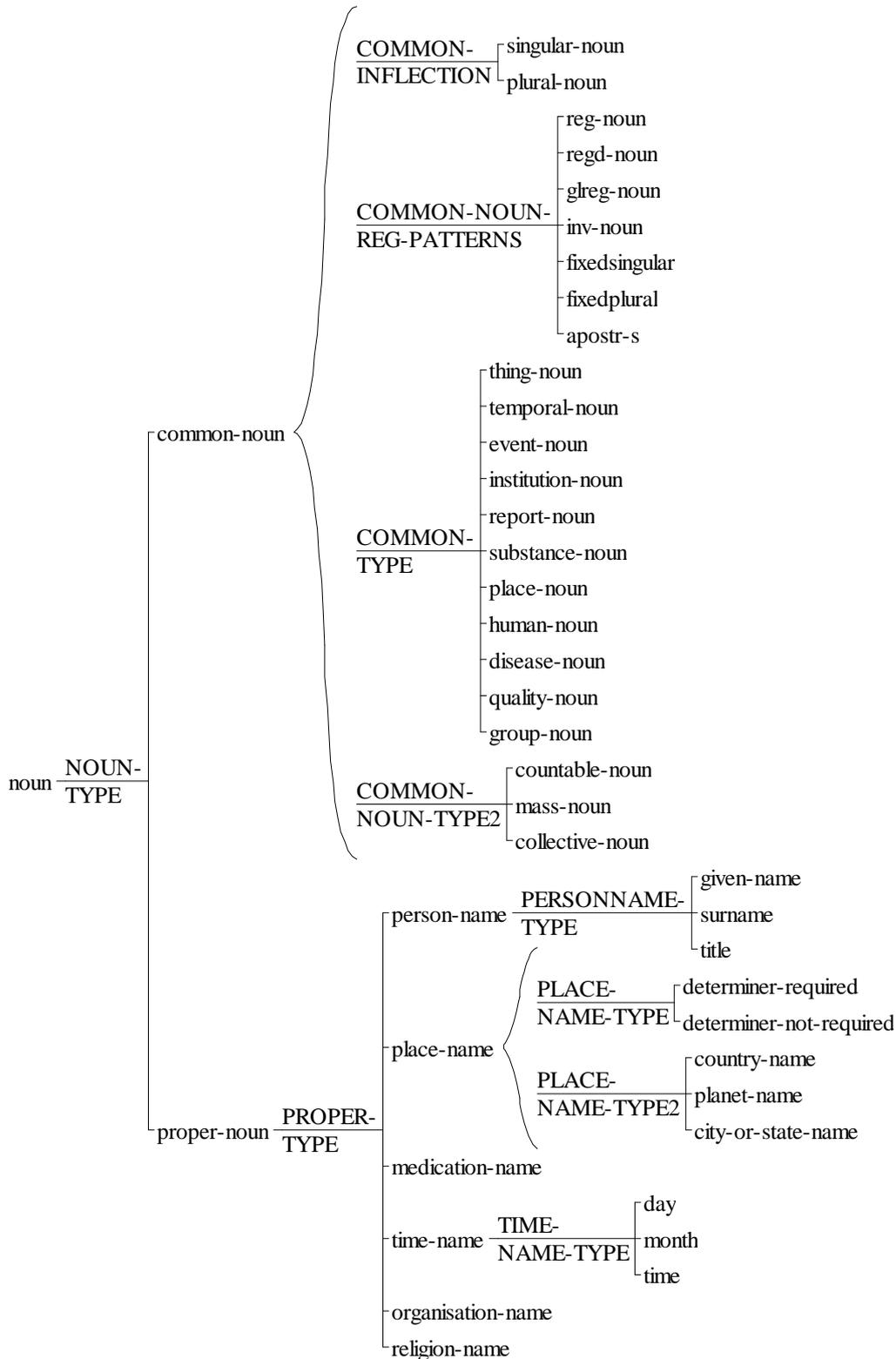
Some problems may arise:

- CorpusTool says it cannot read one or more of your .cd3 files: it may contain characters which are outside of ASCII text. CorpusTool should handle this, but currently cannot. Send me your files and I will import it for you.
- If you have any other problem importing cd3 files, send them to me (make a zip of the folder) and I will look at it (this is good for me, to see the kinds of problems people are having, so I can fix them).

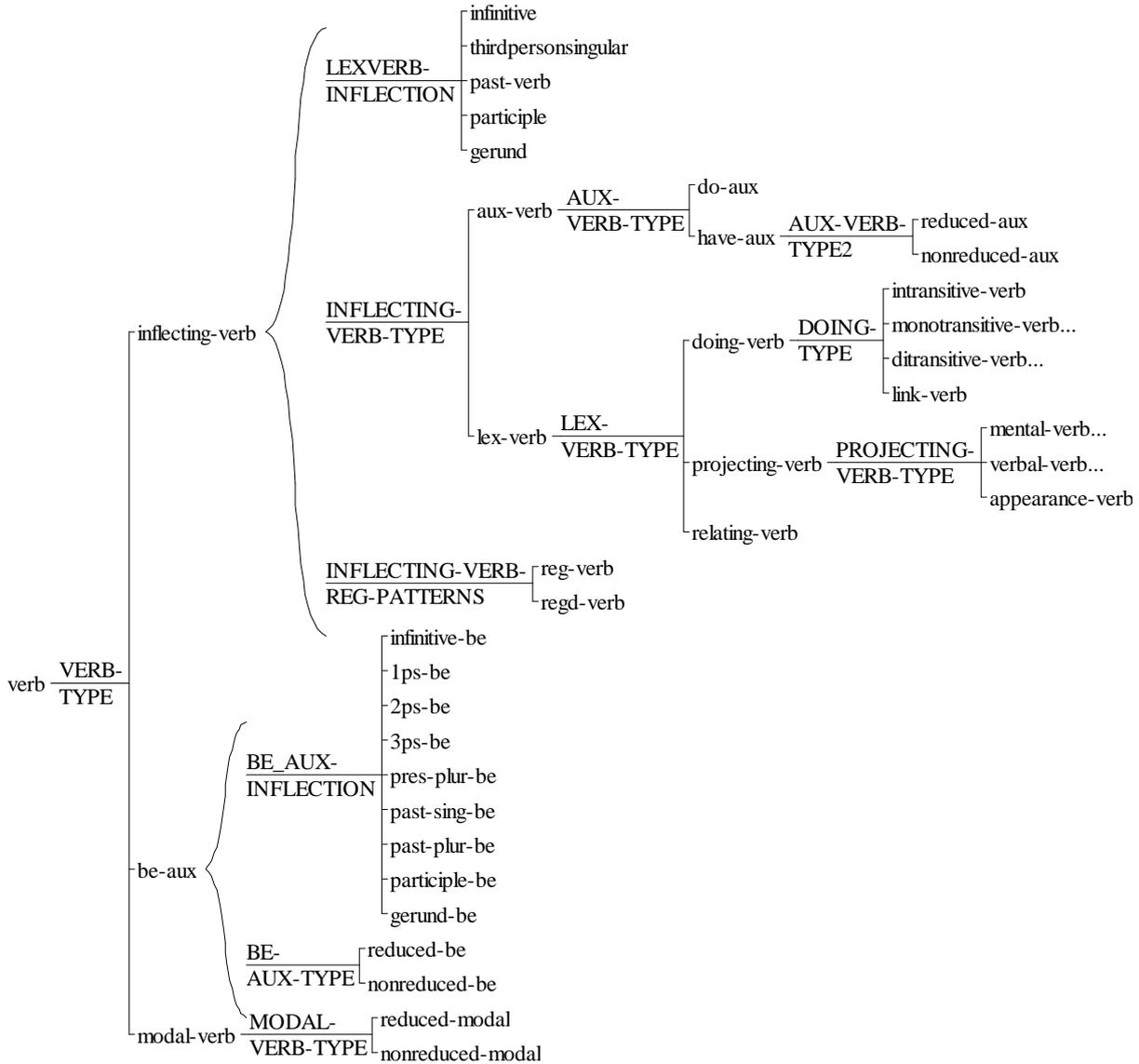
Appendix II:

Lexical Features for Concordance Searching

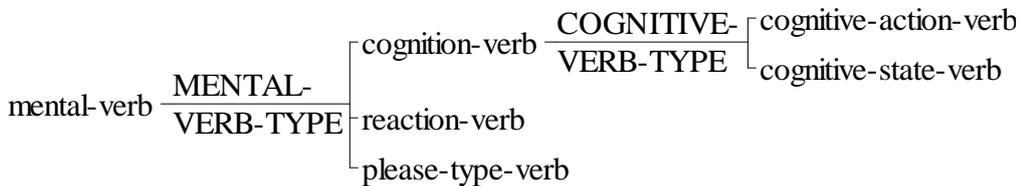
1 Nouns



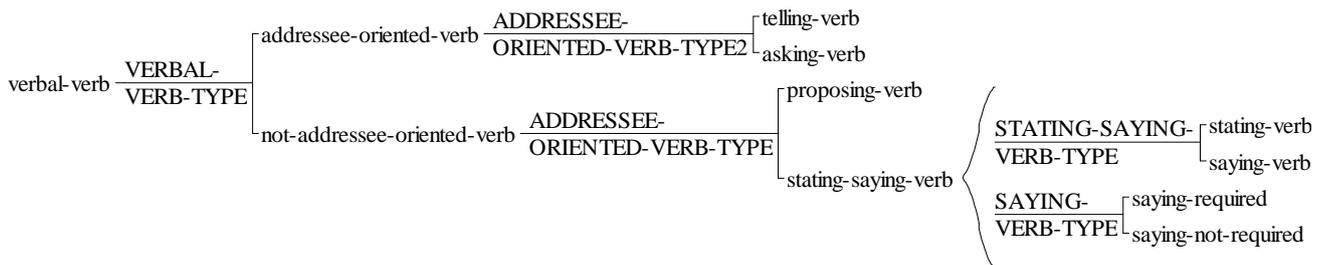
• Verbs



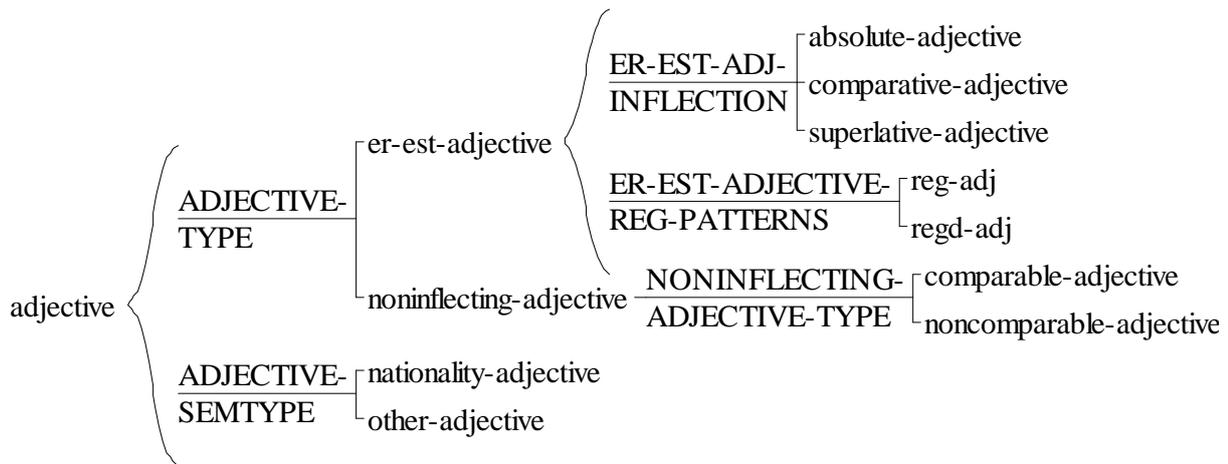
Subclasses of mental verb



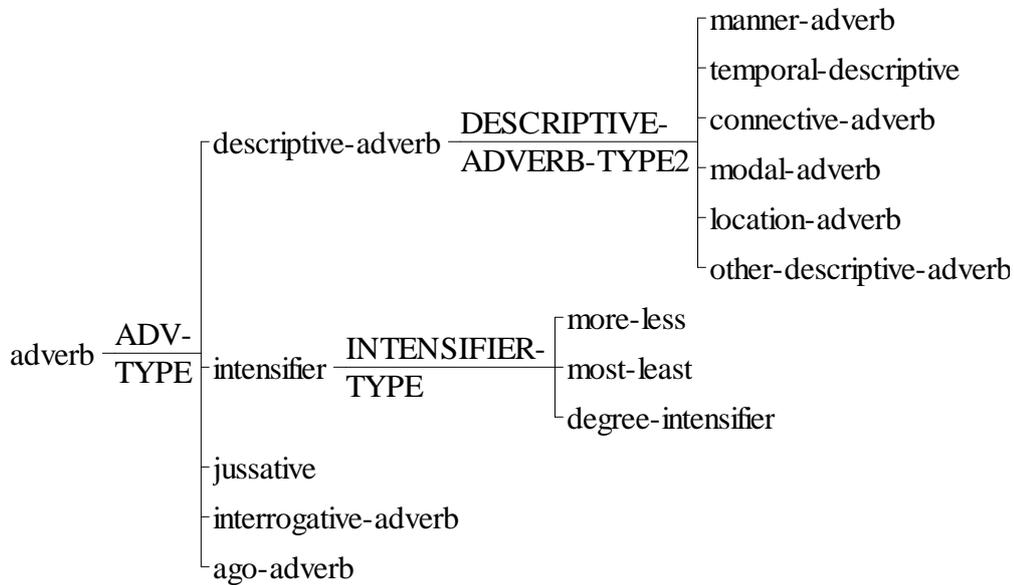
Subclasses of verbal verb



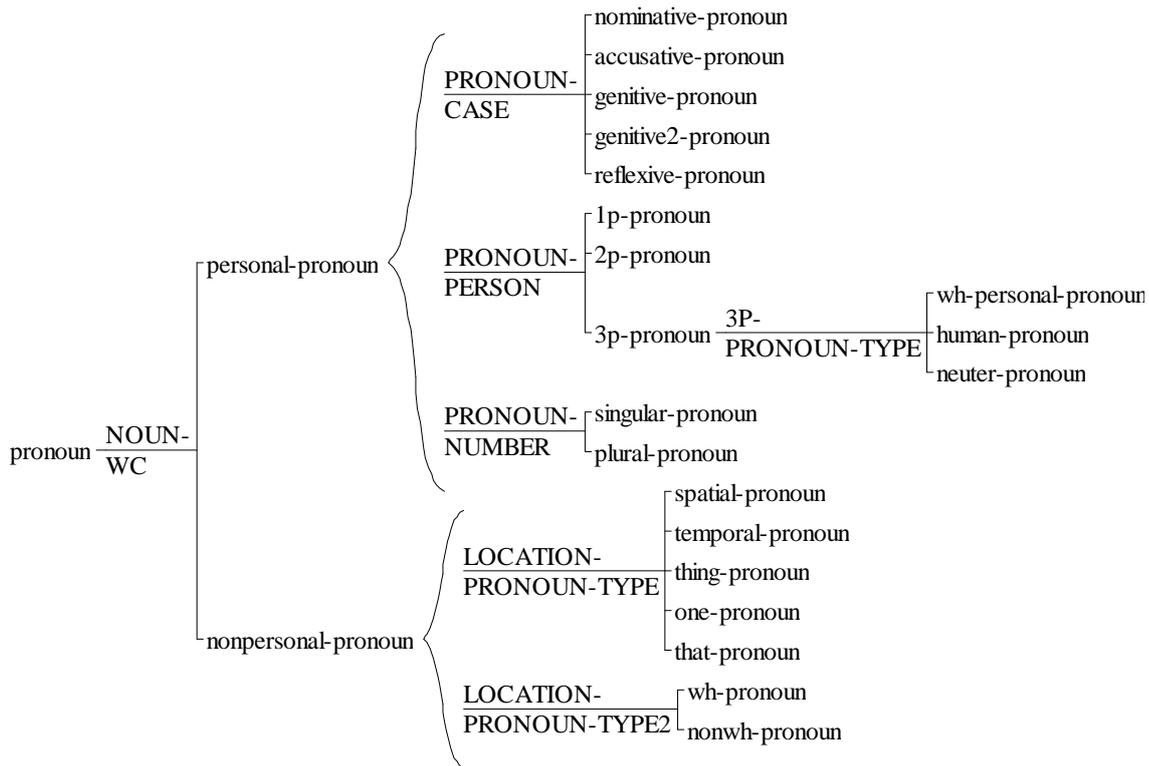
• Adjectives



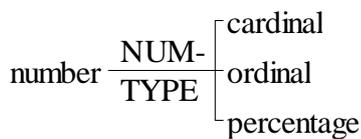
• Adverbs



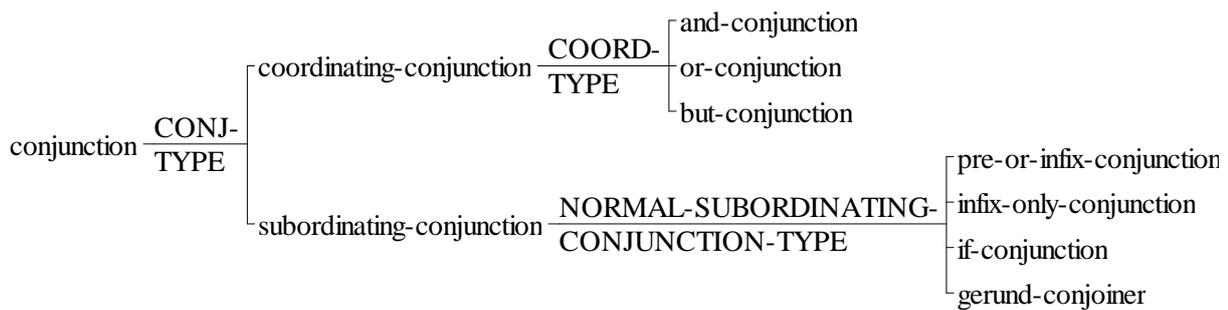
• Pronouns



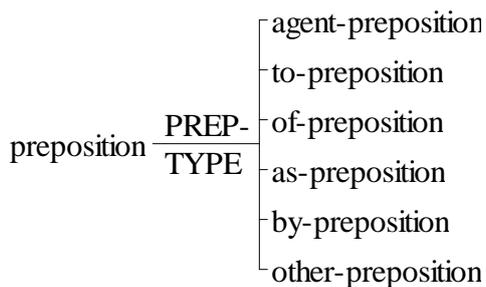
• Number



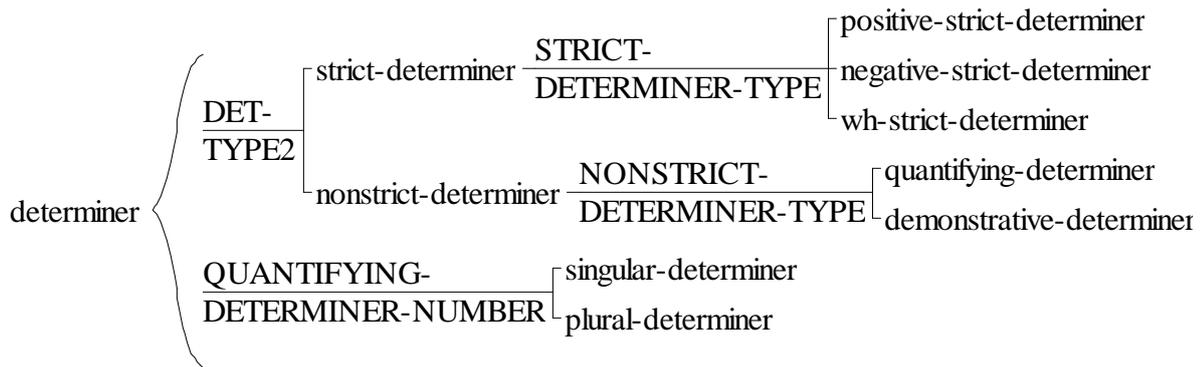
• Conjunction



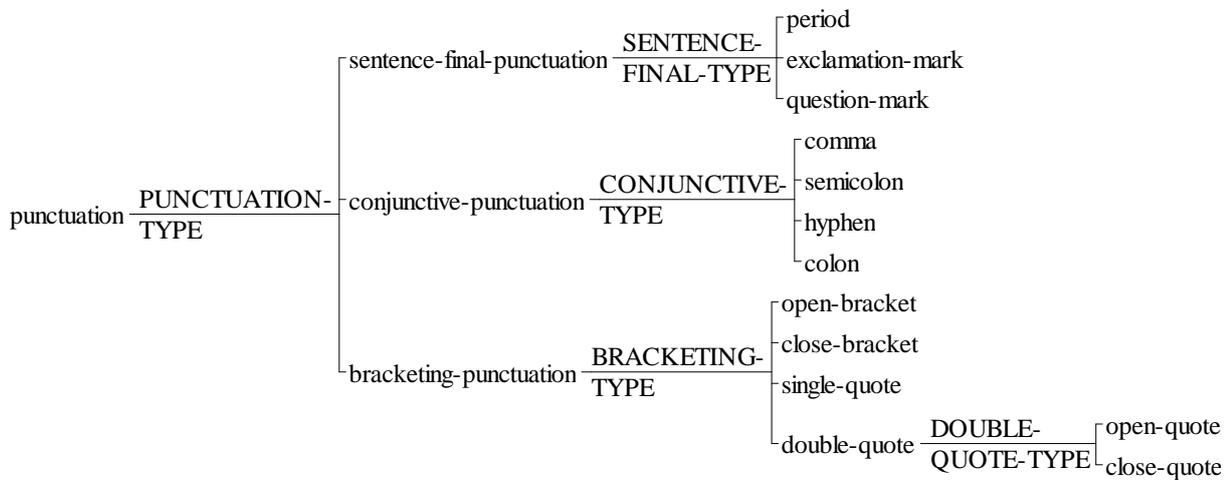
• Prepositions



• Determiners



• Punctuation



genitive-s