UAM CorpusTool

Version 2.6 User Manual

(October 2010)

Mick O'Donnell

michael.odonnell@uam.es

Contents

Section 1: About the UAM CorpusTool 4

Section 2: Starting a Project **5**
1 Starting a New Project 5
2 Adding a Layer 7
3 Adding Files To Analyse 9
4 Actions on Incorporated Files 12
4.1 Changing File Metadata 12
4.2 Viewing General Statistics for a file 12
4.3 Unincorporating a File from the Corpus 13
4.4 Opening an Annotation Window 13
5 Quitting CorpusTool 14
6 Continuing a Project 14

Section 3: Defining the Coding Scheme **15**
1 Opening the Scheme Editor 15
2 Editing the Scheme 16
3 Adding “Glosses” to features 19
4 The Options menu 19
5 Producing Images for inclusion in documents or the web 19

Section 4: Annotating Files **20**
1 Annotation Types 20
2 Annotating Code-Document files 20
3 Annotating Code-Segment files 21
3.1 Making, Moving and Selecting Segments 23
3.2 Ignoring Segments 23
4 Annotating Image Files ..... **¡Error! Marcador no definido.**
5 The “Other Actions” Menu 23

Section 5: Corpus Search **25**
1 Introduction 25
2 Specifying Search Queries 26
3 Concordance Searching 27
4 Running a Query 28
5 Modifying a Query 28
6 The Result Space 28

Section 6: Automating Coding 29
1. Introduction 29

Section 7: Corpus Statistics 31
1 Introduction 31
2 A Contrastive Feature Study 32
3 Performing a Study 33
4 Interpreting the Results: Feature-based Studies 33
5 Presenting Results as a Network 34
6 Saving Statistics 35

Section 8: Text Styling 38
1 Text Styling 38
2 Opening the Text Styler 38
3 Styling the Text 38
4 Saving Styled Text 39

Appendix I: Importing Systemic Coder Studies 40

Appendix II: Lexical Features for Concordance Searching 42

Section 1:

About the UAM CorpusTool

1 Introduction

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.
- 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.

Other annotation types will be added in later versions, allowing annotation of rhetorical structure theory (RST), Generic Structure (GSP), participant chaining, sentence structuring (e.g., Subj, Pred, Mood, Adjunct, etc.), annotation of spoken data etc.

UAM CorpusTool replaces prior software of the author, *Systemic Coder*, which allowed coding of single documents at a single layer. CorpusTool is an attempt to overcome the various limitations that constrained users of Coder. I wish to thank the many users of Coder who forwarded their comments over the years, and to thank those sending me comments on this new tool. To import Systemic Coder studies into CorpusTool, see Appendix I.

CorpusTool is available from:

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

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

Section 2:

Starting a Project

1 Starting a New Project

1.1 Open 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 .cptr file in the Project folder. This file has an icon as below:

MacOSX:

Windows:

The Opening Window

A window should appear as in Figure 2.1. This window provides, amongst other information, the version number you are using (useful if you need to communicate bugs).

The Window offers several options, *Start New Project*, or *Open Project*, to continue with a project you have already started. If you have opened a project previously on this machine, there will also be a button to open the last project opened.

Figure 2.1: The Opening Window

1.2 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 .cptr file).

Once you have finished with the *Create Project Wizard*, the *CorpusTool Main Window* will open, showing the *Project Management pane*. See Figure 2.2. This pane is where you control details of your project, such as which files are included, and what types of analyses are involved.

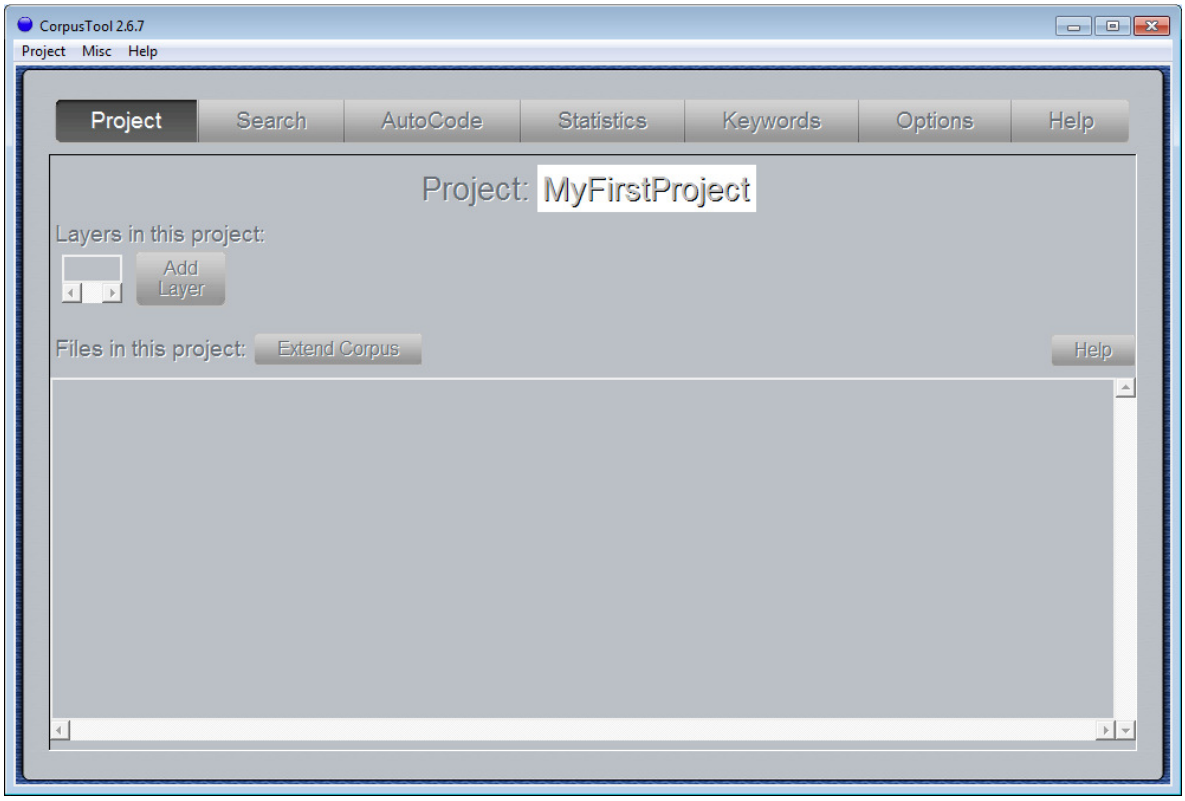


Figure 2.2: The Project Management pane

The buttons at the top of the pane allow you to switch between the different panes of CorpusTool: **Project** (this section), **Search** (section 5), **Autocode** (section 6), **Statistics** (section 7), **Keywords** (section 8) and **Help**.

We will assume for now that the "Project" pane is selected. The big letters at the top show the name of your project.

Below this is a space showing which analyses ('layers') are involved in the project. Initially this is empty.

Below the "Layers" space 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.

Let us first add a layer to the project.

2 Adding a Layer

The first thing to do in a new project is to specify what analyses we want in the project. Lets start by adding just one layer.

1. *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".

- 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). Lets 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 use 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.

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.

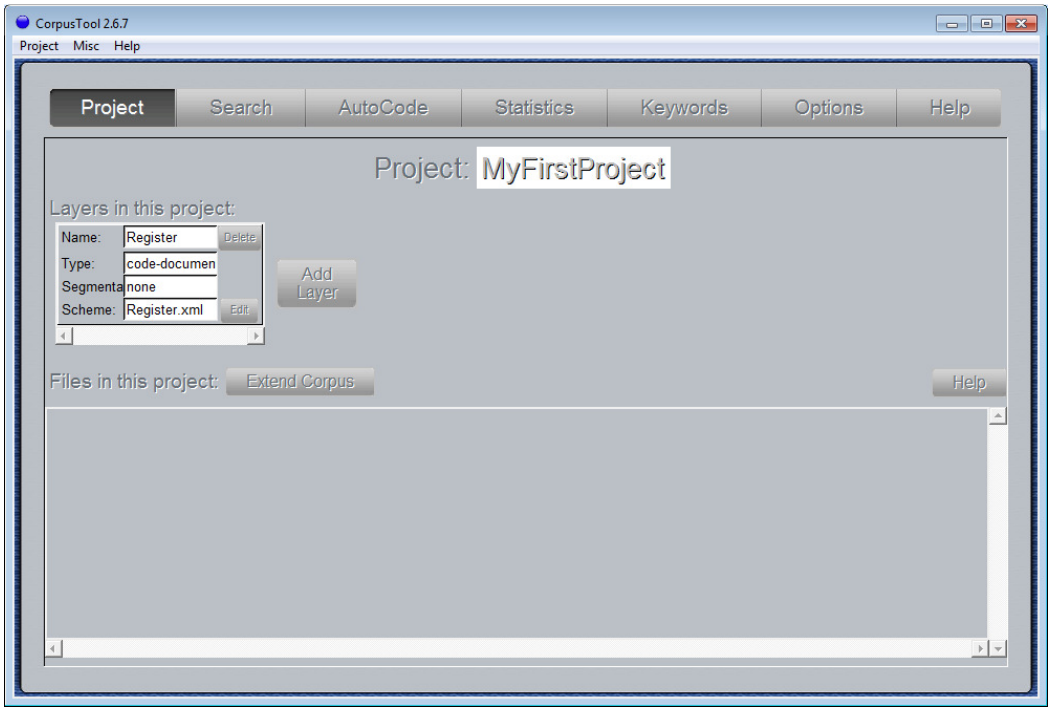


Figure 2.3: The Project Window with one Layer added

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.

3 Adding Files to Analyse

The next step is to add some files to the project. If, during creation of your project, you nominated text files to include in your project, they will be displayed in the File Pane of your Project Window. For now, I will assume you did not do this, so the File Pane will appear as in Figure 2.3, empty.

3.1 Extending the Corpus

To add files to your corpus:

1. Click on the *Extend Corpus* button: a "wizard" will appear to guide you through the process of adding files. You have a choice of adding a single file, or adding a folder of text files.
2. If you select to add a single file, you can either add it to an existing subcorpus (a folder within your project's Corpus folder), or to add it into a new subcorpus (in which case, a new folder will be created with the name you supply). In either case, the file you select will be copied from where it is into the subcorpus folder.

   If you select to add a folder, you select a particular folder on your disks, and it is copied into your project's Corpus folder.
3. Once the file or folder is nominated, click on the Next button, and then the Finalise button.

The file or files will now be shown in the File Pane (see Figure 2.4). The newly added files are under the caption "Files in corpus but not incorporated in project". CorpusTool 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 lets you get results at each point.

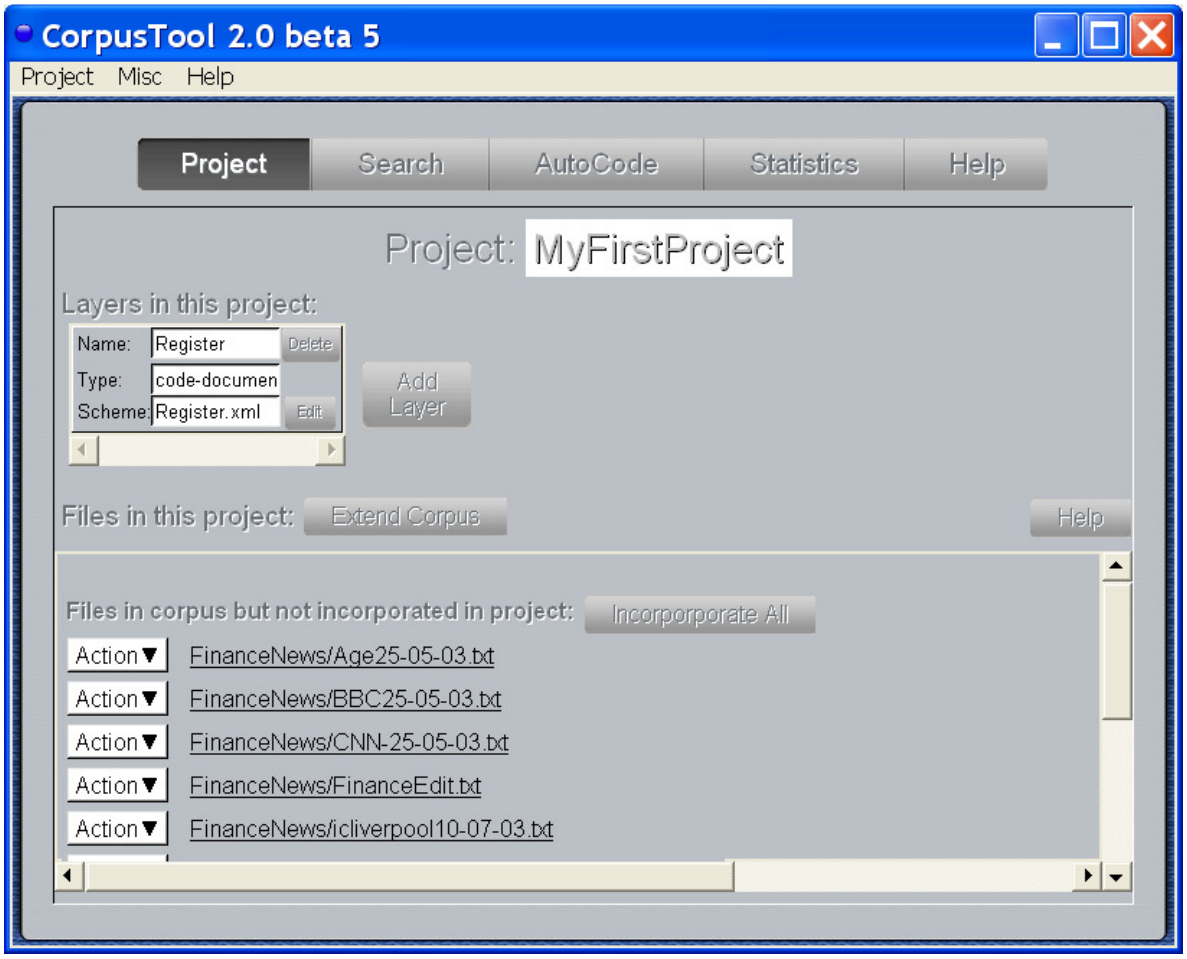


Figure 2.4: The Project Window after extending the corpus

3.2 Incorporating Files

To incorporate a file into the project, making it available for annotation, click on the "**Incorporate**" button next to that file.

Defining Language, Encoding, and Display Font: as you incorporate each file, you will be presented with a window asking for some metadata regarding the file (See Figure 2.5). 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 CorpusTool 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 Firefox, and

see which encoding it assigned using the “Character Encoding” sub-menu under “View”.

- **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.

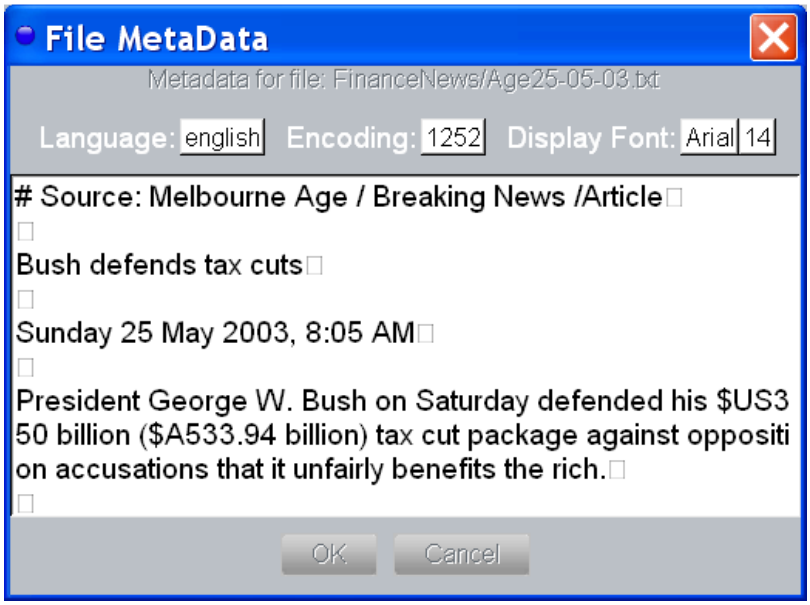


Figure 2.5: File Metadata Window

After incorporating two files, the Project Window appears as in Figure 2.6. Note that these two files appear at the top.

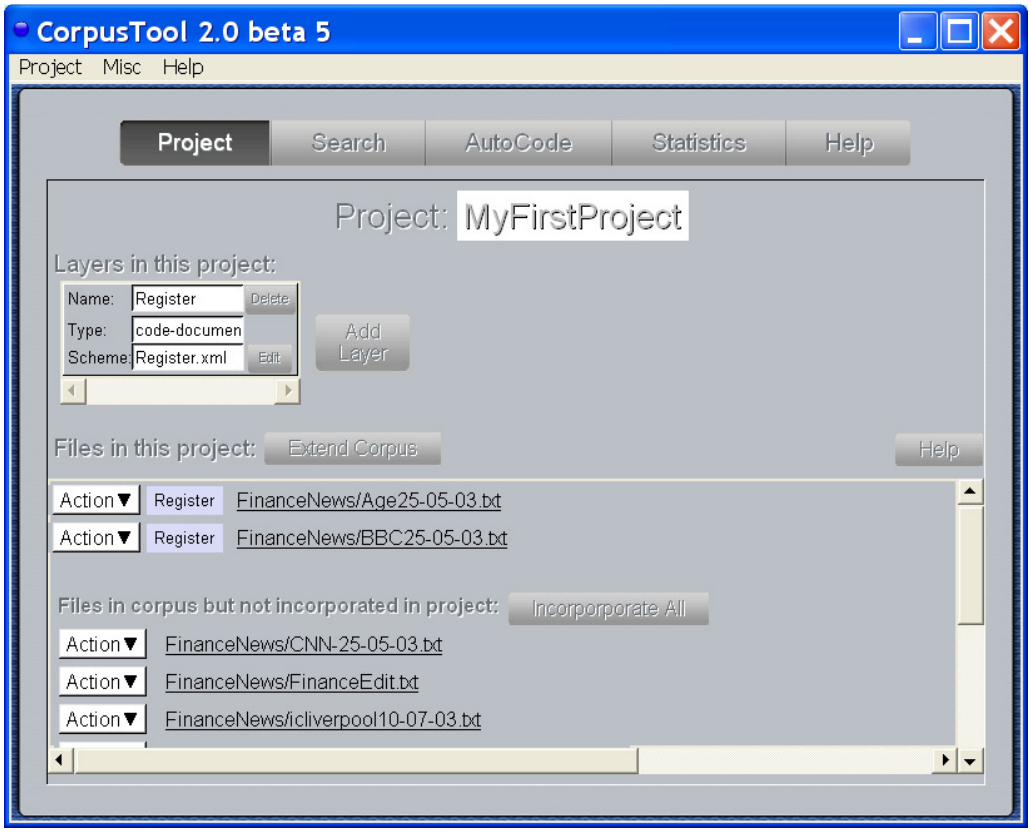


Figure 2.6: The Project Window after incorporating files.

3.3 Other Options on Unincorporated Files

The other options available for unincorporated files are:

- **Info**: gives some statistics about the text file, number of words, sentences, average sentence length etc. For English files, also a measure of lexical density, and some counts regarding pronominal usage (see below).
- **Delete**: removes the file from the corpus. Also deletes the file from the project's Corpus folder.
- **Filename**: clicking on the filename will present the text in full.

4 Actions on Incorporated Files

Once a file is incorporated, it offers one button for each defined layer. In the sample project, we have so far defined only "Register", so the incorporated files only have a button for this layer. As other layers are added, buttons for those layers will appear also.

4.1 Changing File Metadata

(Text annotation only) We saw above that when a file is 'incorporated', you are prompted to specify its language, encoding and display font. You can change these choices at any time by selecting "Change File Metadata" from the Action menu associated with each file.

4.2 Viewing General Statistics for a file

(Text annotation only) To see general statistics about each text file, select "View Basic Text Stats" from the Action menu on each row. This will provide some basic statistics

about the text which do NOT depend on any annotations of the file (see Figure 2.7). This includes:

- number of words in text;
- average word length;
- number of sentences in text (should work for European languages).
- average sentence length in words (again, for European languages).

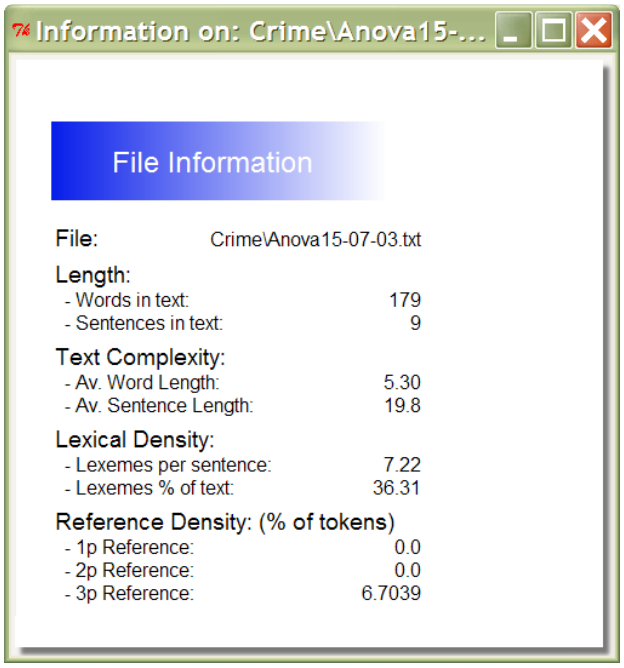


Figure 2.7: Info window for a file

For **English** texts, further information is available:

- **Lexical density**: in terms of average number of open class terms per sentence, or % open-class items in whole text.
- **Pronominal Reference Density**: detailing the usage of 1st, 2nd and 3rd person pronouns as a percentage of the text as a whole.

Note: as lexicons for other languages are added, these statistics will be available for those languages.

4.3 Unincorporating a File from the Corpus

The “Unincorp” button removes the file from the study.

WARNING: Any annotation done on that file will be deleted. The text file will be included in the unincorporated list, so you can add the file back in later (but totally unannotated).

4.4 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.

5 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.

6 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 3:

Defining 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. Click on the Edit button within the Layer Toolbar (see figure 3.1).

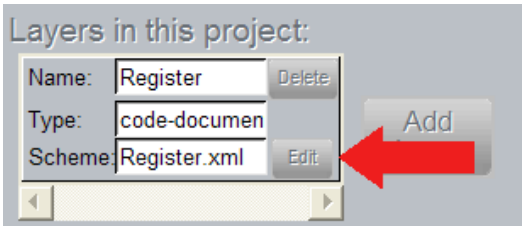


Figure 3.1: The Scheme Edit button

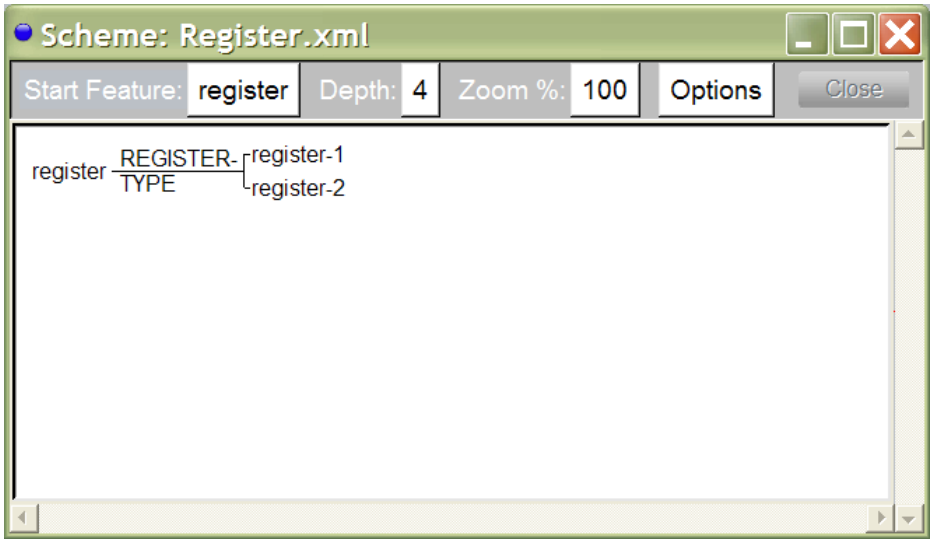


Figure 3.2: The Register Scheme before editing

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

2 Editing the Scheme

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

Click on “register-1” and a menu will appear with options, as in Figure 3.3.

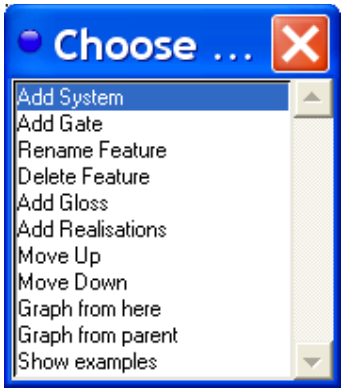


Figure 3.3: Options for Features

These options will be explained more fully below. For now, we want to change “register-1” to something more plausible. Lets assume that all of our texts are news articles, and that they are either front-page-news, or editorials. We thus want to change “register-1” to “fpn”, and “register-2” to “editorial”.

The important option is “Rename Feature”. Click on this option. A window will appear asking you to provide the new name for this feature. Type: *fpn* and then press Return.

Repeat the same process with “register-2”, and rename it as “editorial”.

Notice also that the choice between fpn and editorial has a name, automatically provided as “REGISTER-TYPE”. Lets rename this as “ARTICLE-TYPE”. Click on “REGISTER-TYPE” and when the menu comes up, select “Rename System”.

Coding Schemes can get quite complex. The scheme in Figure 3.4 is marginally more complex, and these schemes can grow to contain hundreds of choices. However, for now, the smaller the scheme, the quicker the coding.

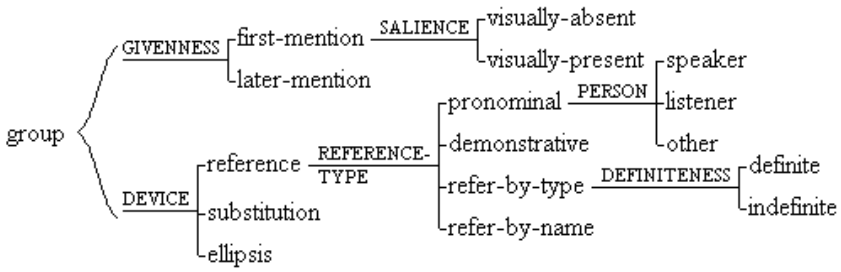


Figure 3.4: A more complex scheme

2.1 System networks

CorpusTool uses the hierarchy representation from *Systemic Functional Linguistics*. The hierarchy is called a *system network*. It consists of a number of inter-dependent choice points called *systems*. The system shown in Figure 3.2 is one such. Figure 3.4 presents 6 systems organised into a network.

A system consists of three parts:

- **system name**: the name of the choice. Typical names in grammar may be MOOD, POLARITY, FINITENESS, etc. System names should consist of a sequence of letters, and perhaps numbers and hyphens, but no spaces within the symbol. The system name needs to be unique -- CorpusTool will not allow you to provide the same name for two systems. The program will automatically display systems in upper case.
- **features**: the alternatives in the choice. In the above example, editorial and fpn are the features of the system. These are displayed in lower case, regardless of how you type them in. The same name cannot be used in two systems.
- **entry-condition**: each system has an entry condition, the feature (or complex of features) which forms the context in which the choice becomes relevant. In the above case, the entry-condition for the system is *register*, which happens to be the root-feature of the system network.

Several systems can have the same entry-condition, in which case, the systems are called *simultaneous systems*. They form a cross classification of the entry-condition. For instance, we might introduce another system with *register* as entry-condition, which might have features *finance, military, sport, etc.*.

The set of systems that you define form a *system network*, with the features of one system forming the entry conditions for more specific systems. How you can create these networks is described blow.

2.2 Creating & Modifying Systems

If you click on one of the feature (lower case) or the system (upper case) of the network, you will be presented with a popup menu of actions. These allow you to extend or modify the network.

2.2.1 Actions on Systems

- **Add Feature**: adds a new feature to the system.
- **Rename System**: allows you to change the name of the system.
- **Delete System**: deletes the system from the network. Note: the features which belong to the system, and any systems which depend upon them will also be deleted. And there is no undo at present. If any codings have been assigned these features, the features will be deleted from the codings.
- **Change Entry Condition**: change the entry condition of the system from one feature to another.
- **Move Up**: moves the system higher up in the graph, to reorganise the layout.
- **Move Down**: moves the system lower up in the graph, to reorganise the layout.

2.2.2 Actions on Features

- **Add System**: creates a dummy system under the feature.
- **Rename Feature**: changes the name of the feature.
- **Delete Feature**: deletes the feature from the system. Note: the any systems which depend upon this feature will also be deleted. And there is no undo at present. If any text has been annotated with the deleted features, the features will be deleted from the codings.

- **Move Up**: moves the feature higher up in the system. Note that the first feature in the system is the default in coding.
- **Move Down**: moves the feature higher up in the system.
- **Edit Realisations**: You can add realisations attached under features. The program doesn't do anything with them, but you can use this feature to annotate features, e.g., with a gloss of the choice.
- **Show Examples**: Once you have annotated some texts, selecting this option will open a Corpus Search window, showing all instances in the corpus tagged with this feature.

2.2.3 Changing Entry Conditions

To change the entry condition of a system, click on the system, and select "Change Entry Condition". You will be presented with a dialog box as in Figure 3.5.

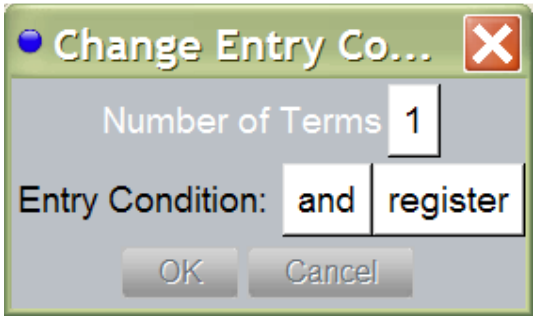


Figure 3.5: The Change Entry Condition Dialog

Simple Entry Condition: If you want a simple entry condition (the system extends from a single feature), then set "Number of terms" to 1, then choose the feature you want as the entry condition. Press "OK" and the graph will be redrawn as specified. The CorpusTool will automatically update the codings which are affected by the change.

Complex Entry Condition: You can also introduce complex entry conditions into your network. A complex entry condition involves a conjunction ('and') or disjunction ('or') of features. Set the number of terms to 2 or higher. Then select the features you wish as the input. Note that at present, you cannot mix AND and OR in the same entry condition. You can simulate such by first making a 'gate' (a system with only one feature). For instance, to construct the entry condition 'A and (B or C)', first make a system with one feature (call the feature b-or-c). Then use this feature and the A feature in an AND entry condition for your original system. The entry condition for this system will now be 'A and (B or C)'.

2.2.4 Moving a feature to another system

If you want to move a feature from one system to another, click on the system you wish to add the feature to, and select "Add Feature". Then type in the name of the feature you wish to move. The feature will be moved to this system. All codings will be adjusted for the change.

3 Adding "Glosses" to features

You can add a description of each of the coding features. Click on the feature and select "Add Gloss". This will open a window where you can type. Type a description of the feature, the criteria under which it could be selected. This gloss will then be visible while annotating a document (see below).

4 The Options menu

Every Scheme menu has a "Options" menu which allows you to do the following:

- **Save as**… : saves the scheme into a separate folder.
- **Show/Hide Glosses**: The "glosses" under features can be hidden or shown by selecting this option.
- **Show/Hide System Names**: The names of features can be hidden or shown by selecting this option. With system names hidden, editing of the scheme is more difficult (you usually click on the system name to access functions such as "add feature").
- **Save Diagram as PDF**: saves the network as currently displayed to a PDF file of your choice.
- **Save Diagram as SVG**: saves the presented network in SVG (Scalable Vector Graphics) format. See below for more on SVG.
- **Copy to Clipboard**: (*Windows only*): copies the graph as displayed to the clipboard. You can then paste into MS Word or other program. Note, for reasons only Microsoft understands, MS Word needs to be open when you copy to clip or else you cannot paste into MS Word.

5 Producing Images for inclusion in documents or the web

While the SVG format is not that widely supported, it is a great format for converting to other formats, since it stores the image geometrically, rather than as a bitmap.

To produce other formats from your SVG file, download and install *InkScape*. This software is free, and works on Windows, Macintosh and Linux. Download from: http://www.inkscape.org/

Open InkScape and select "open" from the File menu. Select your .svg file.

You can edit the file here if you wish.

To save in another format, there are two options:

1. Select "Save as" to save as PDF, EPS, EMF, or other vector-based file formats.
2. Select "Export Bitmap" to save as PNG format, which is a bitmap format which can be included in web pages or Word documents. The diagram in Figure 3.6 is a PNG file produced via InkScape:

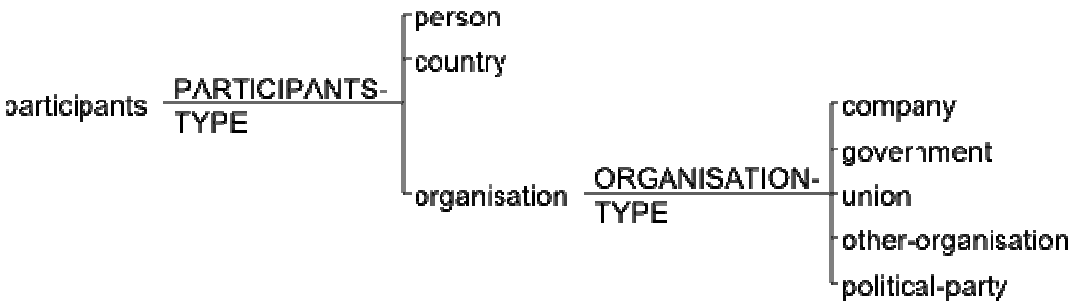


Figure 3.6: PNG file output

Section 4:

Annotating Files

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.

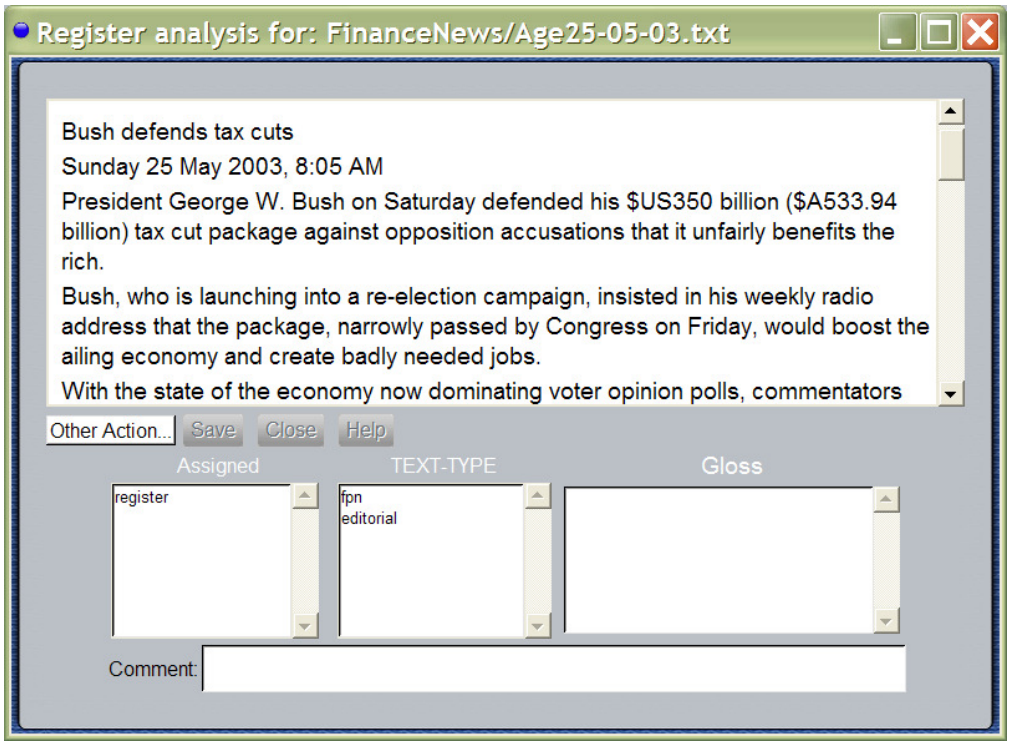


Figure 4.1: The Code-document window

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.

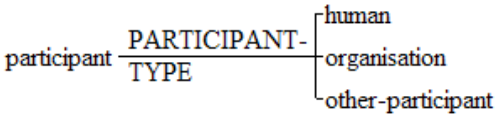
1. Bring the Project Window to the front
2. Click on the Add Layer button on the right of the screen
3. Call the layer "Participant"
4. Select "Annotate Segments"
5. Select "Do not automatically segment"
6. Select "Create New Scheme"
7. Press the "Finalise" button

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

Now, lets 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.

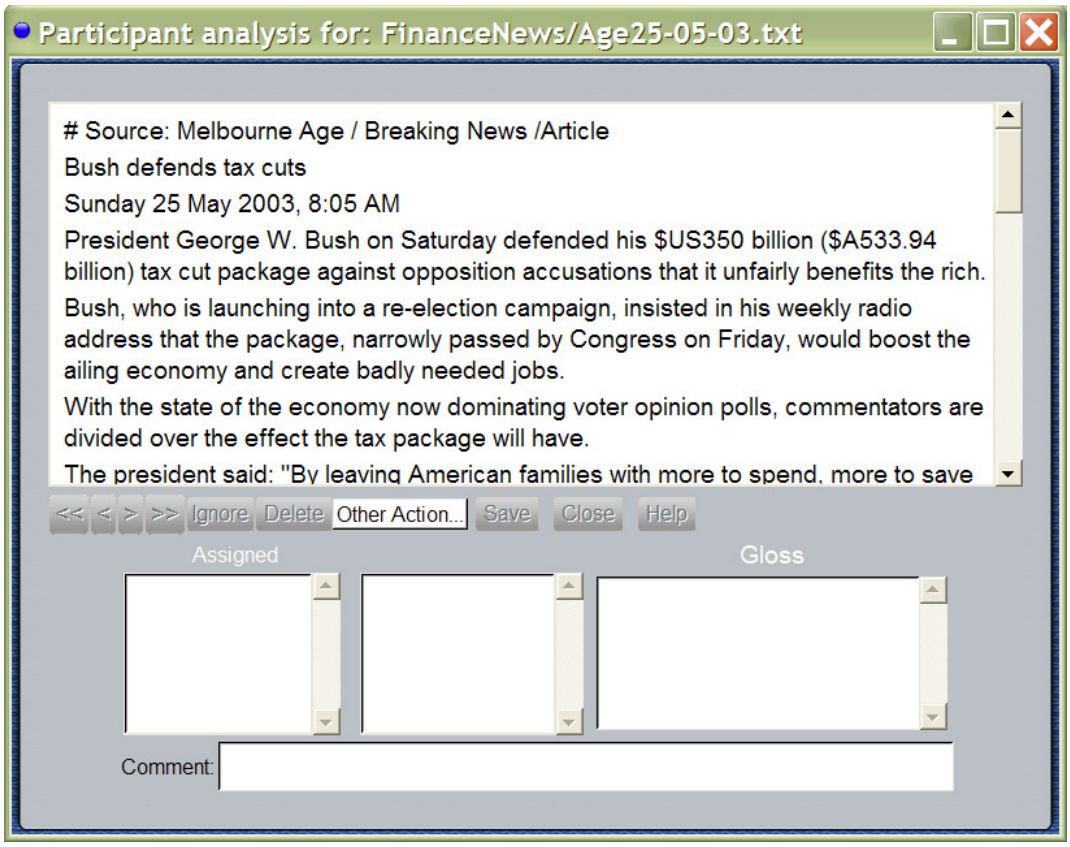


Figure 4.2: Code-segments window

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.

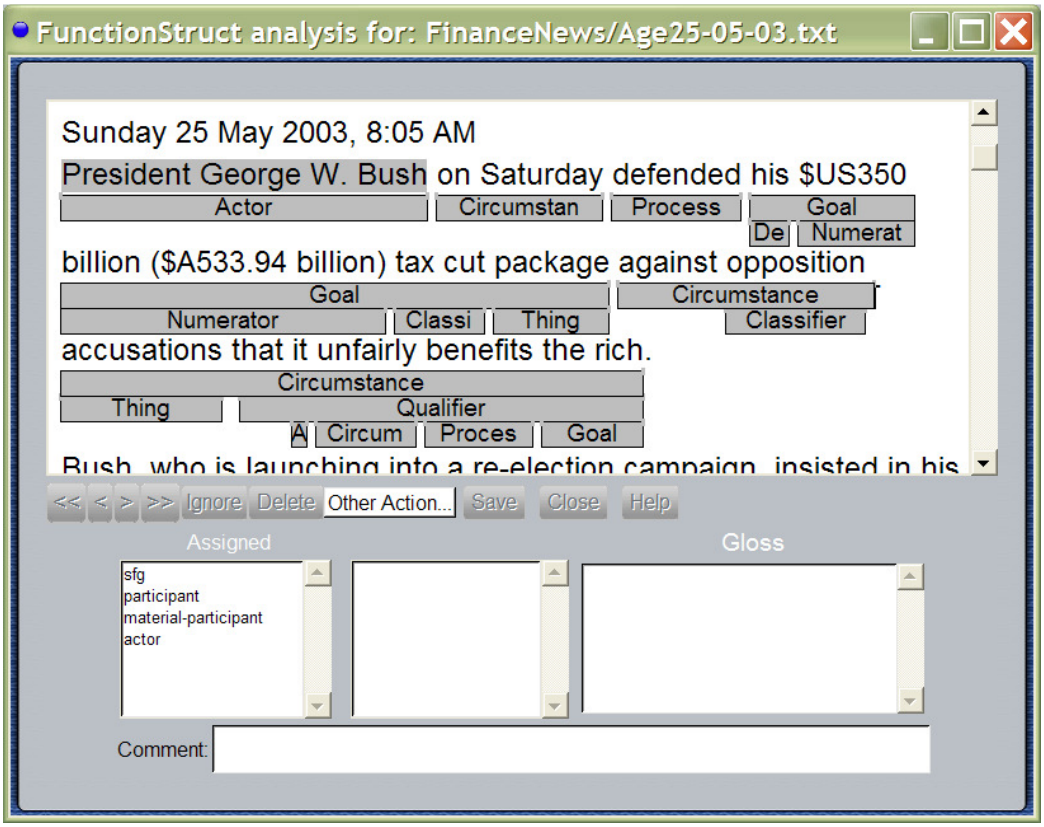


Figure 4.3: Function Structure Display Mode

- **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.

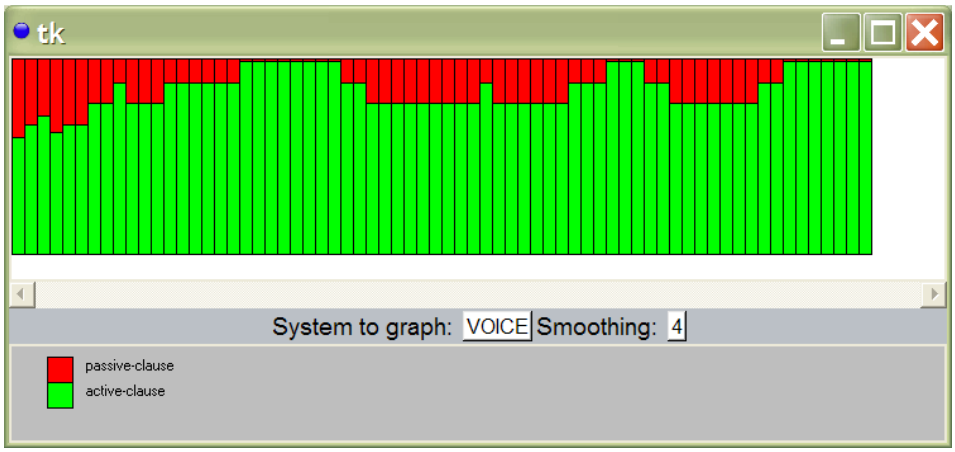


Figure 4.4: Text Stream Window

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.

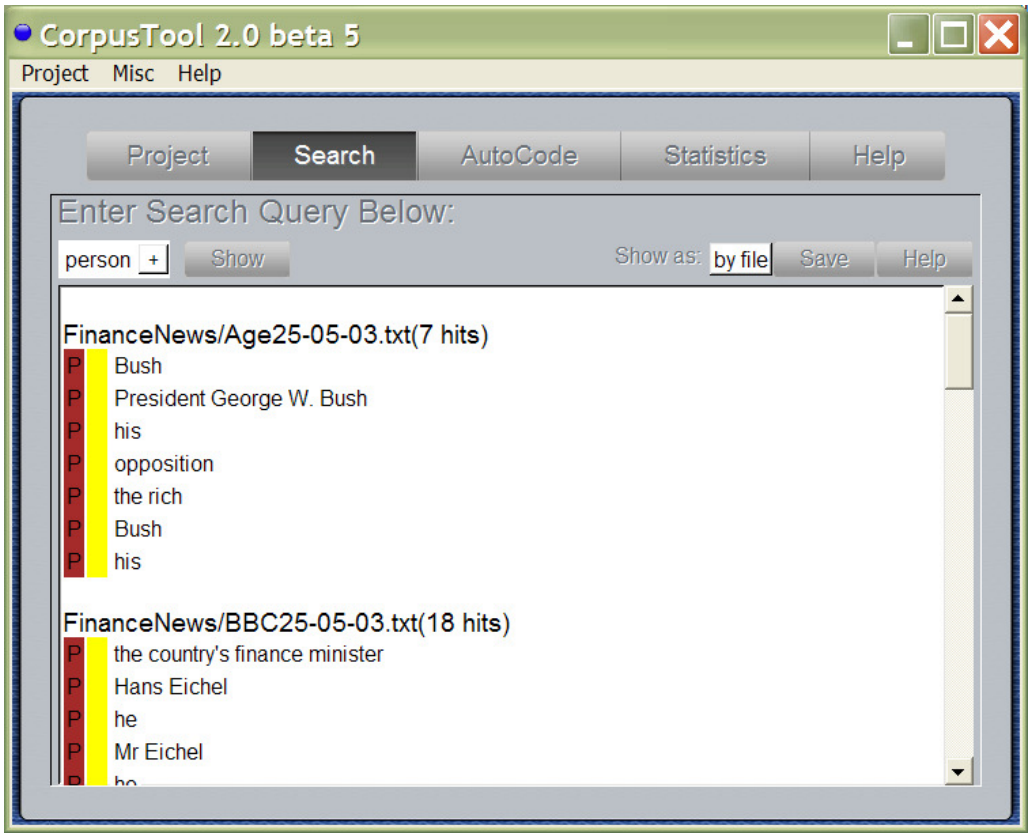


Figure 5.1: The Corpus Search Window

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:

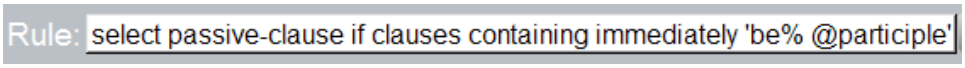
Automating Coding

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:

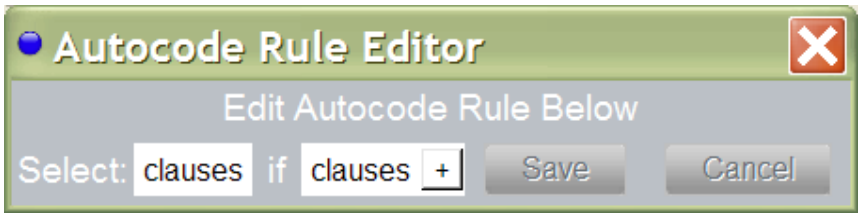


(**Note**: as with Search, lexical-based search patterns currently work for English)

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'.
   - *Nonconficting*: 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).

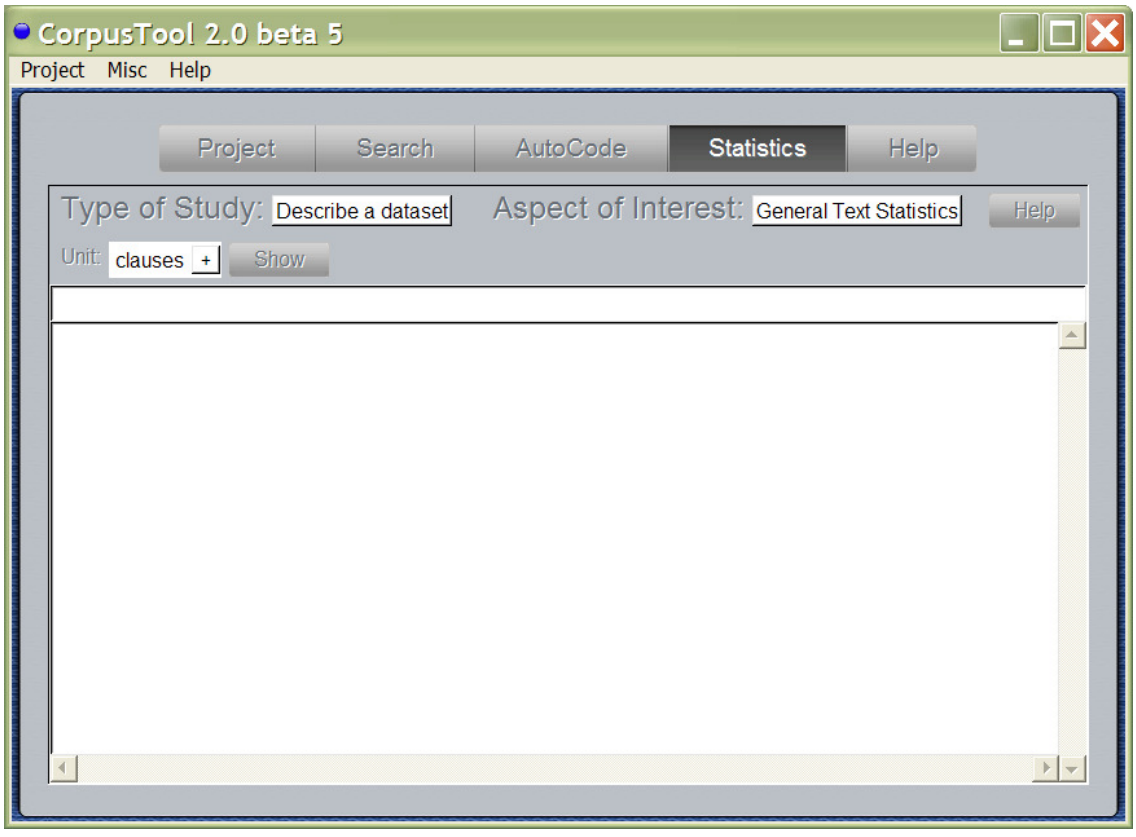


Figure 7.1: The Statistics Pane

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

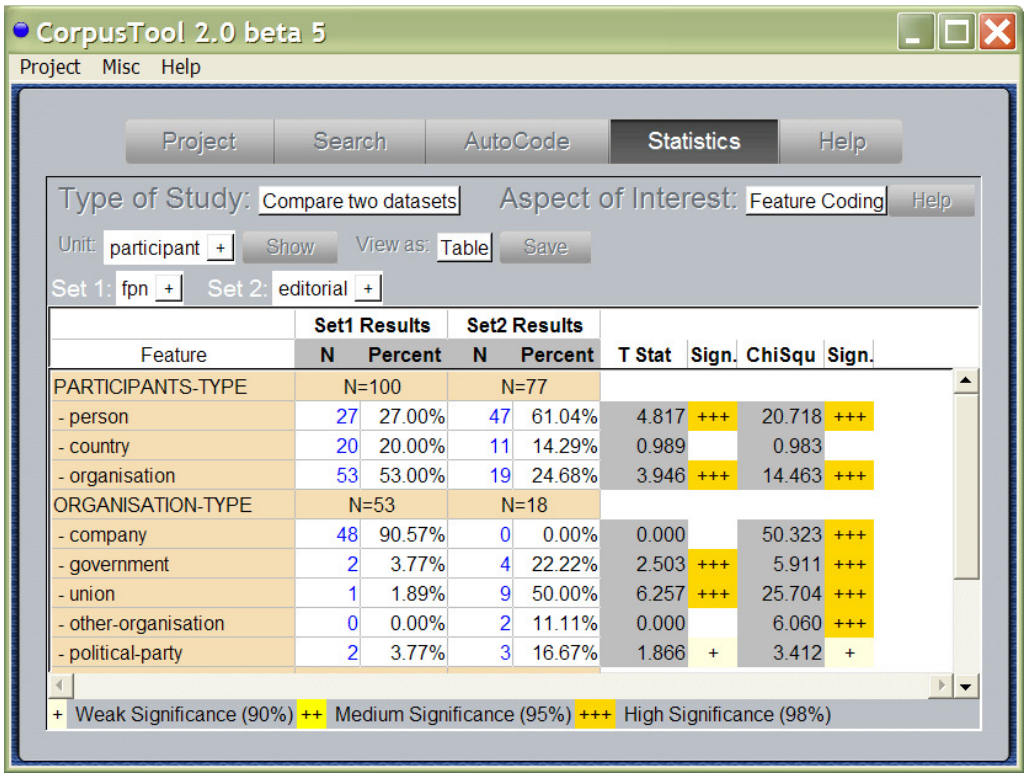
1. **General Text Statistics**: offers general statistics of the corpus, such as total number of segments, number of words per segment, lexical density in the corpus, pronominal usage, etc.
2. ***Feature usage***: you specify a feature in a layer (most typically, the root feature of the layer), and the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



Feature	Set1 Results N	Set1 Results Percent	Set2 Results N	Set2 Results Percent	T Stat	Sign.	ChiSqu	Sign.
PARTICIPANTS-TYPE	N=100		N=77					
- person	27	27.00%	47	61.04%	4.817	+++	20.718	+++
- country	20	20.00%	11	14.29%	0.989		0.983	
- organisation	53	53.00%	19	24.68%	3.946	+++	14.463	+++
ORGANISATION-TYPE	N=53		N=18					
- company	48	90.57%	0	0.00%	0.000		50.323	+++
- government	2	3.77%	4	22.22%	2.503	+++	5.911	+++
- union	1	1.89%	9	50.00%	6.257	+++	25.704	+++
- other-organisation	0	0.00%	2	11.11%	0.000		6.060	+++
- political-party	2	3.77%	3	16.67%	1.866	+	3.412	+

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, *fpn* 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 SysFan[1]. I thank Canzhong Wu, the author of SysFan, for allowing me to use this feature.

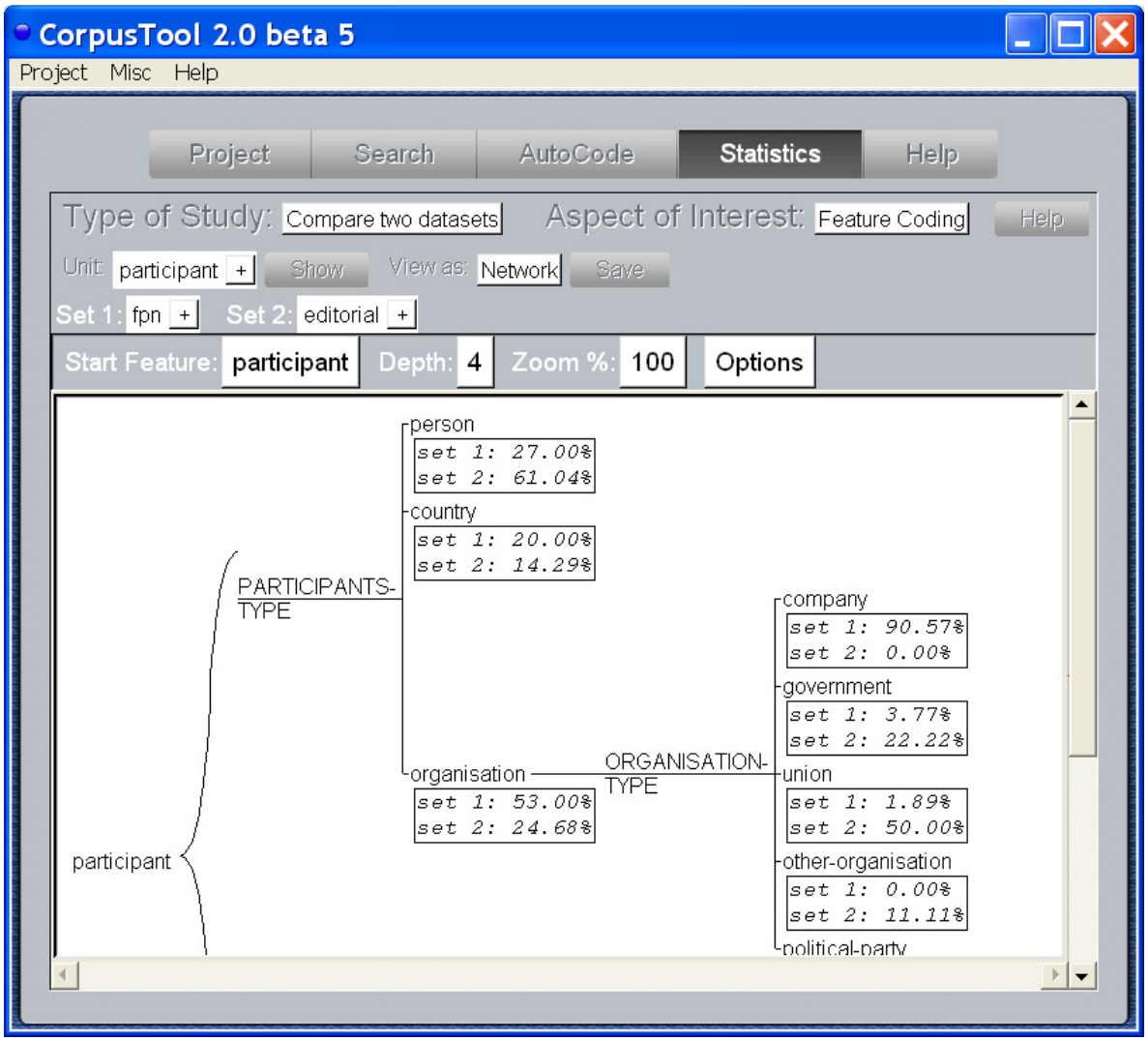


Figure 7.3: Network View of Statistics

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

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".

Section 9: Text Styling

3 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.

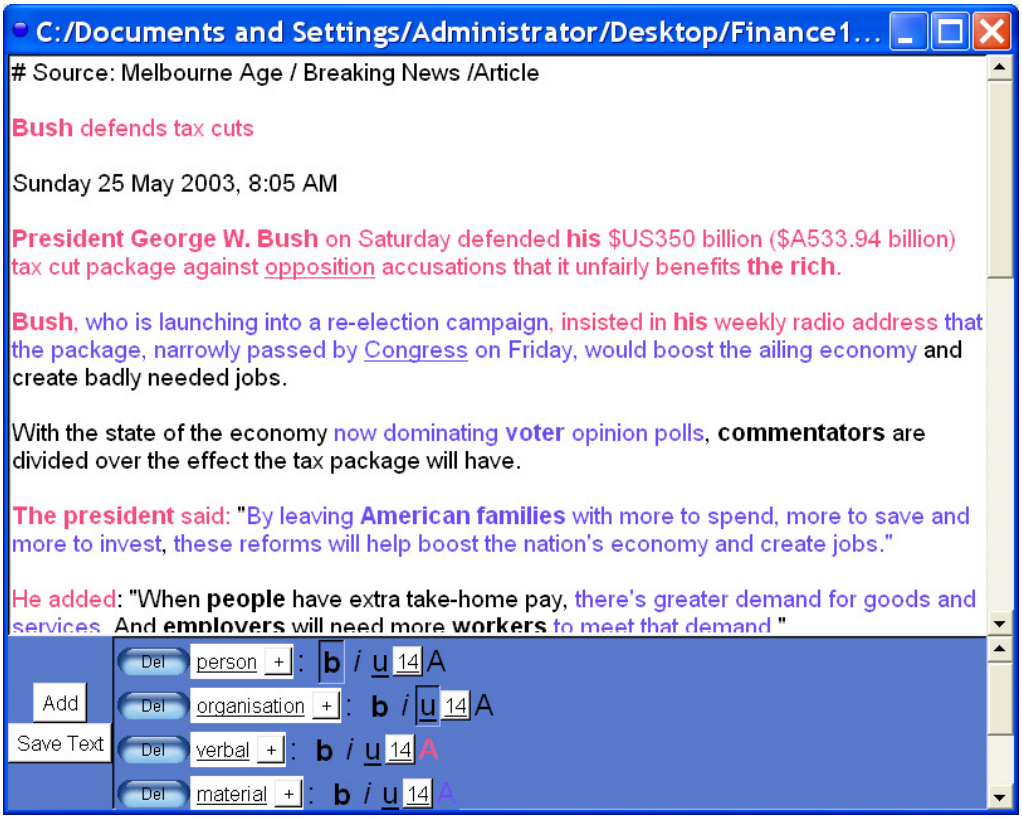


Figure 8.1: The Text Styler Window

4 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.

5 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.

6 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.

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.

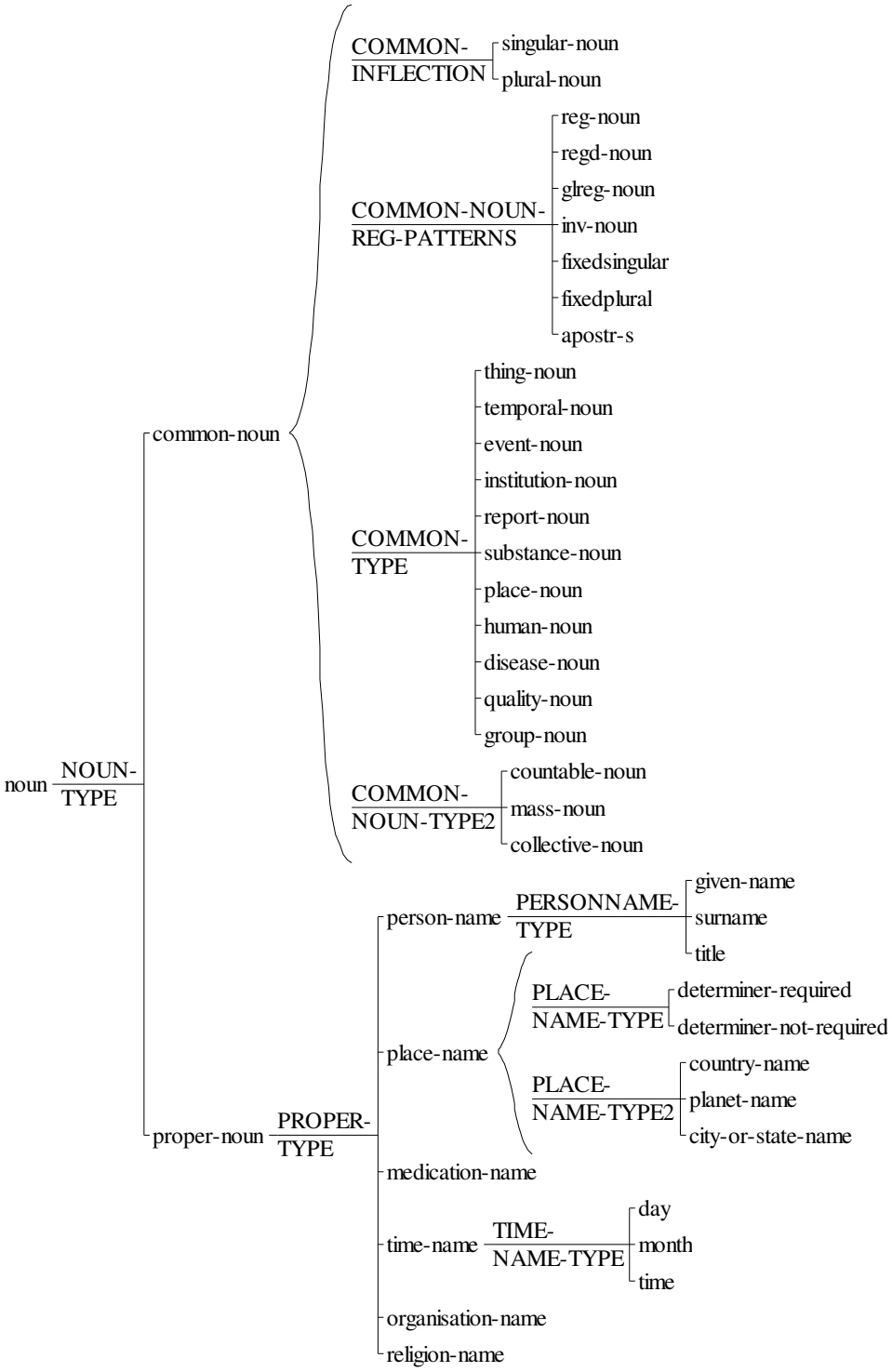
Some problems may arise:

- 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).

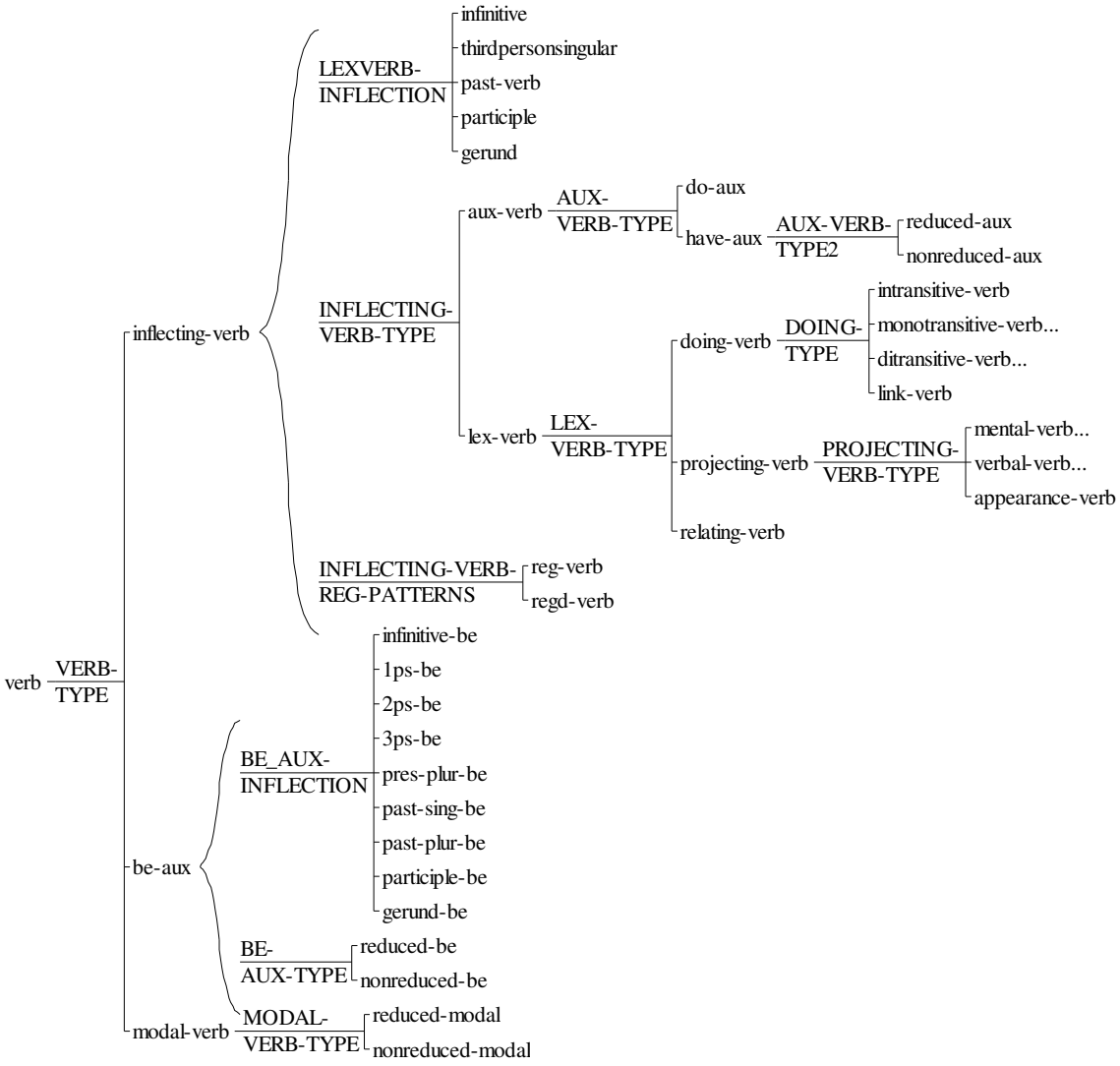
Appendix II:

Lexical Features for Concordance Searching

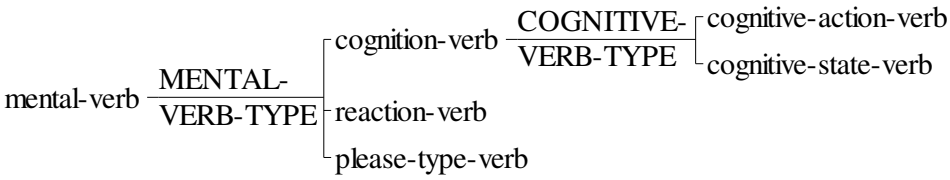
1 Nouns



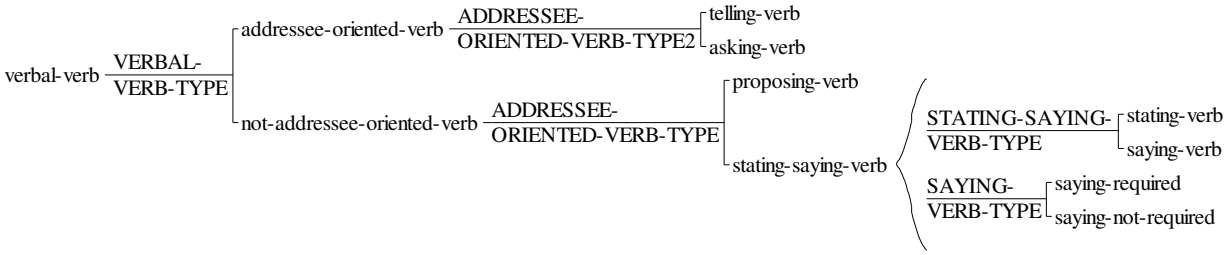
2 Verbs



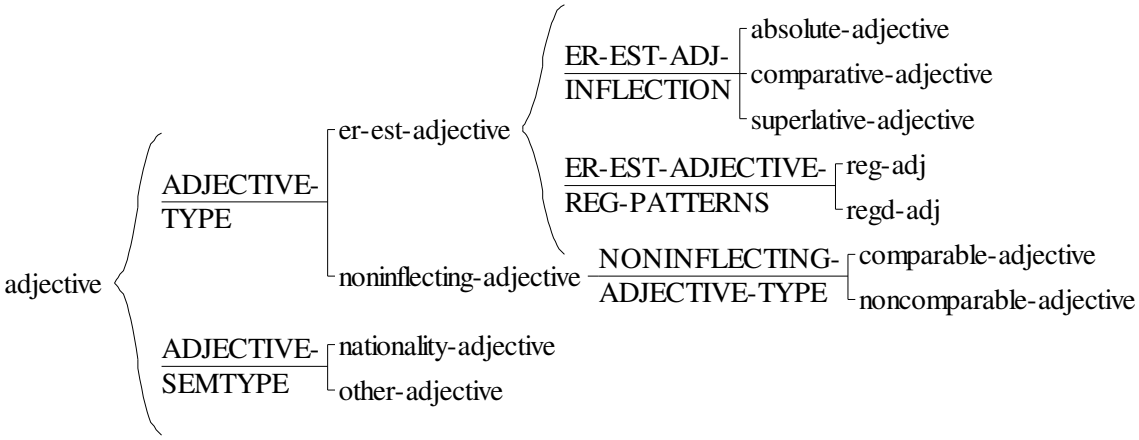
Subclasses of mental verb



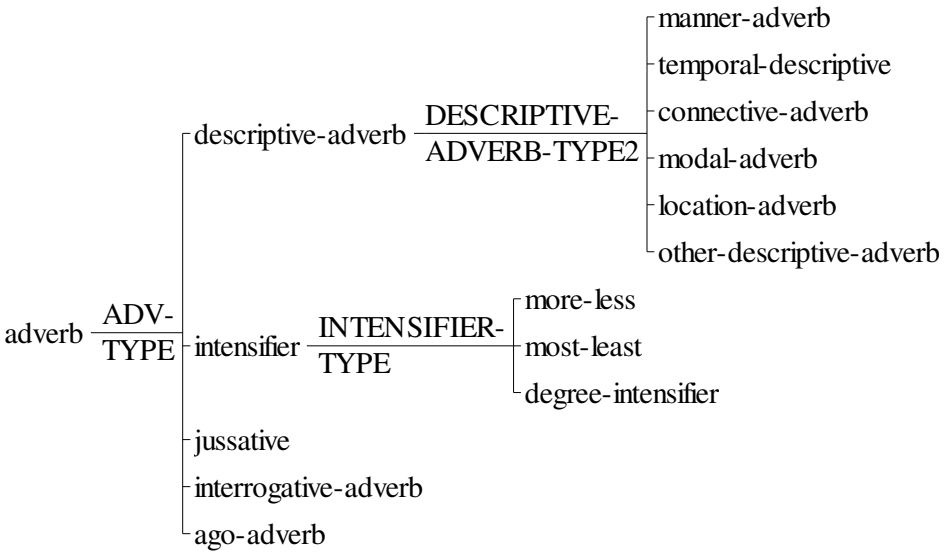
Subclasses of verbal verb



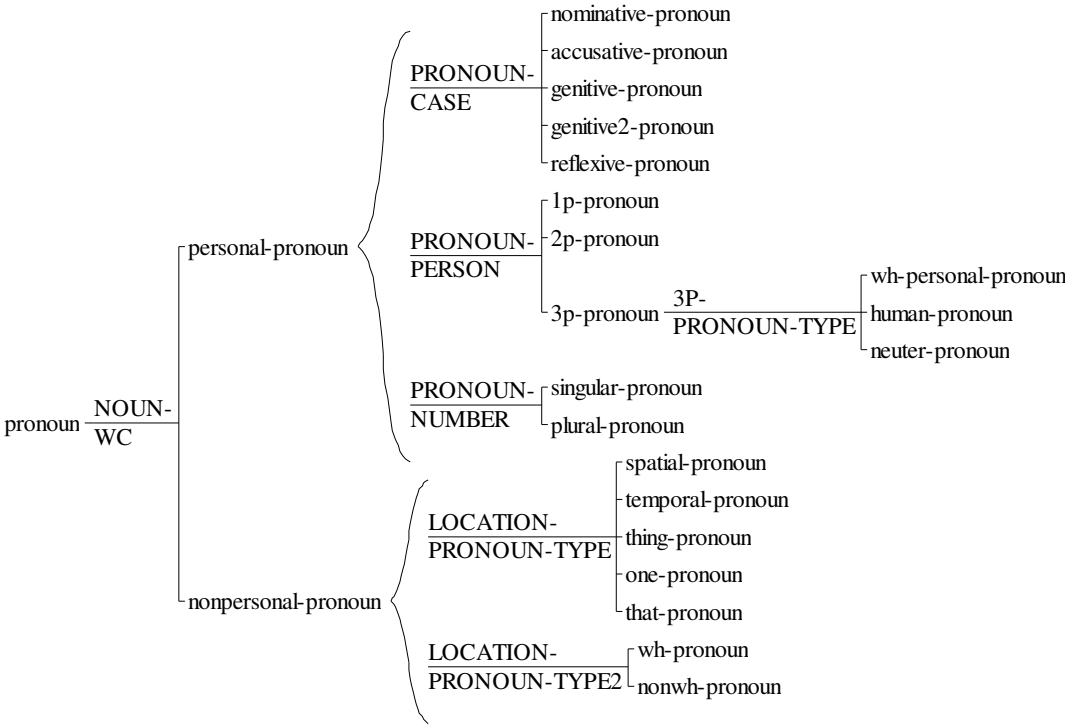
3 Adjectives



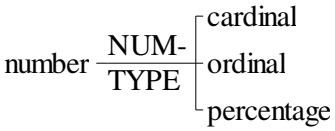
4 Adverbs



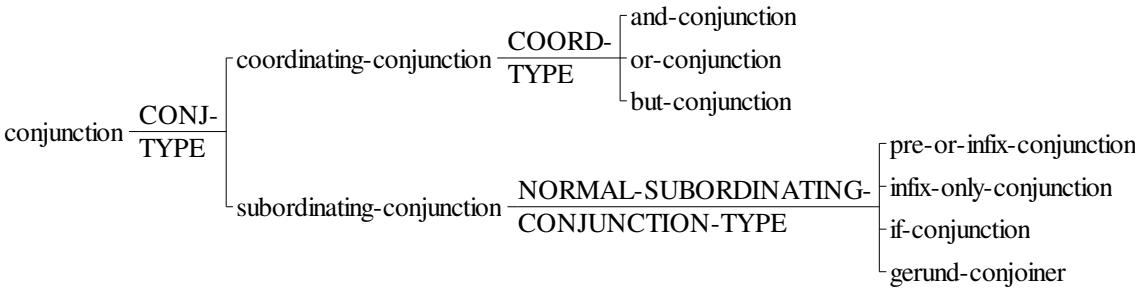
5 Pronouns



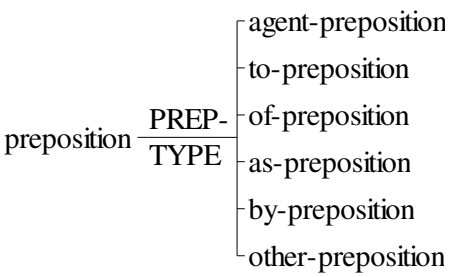
6 Number



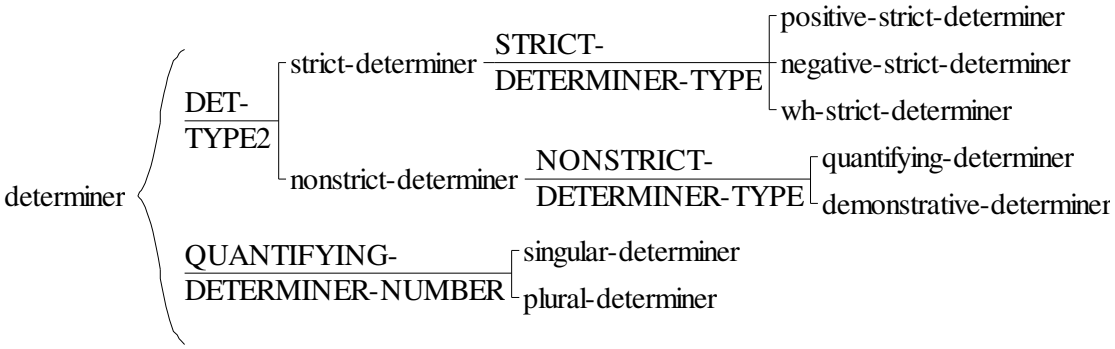
7 Conjunction



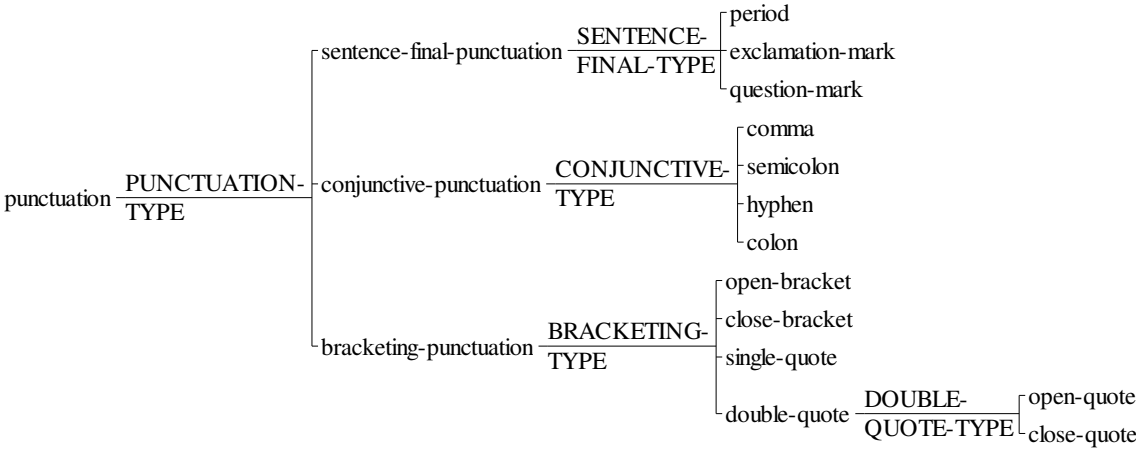
8 Prepositions



9 **Determiners**



10 **Punctuation**



genitive-s