

## Contents

<b>THE ICE PROJECT</b>	<b>2</b>
<b>THE DCPSE PROJECT</b>	<b>4</b>
<b>ICECUP 3.1</b>	<b>6</b>
<b>Acknowledgments</b>	<b>8</b>
<b>Introduction</b>	<b>9</b>
<b>PART 1: Getting Started</b>	<b>9</b>
1.1 Installing ICECUP and the corpus	9
1.2 The Corpus Map	11
<b>PART 2: Searching the Text</b>	<b>13</b>
2.1 Searching for one word	13
2.2 Concordancing your results	14
2.3 Displaying the wordclass tags	14
2.4 Viewing context	15
2.5 Searching for word sequences	16
2.6 Searching for word + tag combinations	16
2.7 Searching for alternative words	17
2.8 Searching with wild cards	17
<b>PART 3: Searching the Grammar</b>	<b>19</b>
3.1 The Syntactic Trees	19
3.2 Fuzzy Tree Fragments (FTFs)	19
3.3 Searching for a single node	20
3.4 Searching for two or more nodes	21
3.5 The FTF Focus	23
3.6 Adding words to an FTF	23
3.7 The FTF Wizard	24
3.8 The FTF Wizard (Version II)	26
3.9 Saving FTFs	27
<b>PART 4: Subcorpora</b>	<b>28</b>
4.1 Searching within a given subcorpus	28
4.2 Opening and browsing a subcorpus	28
4.3 Searching in an open subcorpus	29
4.4 Combining variables to build specialised subcorpora	30
<b>PART 5: Saving your Results</b>	<b>33</b>
5.1 Saving the results of a search	33
5.2 Saving Corpus Map tables and Lexicons	34
<b>PART 6: Advanced Features</b>	<b>35</b>
6.1 Drag and Drop Logic	35
6.2 Random Sampling	35
6.3 The Lexicon	36
6.4 The Grammaticon	37
<b>PART 7: Using Help</b>	<b>39</b>
<b>Further reading</b>	<b>39</b>
<b>References</b>	<b>40</b>

## THE ICE PROJECT

When Sidney Greenbaum conceptualized the creation of the International Corpus of English (ICE) in the late 1980s, he envisioned international teams of researchers collecting and computerizing similar types of speech and writing representing the national varieties of English that exist around the world, such as British English, American English and Indian English (Greenbaum 1988). Once computer corpora of these varieties were created, they would be fully tagged and parsed. The resultant corpora would enable not just the comparison of the various national varieties of English that have evolved around the world, but the linguistic analysis of one of the lengthiest and most extensively analyzed corpora of speech and writing ever created.

Unfortunately, Sidney did not live to see the completion of the International Corpus of English, but his dream to create computerized corpora of the many national varieties of English is currently being carried out by ICE teams in the following countries or regions:

Australia	Ireland
Canada	Malaysia
Caribbean (Jamaica)	New Zealand
East Africa (Kenya, Tanzania)	Philippines
Fiji	Singapore
Great Britain	South Africa
Hong Kong	Sri Lanka
India	USA

ICE-Great Britain (ICE-GB) was the first national component of ICE to be released. This component contains a million words of speech and writing that have been fully tagged and parsed, and was released simultaneously with ICECUP (The ICE Corpus Utility Program), a text analysis program that fully exploits the extensive grammatical annotation that ICE-GB contains. Taken together, ICECUP and ICE-GB provide the corpus linguistic community with a powerful resource for the analysis of present-day British English.

ICE-GB was designed along the same parameters that all ICE corpora were. Each ICE Corpus is divided into 2,000 word text samples representing various kinds of spoken and written English. It was decided to include in most cases text fragments rather than entire texts so that as many different speakers and writers as possible could be represented in the corpus. While it is true that 2000-word samples may make certain kinds of discourse studies difficult, corpora based on samples of this size, such as the Brown and Lancaster-Oslo-Bergen (LOB) corpora, have stood the test of time and shown that significant linguistic information can be obtained from such samples. Although many ICE teams continue to gather texts, the texts in ICE-GB were collected between 1990 and 1993.

Extensive discussion among ICE teams took place to determine what kinds of English should be represented in ICE corpora and how many samples from each type should be included. In the end, it was decided to include a range of different text-types, but a greater number of samples of those types that were more common in English. This decision is particularly reflected in the spoken part of the corpus, where nearly one-third of the samples consists of spontaneous dialogues. This is the most common type of speech that English speakers engage

in. Consequently, it is well represented in the corpus. Other types of speech that are represented in the spoken part of the corpus include radio broadcasts, telephone conversations, scripted speeches, and classroom dialogues (Nelson 1996a).

The written part of the corpus represents the broad spectrum of writing that exists, such as fiction, press reportage and editorials, and popular and learned writing. In addition, three types of writing not typically found in corpora are included: personal letters, business correspondence, and student essays and exams. Notably missing from the written part of the corpus is legal English, a highly specialized type of English that was excluded on the grounds that it represented a highly fossilized type of English intended mainly for a very specialized audience.

ICE-GB is annotated with three kinds of tags: structural tags, part-of-speech tags, and grammatical tags. Structural tags provide descriptions of the texts themselves. All texts contain file headers, which provide general descriptive information: a full bibliographic citation for written texts, for instance, and an identification of speakers in spoken texts. Each text is divided into text units, which correspond either to grammatical sentences or, in some instances in spoken texts, to coherent utterances. Some structural tags are peculiar to speech or writing. In spoken texts, structural tags identify speakers, mark segments of speech that overlap, and indicate two lengths of pauses. In written texts, structural tags mark paragraph boundaries and indicate font changes. Because the ICE project began in the late 1980s, the structural tags it uses do not reflect the standards recently proposed by the Text Encoding Initiative (TEI). Nevertheless, because the structural tags used in ICE-GB are SGML-conformant, they can easily be converted into TEI markup (Nelson 1996).

ICE-GB is fully tagged and parsed. For each text unit, a parse tree has been created, providing a visual representation of the part-of-speech of each word in the tree, the particular phrases and clauses that these words are members of, and the function that they serve (Subject, Object, etc.). Because corpora containing 2,000-word samples are best suited to grammatical analysis, it was decided that the part-of-speech tags and grammatical tags for ICE corpora would be quite detailed. The ICE tagset consists of 20 main wordclasses, while the parsing scheme has over 90 function and category labels. To provide this detailed an analysis, the TOSCA ICE Tagger and Parser were used (J. Aarts *et al.*, 1996).

A fully tagged and parsed corpus is only as useful as the tools that are used to analyze it. To analyze texts annotated with ICE tags, a special text analysis program, ICECUP, was developed. This program can perform tasks normally associated with text analysis programs: it can do simple string searches as well as generate KWIC (key word in context) concordances. But ICECUP performs other tasks specifically oriented towards corpora annotated with ICE-markup. For instance, ICECUP allows users to view or not to view ICE markup, enabling them to browse texts and view as much or as little markup as they desire. In addition, ICECUP can search for particular markup: someone wishing to study overlapping speech, for instance, can search for all strings in the spoken corpus containing overlaps. If in the analysis of overlapping speech, the user wished to search only sections of the spoken part of ICE-GB, the search could be narrowed to these sections. And if the user wished to study whether males overlap more than females, then the search could be narrowed to males or females.

ICECUP can also do a range of powerful searches. A user wishing to study coordination, for instance, can not only search for the individual coordinators, but search for actual tree structures containing coordination: a complex search can be devised to retrieve all instances of noun phrase coordination in the corpus. And an additional search mechanism allows for the retrieval of “fuzzy tree fragments” or FTFs, that is, parts of trees rather than entire nodes.

*Charles F. Meyer*

International Corpus of English Project Co-ordinator 1997-2001

## THE DCPSE PROJECT

Traditionally a distinction is made between diachronic and synchronic approaches to linguistics. The first considers language as it develops through time, whereas the latter takes a ‘snapshot’ look at languages viewed from the present. This old Saussurean dichotomy has recently been called into question, and some linguists have argued that the distinction is an artificial one. These linguists would argue that languages change all the time, even within the synchronic phases. As a result of these new attitudes to language development there is a new research impetus in linguistics which concerns itself with recent change (see Mair 1995, 1997; Mair and Hundt 1995, 1997, Denison 1998, Leech 2000, Smith and Leech 2001, Mair and Leech 2006).

For linguists who are interested in recent change corpora are especially valuable for data-gathering. At present they will need two separate corpora from two different periods. Naturally, these corpora must be comparable as regards their internal composition (i.e. sampling criteria). An example of work done in this area is Aarts and Aarts (2002) which investigates the use of the English relative pronoun *whom*. In order to compare data from two periods of Present-Day English (PDE) the authors looked at material from the *London-Lund Corpus* (LLC) and ICE-GB. They found that the overall use of *whom* as a Direct Object has become 90% less frequent over thirty years. Although ICE-GB is grammatically annotated and fully searchable, manual counts had to be carried out to find data in the older corpus. Thus, while the corpora were indispensable tools for this study, the research phase still required the careful pre-selection of comparable texts and manual searching of the LLC.

In order to support research into current change Professor Christian Mair at the University of Freiburg has constructed two corpora of 1990s English: FLOB (*Freiburg-Lancaster-Oslo-Bergen*) and FROWN (*Freiburg-Brown*). These corpora are intended to match the LOB (*Lancaster-Oslo-Bergen*) and *Brown* corpora containing written English from the 1960s. These are excellent resources enabling linguists to research changes in written English over 30 years. Manual searches are still unavoidable, however, as these corpora have not been parsed. We have taken Mair’s initiative further: we have constructed a corpus of British English comprising selections of spontaneous spoken English from the LLC and from ICE-GB. The new corpus will provide linguists interested in recent changes in English with a new and innovative database containing spoken English covering a period of 25-30 years.

We have opted for a corpus of spoken English because it is generally recognised that spoken language is primary and the first locus of changes in lexis and grammar.

The resulting resource, which we call the *Diachronic Corpus of Present-Day Spoken English (DCPSE)* will allow researchers to investigate changes in the grammar and usage of PDE over a period of 30 years. DCPSE differs from FLOB and FROWN in a number of important ways. Firstly, the corpus is unique in containing exclusively spontaneous spoken English. We provide a playback facility enabling linguists to listen to the original recordings. Secondly, the corpus is parsed which permits research into synchronic and diachronic grammatical variation. Thirdly, the corpus is fully searchable using the ICECUP software that we developed for ICE-GB, the basics of which are described in this volume. We envisage that DCPSE will be a major new resource complementing the Freiburg corpora, allowing access for the first time to recordings that could hitherto only be listened to at the Survey premises.

DCPSE contains spoken material from two corpora of Modern British English, both founded at the Survey of English Usage (SEU) at University College London: the *London-Lund Corpus*, compiled in the 1960s, and the *British Component of the International Corpus of English*, compiled in the 1990s. The London-Lund Corpus is the spoken part of the *Survey of English Usage Corpus*, founded by Randolph Quirk. It contains 510,576 words of 1960s spoken English (from which we have made a selection of 400,000 words). The corpus is divided into ‘texts’ of 5,000 words each which were transcribed and prosodically annotated (incorporating tone units, onsets, stresses etc.). Thirty-four were published in Svartvik and Quirk (1980). The corpus was computerised by Jan Svartvik (Svartvik 1990).

Many scholars have used the LLC for their research, resulting in hundreds of publications, principal among them Quirk et al. (1972, 1985). It is still one of the largest and most widely used corpora of spoken English, not least because it is the only English corpus that is prosodically annotated. Kennedy (1998: 32) stresses the importance of the LLC in its own right for the study of spoken British English, but also as “a very important baseline record of data...by which other corpora of spoken English can be evaluated... [The texts] have been used by researchers in many countries for studies which go well beyond the study of phonology. The detailed annotation has also facilitated numerous studies of lexis, grammar and especially discourse structure and function”. The SEU has enhanced the corpus by adding wordclass tags and parsing the corpus using the ICE-GB scheme. In addition, the SEU has digitised the original sound recordings which will be supplied – for the first time in the LLC’s history – with DCPSE.

In sum, DCPSE is a fully parsed and searchable diachronic corpus of 800,000 words of spontaneous spoken English, containing carefully selected and directly comparable texts from the LLC and ICE-GB corpora. In addition to the sociolinguistic variables in ICE-GB, two new variables are available. These are ‘awareness’ (whether the speaker was aware of being recorded) and ‘source corpus’ (ICE-GB and LLC).

This corpus is a unique resource for linguists studying the spoken English of a period spanning 25-30 years. There is currently no comparable resource available, and the corpus is the first of its kind enabling research into current change in spoken language.

*Bas Aarts*

Director, Survey of English Usage

## ICECUP 3.1

Welcome to the new version of ICECUP and the Second Edition of *Getting Started*. We hope you will benefit from enhancements to the software in this latest edition.

ICECUP is a corpus exploration program for parsed corpora like ICE-GB and DCPSE. Like its predecessor, ICECUP 3.1 uses Fuzzy Tree Fragments to build grammatical queries. You can still explore your results to devise new queries in a cyclic manner, and allow your own thought processes to evolve as you get to grips with the corpus and its annotation.

Version 3.1 is an evolutionary advance on ICECUP 3.0. In the main, facilities have been extended rather than replaced and the interface is as similar as possible. We trust that users of ICECUP 3.0 will immediately feel at home with the new software. Here are some highlights.

1. An integrated *lexicon* derived from the corpus
2. An integrated *grammaticon* of grammatical nodes
3. Simple *drag-and-drop statistics* in the lexicon, grammaticon and corpus map with the option to save statistical tables to disk
4. Enhanced *Fuzzy Tree Fragments*, with
  - a) sets of lexical *wild cards*
  - b) the ability to employ *logical expressions* in tree nodes
  - c) the option to include *structural features*
  - d) an improved user interface with a floating window to make editing nodes easier
5. An improved *FTF Creation Wizard*
6. The ability to *select sentences* manually (e.g., to limit searches to the sentences you want)
7. Improved browsing facilities including
  - a) *word wrapping*
  - b) *context options* (view sentence before/after; this context may also be saved to disk)
  - c) enhanced *grammatical concordancing* commands
  - d) integrated *speech playback* (with optional sound files)
8. General improvements to the user interface, such as
  - a) a new tree editor with zooming and panning using the mouse
  - b) new *quick-find* commands
  - c) faster and *parallel searching*

One result of publishing ICECUP and ICE-GB has been a steady stream of requests for changes to the software. Some of these requests have been minor, others less so. We recognise that we are providing software for the whole corpus research community, so we have taken all requests seriously.

The main extensions described in this booklet are outlined in the ICE-GB handbook, *Exploring Natural Language: Working with the British Component of the International Corpus of English* (Nelson, Wallis and Aarts, 2002; especially Chapter 7). Some extensions have been added since the handbook was written or that could not be anticipated in the book. However, all changes are described in on-line help provided on the CD. (For reference information on Fuzzy Tree Fragments see also [www.ucl.ac.uk/english-usage/resources/ftfs](http://www.ucl.ac.uk/english-usage/resources/ftfs)).

The Lexicon and Grammaticon (see Part 6) are entirely new to Version 3.1. These are large overviews of the corpus, like the Corpus Map (see Section 1.2 below), *entirely derived from the corpus*. Users can construct sub-lexicons at will, and structure the lexicon by defining a path, e.g. first subdivide by category, then by initial letter, and so forth.

In Version 3.1 all overviews, including the corpus map, now contain a table of statistics. Users can examine, for example, how a sociolinguistic subset subdivides the lexicon, by dropping a sociolinguistic query into the table, and in this way creating a new column of data for this intersection.

Go to **Corpus** | **Lexicon** and **Corpus** | **Grammaticon** to open these views. Experiment!

*Fuzzy Tree Fragments* (FTFs) have been extended significantly. The FTF (see Section 3.1) is a grammatical query, a structural model of what to look for in the corpus. FTF structures mirror trees in the corpus, and are therefore relatively simple and intuitive to understand. ICECUP 3.1 leaves the basic FTF idea intact but provides a series of powerful enhancements to search that you can use when you need to.

In Version 3.0 we could only look for a specific word within an FTF. Now a *wildcard* may be placed in any lexical position. So we can search for *\*ing* (any word ending in ‘ing’) in a Noun slot in an FTF. For more examples, see Section 2.7.

The icing on the cake is that we can also use a simple *set* notation to combine wild cards. So it is possible to list all the forms of the verb *work*, using the expression ‘{*work works working worked*}+<V>’ (where ‘<V>’ means the matching members of the set must be verbs). A similar mechanism may be used to exclude alternatives. Thus, the expression ‘{*\*ing ~thing*}+<N>’ retrieves all nouns ending in *ing* that are not *thing*. Although neither ICE-GB nor DCPSE is morphologically annotated, selective use of lexical wild cards and sets can, in many cases, achieve a similar result as searching for prefixes and suffixes.

The same principle applies to nodes in FTFs. Users can specify that a function or category is a member of a set. If we wish to specify that a node must be either a noun or a pronoun we put ‘{N, PRON}’ in a query. (Users can similarly state that the category must *not* be a member of a given set.) Occasionally, users may wish to express more complex expressions consisting of different node definitions, e.g. ‘SU and ~NP’ (a subject which is not a NP). ICECUP can now handle this as well. Most of the time you may not need this power, but it is there for when you do. A floating ‘edit node’ window makes editing complex expressions easier.

Finally, the *FTF Creation Wizard* (see Section 3.6) has also been enhanced. The default is the old ‘Version 1 Wizard’. This makes a new FTF by taking material from your current position in a corpus tree. But you can now also approach this question a different way. You can select nodes from the tree you wish to include in your FTF and then press the Wizard button to create it. This new Wizard is simpler, with options instructing ICECUP on how to treat the relationship between the different nodes you have selected.

Good luck, and enjoy!

*Sean Wallis*

ICECUP author and Senior Research Fellow, Survey of English Usage

## Acknowledgments

We gratefully acknowledge support for the initial part of the ICE-GB project from the Economic and Social Research Council (ESRC), under grant R000232077. ICECUP was initially funded by ESRC grant R000222598, and DCPSE under grant R000239643.

The TOSCA Research Group at the University of Nijmegen, under the directorship of Professor Jan Aarts, provided the tagging and parsing software used on ICE-GB. In particular, we wish to thank Dr. Nelleke Oostdijk and Dr. Hans van Halteren. Final stages of parsing were carried out using the Survey Parser developed by Alex Fang and supported by the Engineering and Physical Sciences Research Council (EPSRC), under grant GR/K75033.

We are indebted to the following people who worked on the annotation of ICE-GB at various stages: Celine Bijleveld, Judith Broadbent, Justin Buckley, Brian Davies, Ken Fletcher, Yanka Gavin, Marie Gibney, Howard Gregory, Jasper Holmes, Gunther Kaltenböck, Evelien Keizer, Ine Mortelmans, Yibin Ni, René Quinault, And Rosta, Oonagh Sayce, Laura Tollfree, Ian Warner, Jonathan White and Vlad Žegarac. For technical support, we thank Professor John Campbell, Tony Dodd, David Elkan, Isaac Hallegua, Neil Morgenstern, Richard Wilson, and especially Nick Porter and Akiva Quinn. For advice and assistance in collecting the corpus, we thank Brian Bennett, Dr. Mark Huckvale, Dr. Robert Ilson and Sue Peppe.

For their hard work in annotating DCSPE we thank: Dirk Bury, Amela Čamdžić, Yordanka Kostadinova-Kavalova, Lesley Kirk, Anne Law, Gabriel Ozón and Kate Scott.

We are, of course, wholly indebted to Randolph Quirk and Jan Svartvik and their teams in compiling, annotating and computerising the London-Lund Corpus, without which DCPSE would not have been possible. We hope DCPSE provides new perspectives on their corpus.

Many researchers have given us valuable feedback on ICECUP since the release of version 3.0 in 1998 by means of emails, visits to the Survey and discussions at conferences. Many people have contributed to the software you have in front of you. Particular thanks are due to Ilse Depraetere, Stefan Gries, Rolf Kreyer, Geoffrey Leech and Joybrato Mukherjee for their beta-testing and suggestions for ICECUP 3.1. Special thanks are due to Isaac Hallegua for his painstaking proof-reading.

The sources of individual texts in the corpus are acknowledged in the Corpus Map, which can be viewed in ICECUP. However, we particularly wish to thank the following organisations and institutions which made extensive contributions:

Audio Visual Centre, UCL	Independent Television
BBC Copyright & Artists' Rights Department	The House of Commons
British Museum Board	The Royal Courts of Justice, London
Careers Service, UCL	The Royal Society of Arts
Channel 4 Television	Staff Development & Training Unit, UCL
Faculty of Arts, UCL	UCL Students' Union
HMSO	

*Gerald Nelson, Sean Wallis and Bas Aarts*  
March 2006



## Introduction

*Getting Started* is a simple introduction to the corpus and the retrieval software, ICECUP. It shows you how to install the package, and provides a brief introduction to some of the basic features. It will enable you to conduct simple searches for words, grammatical tags, and syntactic labels, and to define a subcorpus. Finally, it shows you how to save your results. More detailed information, including tutorials, are provided in the extensive help manual.

## PART 1: Getting Started

### 1.1 Installing ICECUP and the corpus

The CD contains a program that will install ICECUP and the corpus for you. You can also use this program to start ICECUP directly from the CD.

- Place the CD in your CD-ROM drive.

If the install program does not appear immediately, open My Computer or Windows Explorer and open your CD drive.<sup>1</sup> The install program is called “Install.exe”.

- Launch “Install.exe” and you should get a window that looks like Figure 1 overleaf.
- If this does not work, contact your IT department to help you install the program. In the meantime you can run “Icecup31.exe” directly, without installing, even if your user account does not let you install programs in Windows.

The window in Figure 1 offers three options:

**Cancel** – which quits the installation without doing anything.

**Start!** – which starts ICECUP from the CD *without* installing.

**Install** – which installs the corpus and the software to your hard disk.

To install either corpus, you will need around 100Mb of free disk space on one drive. Of this, ICE-GB takes up 95Mb, while DCPSE requires about 85Mb. The ICECUP software is tiny by comparison (1.5Mb), although the Help file is around 2.5Mb.<sup>2</sup>

The install program places the ICECUP software and Help file in the directory location you specify. You can change this by clicking on the lower right panel or pressing <Alt> and ‘C’.

It installs the corpus in a subdirectory of this ICECUP directory. You can change this directory as you wish. Remember that the drive will need to have 100Mb of free space.

When you are happy with these locations, press **Install** to install ICECUP and the corpus.

<sup>1</sup> **Windows 3.1 compatibility.** ICECUP 3.1 is ‘backwards compatible’ even with Windows 3.1. However you will need to use *File Manager* to start the program. See the corresponding page to this one in the help manual for more information about running ICECUP with Windows 3.1.

<sup>2</sup> **Upgrades.** We strongly recommend that you install the new corpus in a different location to ICECUP 3.0, or the older program will no longer run properly.

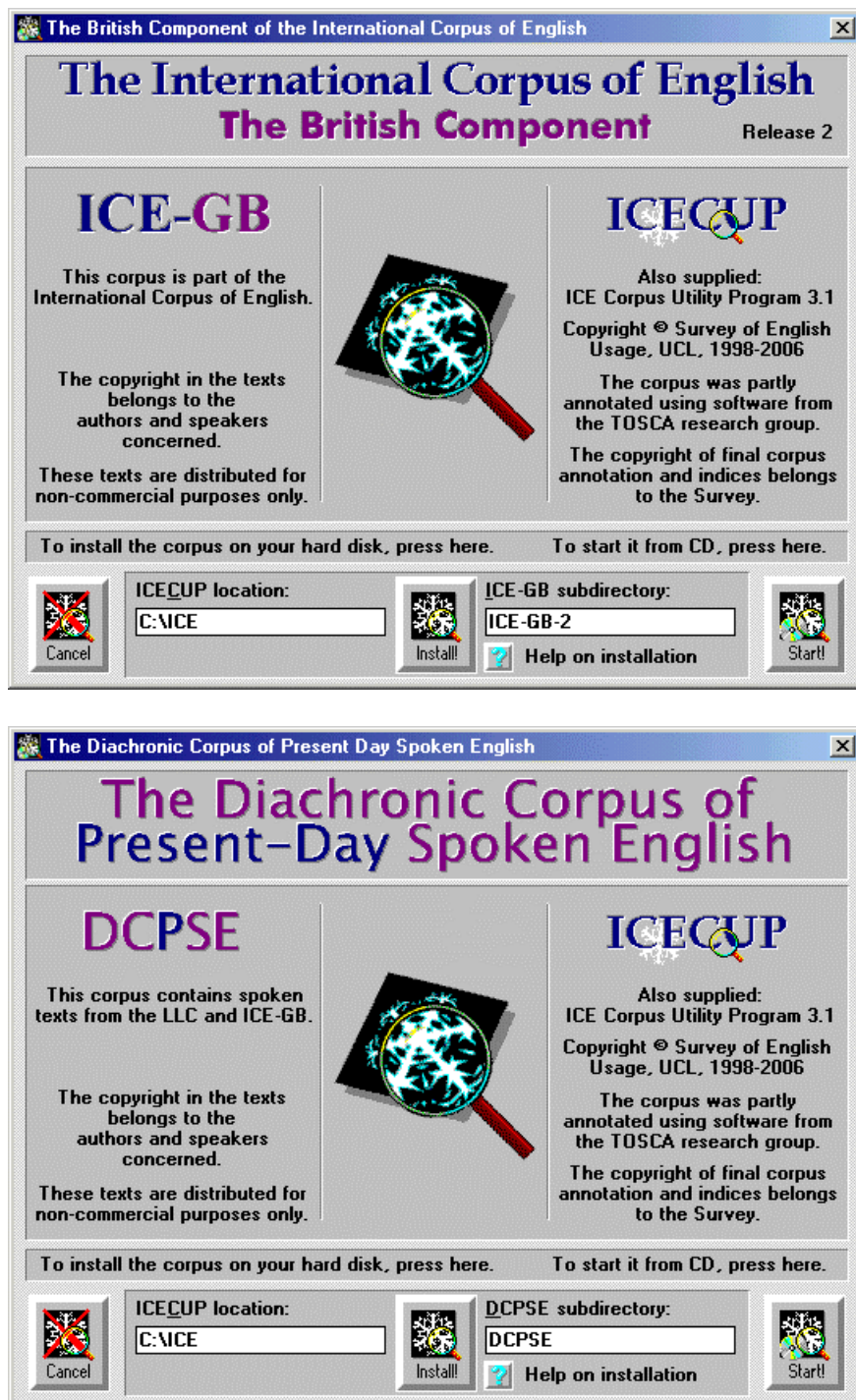


Figure 1: Installing ICE-GB (top) and DCPSE (bottom)

The figure shows the installation screens for ICE-GB and DCPSE. Instructions for installing sound recordings are provided with the CDs.

To strike some kind of balance, in this booklet we will demonstrate the software using both corpora. The same principles, if not identical query results, apply to both.

## 1.2 The Corpus Map



When you start ICECUP, the first thing you will see is the **Corpus Map**. This provides an overview of the corpus, according to a number of variables. The most important of these is the **Text Category** variable, which defines the hierarchy of text types in the corpus.

Part of the ICE-GB Corpus Map organised by Text Category is shown in Figure 2.

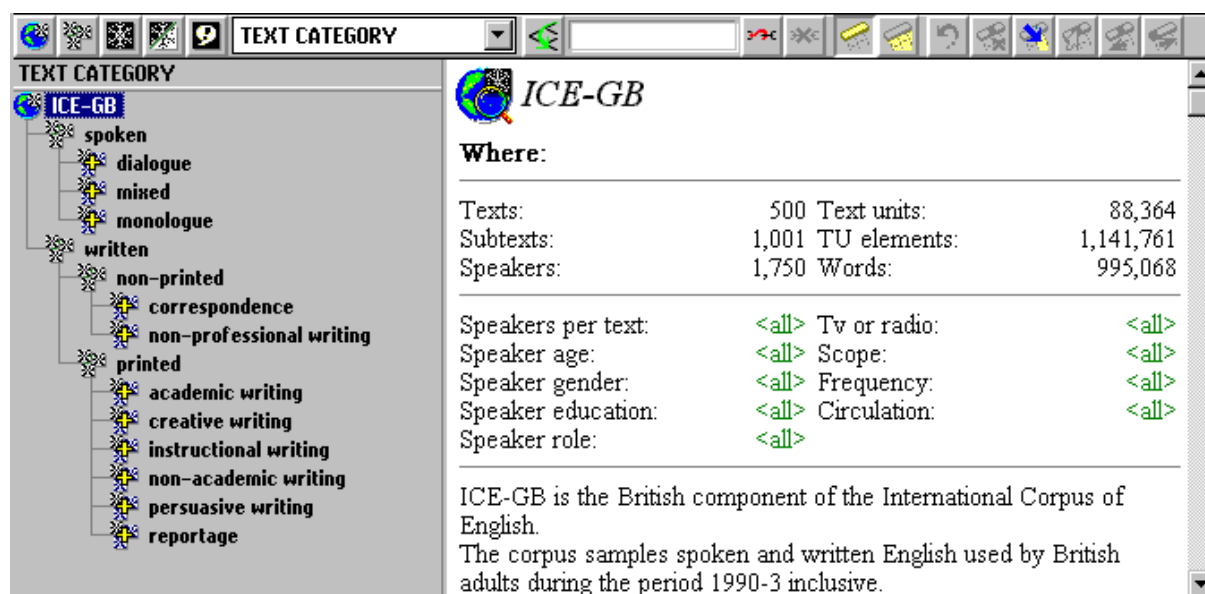


Figure 2: The Corpus Map showing the structure of ICE-GB

General information about the current selection is shown to the right of the map.






Just above the map, you'll see five expansion buttons, and the **Variable Selector**, shown in Figure 3.



Figure 3: Corpus Map Expansion Buttons and Variable Selector

By default Text Category is the selected variable, but you can choose other variables from the pull-down list (Figure 4) or hit <F2> to select it.

The expansion buttons allow you to expand and collapse the branches of the map in a number of ways summarised below:

-  collapses the entire map
-  shows the *value* structure of the current variable
-  shows all the individual *texts*
-  shows all *subtexts* (if a text is composite)
-  shows the individual *speakers* in texts and subtexts (i.e., it expands to show everything)

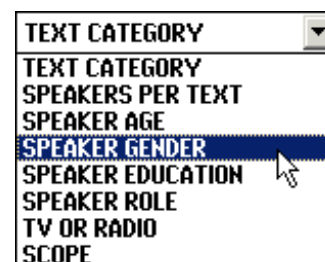









Figure 4: Variables in the Variable Selector

The iconic tree structure on the Corpus Map can be selectively expanded at any point by clicking on an element. Pressing <Ctrl> and <Right> together also opens a selected element. The Map can be expanded from the topmost level, 'ICE-GB', down to individual speakers, texts, and subtexts.

You will notice that as you change the currently selected element of the Map, information in the right hand panel changes accordingly. At the top of this panel are general statistics about each category and below is more descriptive information.

In the new version of ICECUP these general statistics can also be viewed in the form of a *table*. The buttons on the right of the bar let you view and modify the structure of this table.

-  shows the heading (visible by default)
-  reveals the table columns
-  deletes a column
-  insert a new column or edit an existing one
-  shift the current column to the left or right
-  undo – reverse the previous command

- Try clicking on the 'show table' button (''). The table will appear, pushing the panel further right. You can click on any cell in the table and the row and column will be highlighted.
- Now you can select a column, add or remove columns from this table and rearrange the columns using the buttons.

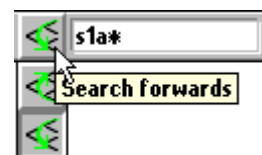
A new Quick Find option searches the corpus map from top to bottom, (or bottom to top if you prefer).

- Click in the white *find element* rectangle on the bar or hit <F3> and type 's\*' (anything starting with 'S') and hit <Enter>.

**Tip:** 'Drag and drop' also works with tables. Dropping query elements into the corpus map adds a new column to the table.

ICECUP 3.1 has two new tools that are similar to the Corpus Map. These are the Lexicon and the Grammaticon.

We will see what these do in Part 6.



**Figure 5:**  
Quick Find –  
direction button  
and 'find element'  
rectangle

## PART 2: Searching the Text

### 2.1 Searching for one word



Searches for single words are very fast in ICECUP, because these queries have been pre-calculated and stored.

To retrieve a single word:

- Click on the large **Text** button on the button bar. The **Text Fragment** dialog box appears (Figure 6).
- In the dialog box, type the word you are looking for, say **'play'**. Click on **OK**.



Figure 6: The Text Fragment dialog box

The search results now appear in a new window (Figure 7). The word *play* is highlighted. These results are from DCPSE. Each citation, or *text unit*, occupies a separate line in the view, differentiated by a *text code* reference and unit number. The word *play* is found 297 times in 287 different text units in DCPSE. This is summarised in the status line.

To see the number of hits per text unit, click on the small red 'Number of Matches' button ('1') in the bar.

Citations are displayed in *source order*. In DCPSE, citations from 'DI' (ICE-GB) texts precede 'DL' (LLC) texts. (In ICE-GB citations from 'S1A' texts, conversations, appear first, followed by those from 'S1B' texts, etc., and finally by written, 'W' texts.)

- Double-click on the total number of text units (287) on the status line to show full word-wrapped sentences.

**Tip:** DCPSE will also reveal original (ICE-GB or LLC) text codes in word-wrapping mode. To find a text by its original identifier type it into the **Query | Corpus Text...** window.

**Tip:** Part 4 shows how to search for *play* in a subcorpus.

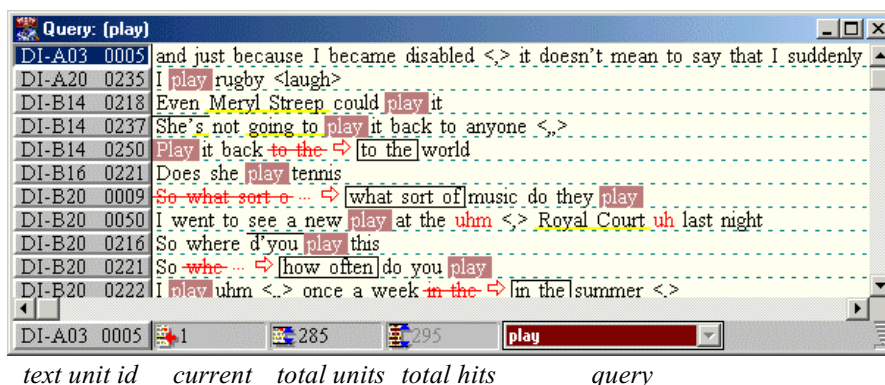


Figure 7: Results of a search for the word 'play' (in DCPSE)



## 2.2 Concordancing your results

The most convenient way to view these results is to concordance them, that is, to align each instance of *play* in the centre of the screen. To do this:

- Click once on the concordance button (☐). Alternatively, double-click with the mouse on the (grey) total number of hits (297) in the status line at the bottom of the screen.

The results now appear as shown in Figure 8.

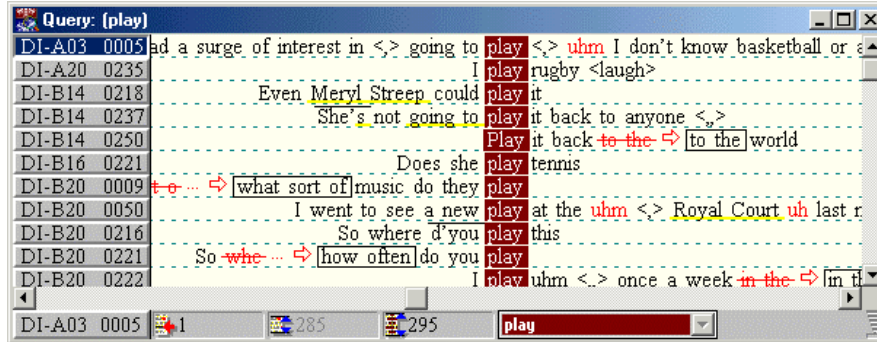


Figure 8: Concordanced results for the word 'play'

**NOTE:** To increase or decrease the size of the displayed font, use the Change Scale button (☐). Press down with the left mouse button to zoom in, right to zoom out.

The mouse can be used to zoom and pan the display as well. Hold down the <Ctrl> key and drag the mouse in the window to zoom in or out. You can scroll the window gently by clicking and dragging the view.

There are several ways in which ICECUP allows you to explore these results. Here we will look briefly at just two of them.

## 2.3 Displaying the wordclass tags

To see the wordclass tags assigned to *play*:

- Click on the small 'C' (display category information) button on the button bar.

The symbols N (noun) or V (verb) now appear after each entry (Figure 9):

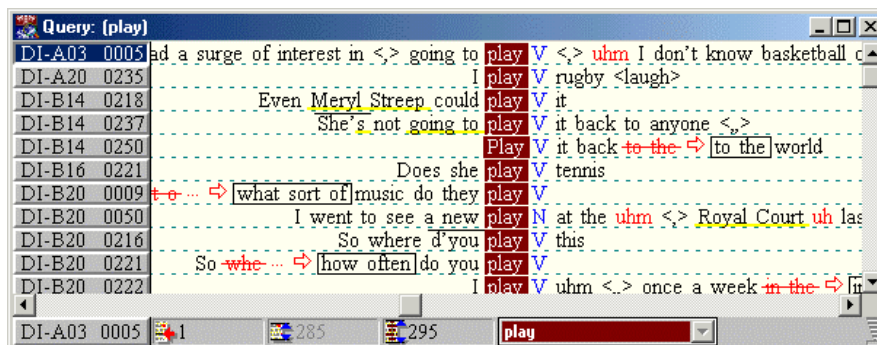



Figure 9: Concordanced view showing category information

Most ICE tags also contain additional information – what we call *features* of the tag. To display these features click on the small ‘’ (display features) button.

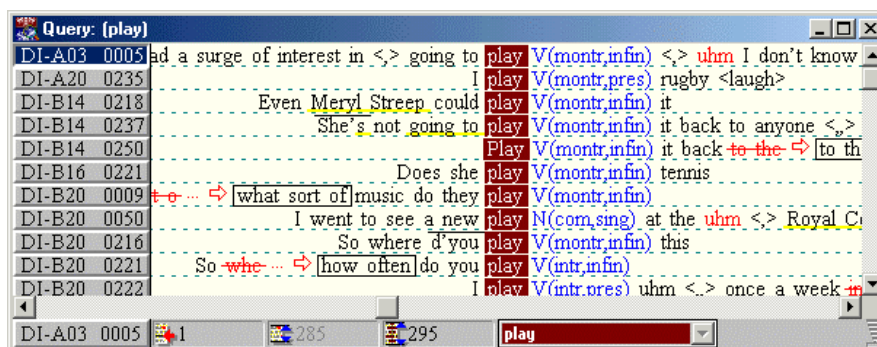


Figure 10: Concordanced view showing category and feature information


The features of each tag are now displayed in addition to the category information.

Noun features denote type (common or proper) and number (singular or plural). The verb features include the transitivity and the form. The fourth line in Figure 10 shows, for instance, that the word *play* is marked as a monotransitive (montr) infinitive (infin) verb in the sentence *She's not going to play it back to anyone* (DI-B14 #0237).

**Tip:** Place the mouse over the grammatical tag to view a popup summarising what it means. The full ICE Tagset documentation can be viewed in the on-line **Help**.

## 2.4 Viewing context

By default, ICECUP displays only the immediate text unit in which your search argument (in this case, *play*) occurs. There are several ways to display more context.

1. Clicking on the ‘’ (browse context) button opens the subtext in which the element occurs. The entire text from which the text unit comes appears in a new window.
2. Alternatively, press down with the right mouse button in the window (not the margin). A pop-up provides a number of options, including ‘Browse context’, ‘Browse context & Query’ and ‘Browse corpus map’. Clicking on the last option opens the corpus map for the current text unit location. Browsing the context with the query opens a window where the search element is restricted to the single subtext (Figure 11).

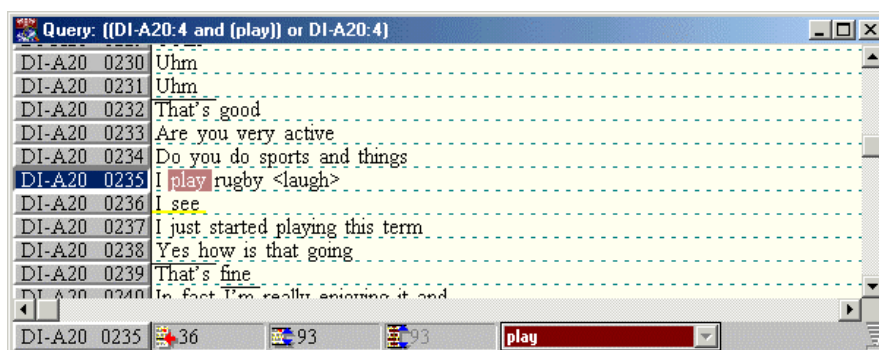




Figure 11: Viewing context: DI-A20:4 with query ‘play’ marked

In both of the above methods, the context is shown in a new window and the current text unit is highlighted. This allows you to read the immediately preceding and following context. The entire text can be scrolled using scroll bars or dragging.

- Another way to see context is to insert it before or after each sentence in the view. Click on the ‘’ (sentence before) or ‘’ (sentence after) buttons to expand this context.

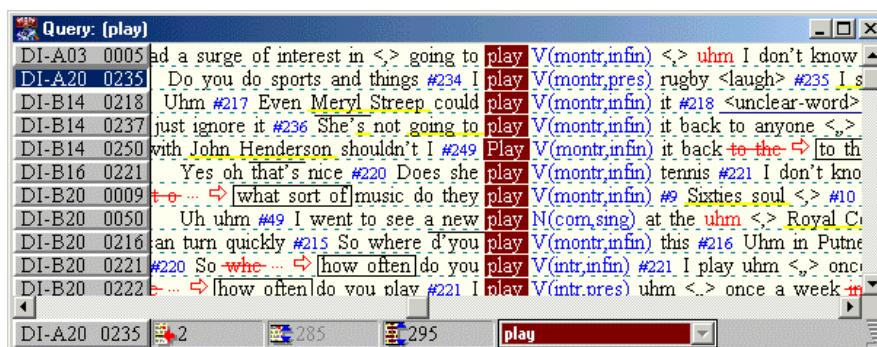




Figure 12: Viewing a concordance with surrounding sentences – note the unit numbers in the text

## 2.5 Searching for word sequences

The **Text** dialog box that we used to search for the word *play* can also be used to launch searches for sequences of words, including discontinuous strings.

To search for a continuous string, simply type the string into the Text Fragment box with spaces between each word, and click **OK**.


Two options are available for discontinuous strings:

-  **1 missing** (**<Alt>+‘1’**) This inserts a light ‘?’ in your query which means “any single word”. So “**would ? be**” will retrieve *would not be*, *would never be*, etc.
-  **some missing** (**<Alt>+‘M’**) This inserts a light, non-bold ‘\*’ in your query, meaning “any number of words, including zero words”. Thus “**would \* be**” will retrieve *would be*, *would not be*, *would perhaps never be*, etc.

**NOTE:** By default, all lexical searches are *case insensitive*. So typing ‘**play**’ will retrieve *play*, *Play*, and *PLAY*. To turn this off, select **Options** and untick ‘Disregard CAPITALisation’.

## 2.6 Searching for word + tag combinations

The **Text Fragment** dialog box can also be used to search for a word plus a specified grammatical tag. In this example, we will specify that we want all instances of *play* as a verb, disregarding those that are tagged as nouns.

- Click on the large **Text** button on the button bar. In the **Text Fragment** box which appears, type ‘**play**’. Now click on the ‘’ (node) button. This inserts a ‘+’ sign and a pair of angled brackets after *play*. Your query should now look like ‘**play+<>**’.
- Using the mouse, insert the cursor between the angled brackets, and type ‘V’. Your query should now look like this: ‘**play+<V>**’.




- Click on **OK**. In DCPSE this retrieves 165 instances of *play* as a verb.

**NOTE:** If you delete the '+' sign, ICECUP will look for *play* followed by a verb, '**play <V>**'.

## 2.7 Searching for alternative words

ICECUP 3.1 lets you specify a query containing a number of possible words in a given position. There are three basic ways of doing this which work together.

- Sets of words listing alternatives, e.g. '{**play plays playing played**}'
- A wild card defining a matching pattern, e.g. '**play\***'.
- Employing negation using the swung dash, e.g. '**~play\***'.

These methods can be used together. '**{play\* work\*}**' will find words starting with either *play* or *work*. Sets can contain exclusions, permitting '**{play\* ~player}**' to mean any word beginning with *play* except *player*. To insert a set of words or wild cards into your query, click on the 'set' button (  ) or press <Alt> and 'S' together and then type.

## 2.8 Searching with wild cards

Wild cards match words to a pattern of symbols. These symbols include regular characters, such as letters and digits, and some special symbols. The two simplest 'special symbols' are '\*' and '?', meaning 'any characters' and 'any single character' respectively.

- In the **Text Fragment** box, type '**pl\*y**' and press **OK**.

In DCPSE this matches *plainly*, *play*, *pleasantly*, *plenary*, *plenty*, *pleurisy* and *ploy*.

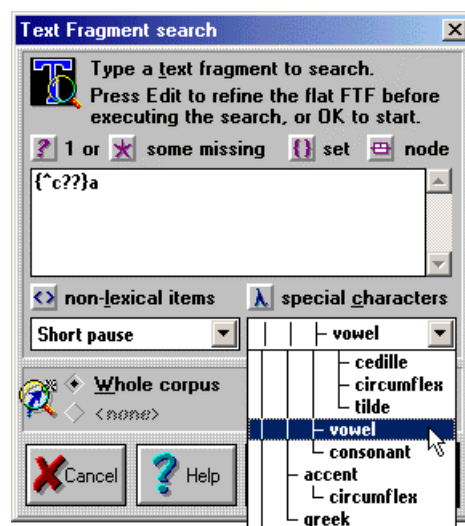
A common use of wild cards is at the start or end of an element. Thus '**p\***' means any word beginning with P. The typed '\*' here means some missing *characters*, not words.

Question marks ('?') stand for a single unstated character. So '**pl?y**' will match *play* and *ploy*.

But often you want to be more specific than this. To express 'any of the following characters', you can specify a set of characters within curly brackets.

- In the box, type '**pl{aeiou}y**' and press **OK**.

**Tip:** To specify several ranges of characters, use dashes to specify each range, and commas to separate them, e.g., '**{a-z,0-9}**' means any alphanumeric character.



**Figure 13: A Text Fragment query for a wild card**

**Notes:** This query will search for two consonants followed by the letter 'a'. In the bottom right the set of lower case vowels ('^v') is being selected.

	description	examples	matches
*	Any number of characters, zero or more	a* *ing b*ing	A word starting with “a”; one ending with “ing”; one starting with “b” and ending with “ing”.
?	Any single character	a??? b?c?u?e	A four-letter word starting with “a”; a seven-letter word with “b”, “c”, “u”, and “e” in odd positions.
{ }	User set.	w{0123} t{a-z??}	“w” followed by 0,1,2, or 3; “t” followed by two letters.
^	Escape. The next character is either: 1: a member of a set 2: literal	b^vd be^c^v  ^? ^* ^{ ^. ^& ^^	A three letter word “b”, vowel, “d”; “be” followed by a consonant and then a vowel. (See Table 2.)  A literal question mark, <i>etc.</i>

Table 1: The four basic components of a lexical wild card (after Nelson *et al.*, 2002)

**Tip:** To make the set apply to more than one character, add a wild card scope mark (‘\*’ or ‘?’) at the end of the set. Thus ‘{a-z??}’ means two letters (Figure 13).

Table 1 summarises the four basic components of a lexical wild card.

The fourth of these, *Escape*, defines a number of previously defined *character subsets*. Table 2 below lists some of these. A full list is given in the on-line help.

Predefined sets are specified by an ‘escape’ (‘^’) character followed by a single character code, for example, ‘^v’ means *any lower case vowel*. It is a quick way of writing ‘{aeiou}’.

Rather than have to look them up, you can insert many pre-defined sets, including ‘^v’, using the **special characters** pull-down control shown in Figure 13.

**Tip:** To make pre-defined sets apply to more than one character, just bracket them inside another set. Thus ‘{^v??}’ finds *oo*, *au*, *ie*, and any other two-vowel fragment.

More information on wild cards, including tips and tricks, and the full list of pre-defined sets, is published in (Nelson, Wallis and Aarts 2002) and in the main on-line help file (<F1>).

symbol	description	explanation
^@	non-alphabetic	Any character not present in the Western alphabet.
^#	digit	0 to 9.
^L, ^I	Letter, letter	Any letter character, Greek and Western.
^A, ^a	alphabetic	Any letter from the Western alphabet.
^V, ^v	vowel	A vowel, a, e, i, o or u.
^C, ^c	consonant	Any English consonant.
^G, ^g	greek	Any letter from the Greek alphabet.

Table 2: Some common predefined sets of characters (case and accent sensitivity settings apply)

## PART 3: Searching the Grammar

### 3.1 The Syntactic Trees

Every text unit ('sentence') in ICE-GB and DCPSE has been syntactically analysed, and the analyses are shown in the form of syntactic trees. To display the tree for any text unit in the corpus, simply double-click on it in any browsing window. Figure 14 shows the tree for DI-B16 (S1A-020 in ICE-GB) #221, *Does she play tennis?*

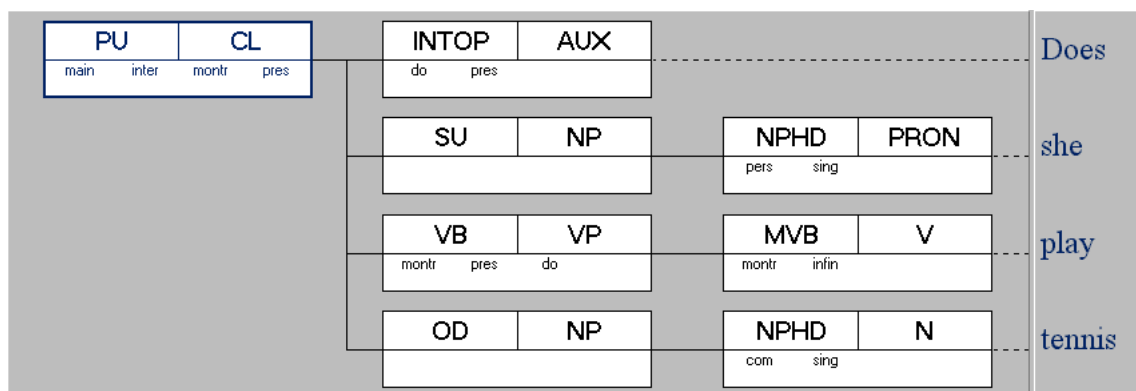


Figure 14: Syntactic tree for *does she play tennis*

By default, the tree “grows” from left to right, and from top to bottom. Each node has three sectors, shown in Figure 15.

The function and category sectors are always labelled, but the sector listing the features of nodes may be empty. In some cases (as above) no features are applicable.

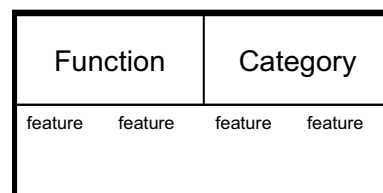


Figure 15: The sectors of a node

The topmost node on every tree carries the function label ‘PU’ (parse unit). In the example above, the PU is realised by a clause (‘CL’).

**NOTE:** For a complete description of the ICE parsing scheme, click on **Help | The Grammar**.

### 3.2 Fuzzy Tree Fragments (FTFs)

To search for syntactic labels and structures, ICECUP uses a technique called *Fuzzy Tree Fragments* or ‘FTFs’ for short. Using FTFs, you just draw a diagram of the structure you want to find in the corpus.

The simplest FTF of all consists of a single node. In our first example, we will construct an FTF to find all subjects in the corpus which are realised by clauses. This involves creating an FTF with just one node, labelled “subject” for function and “clause” for category.

There are two ways to perform this kind of search. A quick way uses the **Node** query window.

- Hit the large **Node** button, type ‘SU,CL’ and then hit <Return> or **OK**.

The other, more extensible method, is to construct an FTF.

### 3.3 Searching for a single node



We will now perform the same search by building a Fuzzy Tree Fragment.

- Click on the large **New FTF** button on the button bar.

A single, empty node appears on the screen (Figure 16).

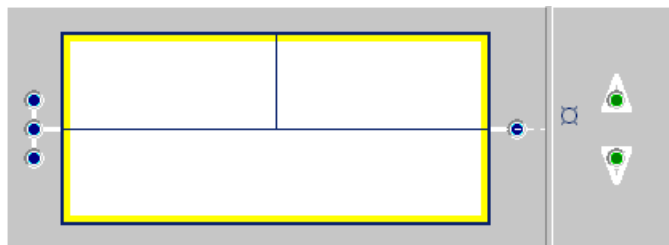



Figure 16: An empty FTF


We now label this node with the information we require. There are two ways of doing this.


- Click down with the right mouse button in the sector and select the function and category from the pop-up list (an example is shown on page 25).
- Click on the small  (edit node) button. Alternatively, hit function key <F2>.



If you do the second of these, an inspector window appears. All function and category labels are available from pull-down lists as in Figure 17.

- From the ‘current function’ control on the left, choose **subject**.

On the right, ‘compatible categories’ lists all valid realisations of subject: *adverb phrase*, *clause*, *disparate*, etc.

- From this list, double-click **clause**.
- If you make a mistake at any time, just press  to undo.

**NOTE:** This inspector window ‘floats’ over the screen. You can continue editing with the inspector open (or part open – click on  to hide the lower section if you wish).

By default you can only choose a single function, e.g. *subject*, for a given node. Occasionally you might want to specify a *set* of functions or categories. To do this, first release the switch  or hit ‘1’. To *negate* the function or category press  or hit ‘-’ or ‘¬’.

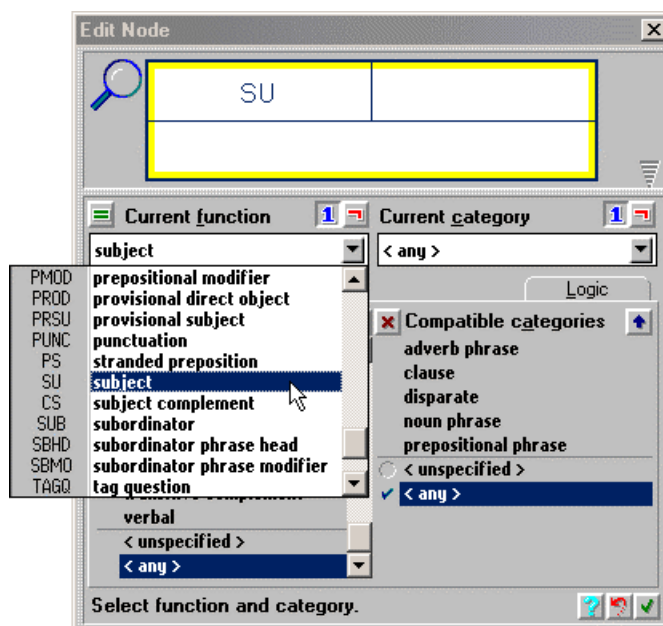


Figure 17: Selecting functions and categories in the ICECUP 3.1 inspector window

If you followed the steps overleaf your FTF should now look like Figure 18.

- Click on **Start!** or press <F4> to search for clausal subjects.

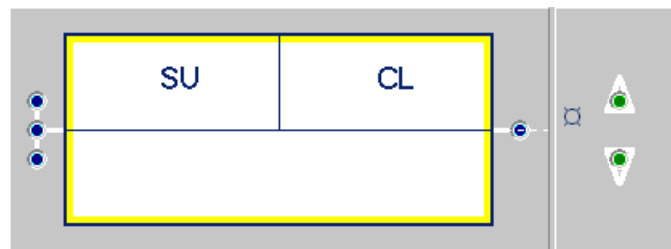


Figure 18: An FTF labelled for function (subject) and category (clause)

### 3.4 Searching for two or more nodes

We will now construct a slightly more complex FTF. In this example, we will search for all instances of direct objects that are immediately followed by a prepositional phrase functioning as an adverbial (something like *He read the letter on the bus*).

- Click on the large **New FTF** button on the button bar. An empty FTF opens on the screen.

For this search you will need a minimum of two nodes, one for the direct object, and one for the adverbial PP. To insert these nodes:

- Click *twice* on the '📄' (insert child after) button. Your FTF should now look like this:

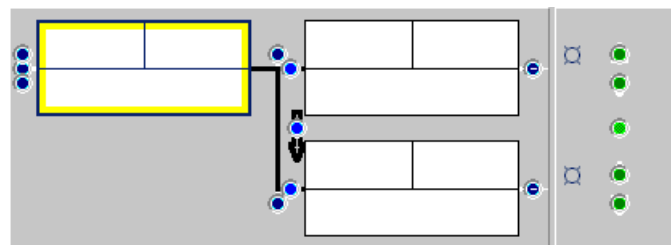


Figure 19: An empty FTF

The node with the yellow border on the left of the figure will be left empty. This node is the shared parent of the other two nodes.

We begin by labelling the node on the top right of the figure as a direct object ('OD').

- Click on this node to make it active. Then click on the '📄' (edit node) button.
- Select **direct object** from the alphabetical list of functions (see Figure 17 if necessary). Close the window if required.

Your FTF should now look like Figure 20, overleaf. If you place the mouse over the OD sector you should see the pop-up explanation.

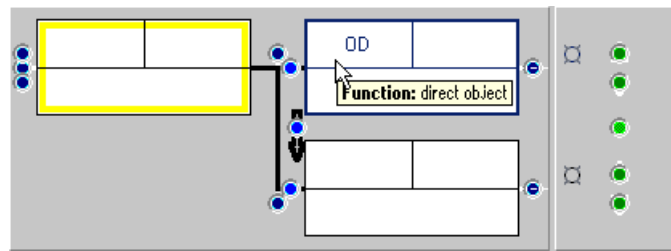



Figure 20: FTF with one node labelled for function (direct object)

Next, we will label the node below the OD as an **adverbial prepositional phrase**:

- Click on the lower node to make it active. Click on the  button again if necessary.
- Select **adverbial** from the alphabetical list of functions. (**Tip**: press ‘A’ several times.)
- Select **prepositional phrase** from the list of categories.

Your FTF should now look like this (Figure 21):

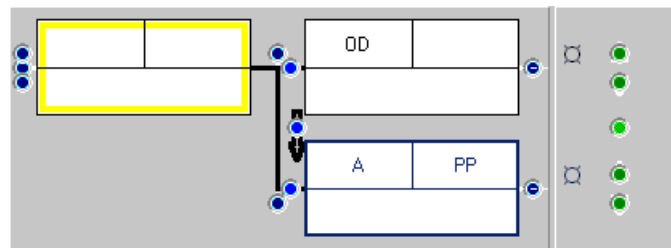


Figure 21: The completed FTF

- Click on **Start!** or press <F4> to launch this new search.

This search takes a little longer to carry out, because all the distinct elements in the FTF – direct object, adverbial, and prepositional phrase – are very frequent in the corpus, and there are a lot of candidates to sort out. The search takes place in the background. You can suspend the search at any time by pressing the **Stop!** button on the top right. ICECUP will display the hits it has found up to that point.

Your results screen will look something like this (Figure 22) while the search progresses:

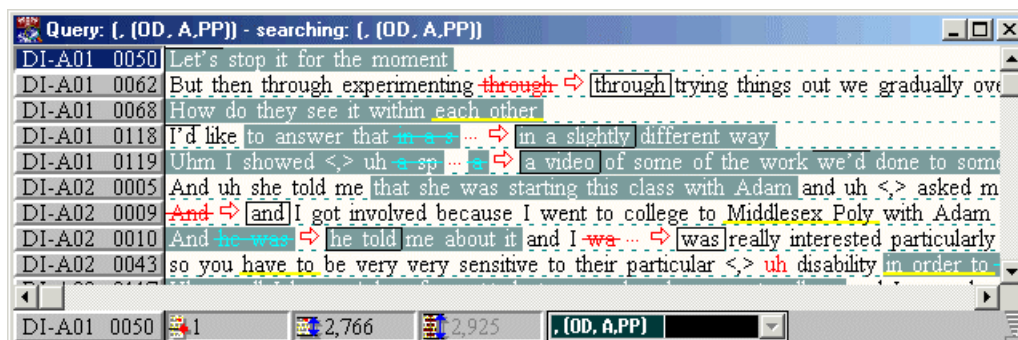


Figure 22: An FTF search in progress in DCPSE

### 3.5 The FTF Focus

In the previous set of results (Figure 22), you may have noticed that the *entire clause* in which our search argument occurred was highlighted. For instance, in the first matching case, the entire parse unit, *Let's stop it for the moment*, is highlighted.

Now, this is not always useful, especially when we're examining complex constructions. Usually, we wish to focus on one part of a construction, and to align our concordance on that part. In this example, it would probably be more useful to focus on the direct object alone, or on the adverbial alone, rather than on the whole clause.

This is where the **FTF Focus** comes in. The FTF focus is indicated by a yellow border around a node in the tree fragment. In our previous FTF (Figure 21), the yellow border is currently on the node which immediately dominates the construction we're looking for (the 'dummy' node). This is why the whole clause is highlighted in the results.

To move the focus to the direct object, we simply move the yellow border, as follows:

- Click on the OD node in the FTF.
- Click on the small '🏠' (mark FTF focus) button on the button bar.

The yellow border will move to the OD node.

- Click on **Start!** or press <F4>.

A new window appears, showing the same results as before, but this time only the direct object is highlighted in each hit. In a concordance view (Figure 23), these results focus on the direct objects:

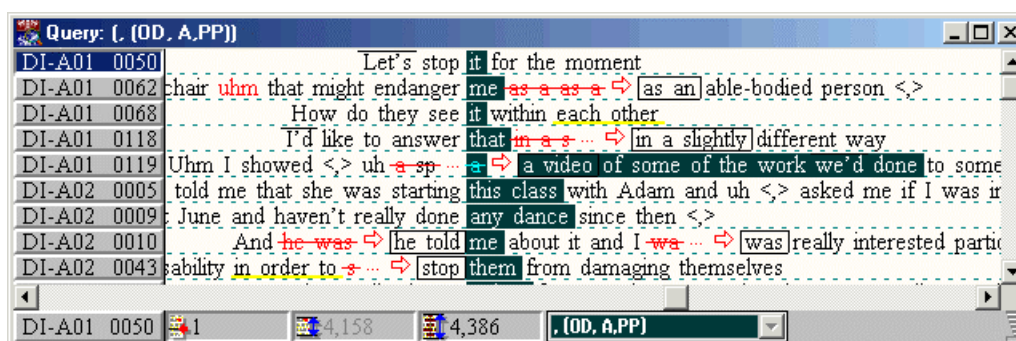


Figure 23: Concordanced results, with the direct object as focus

The FTF Focus can span more than one adjacent node. To do this, first hold down the <Shift> key and click on nodes to include and then press the '🏠' button. All the selected nodes will now display the yellow focus border.

### 3.6 Adding words to an FTF

You can include words in an FTF, as well as syntactic labels. For instance, you may wish to specify that the direct object in our previous example must contain the word *it*.



- In the previous FTF (Figure 18), click on the OD node. Click on the small 'T' (edit word) button. The **Edit Word** dialog box, shown in Figure 24, now appears.
- Type the word *it* into the dialog box. Press **OK**.

In the FTF, the word *it* is shown to the right of the OD node, as shown in Figure 25.

- Click on **Start!** or <F4> to launch the search.

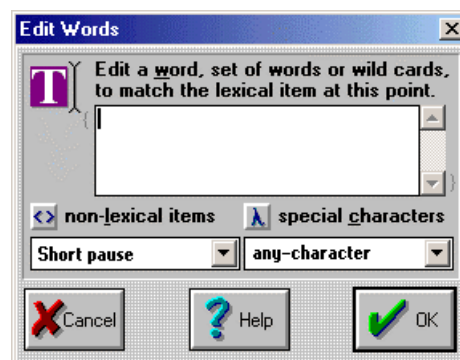


Figure 24: The Edit Word dialog box

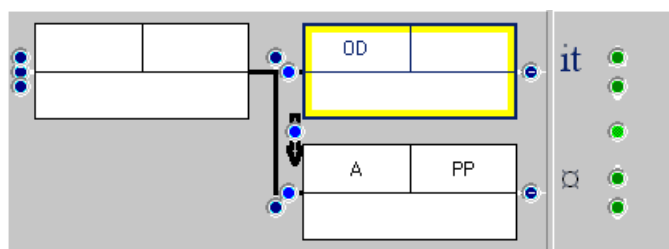


Figure 25: An FTF including the word *it* in the direct object

### 3.7 The FTF Wizard



The FTF Wizard is a quick and easy way to create an FTF from an existing tree in the corpus.

One of the classroom lessons in the corpus, S1B-010 (or DI-B80 in DCPSE), contains the following sequence uttered by one speaker:

S1B-010	106	Science cannot answer the question <b>why</b>
S1B-010	107	Never can it
S1B-010	108	It can only answer the question <b>how</b> <.,>

Figure 26: Extract from S1B-010 / DI-B80

- Open **Query | Corpus Text...** and type "S1B-010" (this works in DCPSE as well). Jump to the correct line by clicking on the "Current text unit number" in the status line and then over-typing the line number. (You can also hit <Tab> twice and then type.)

Suppose we are interested in the inverted construction *Never can it*. Figure 27 shows the tree.

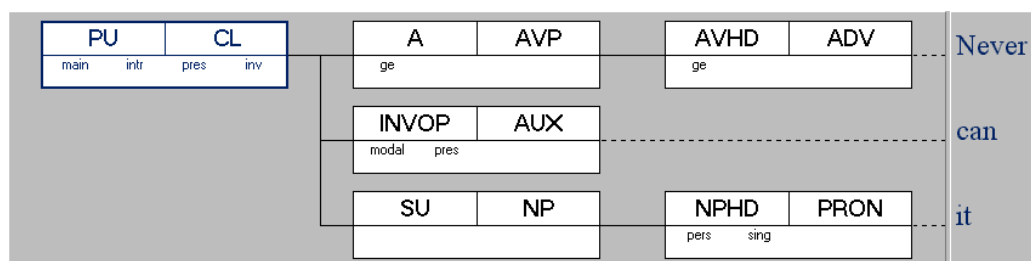


Figure 27: Tree for *Never can it* (S1B-010: 107)



To search for all similar structures in the corpus, we can take a “snapshot” of this one, and use it to construct a new FTF.

- In the tree view window, click on the large **Wizard** button.

A dialog box appears (Figure 28), in which you are offered several options.

Since we’re interested here in the inverted construction, we will select *Base it on the tree* (the default setting). A number of “tree options” are now visible on the left of this dialog box.

The most important option is “**Prune tree**”. This allows you to specify how much of the original tree to keep. Here, we’ll keep the immediate children of the selected node.

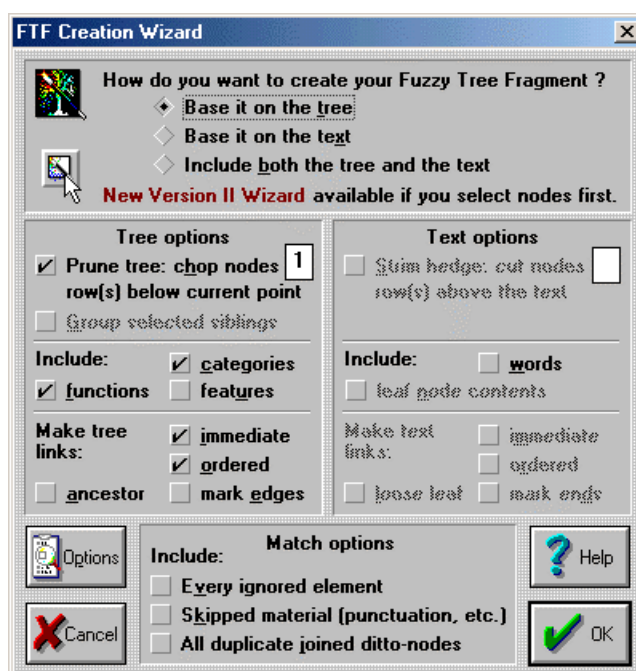


Figure 28: The FTF Wizard dialog box

- Click on **OK**.

The new FTF appears in a new window (Figure 29).

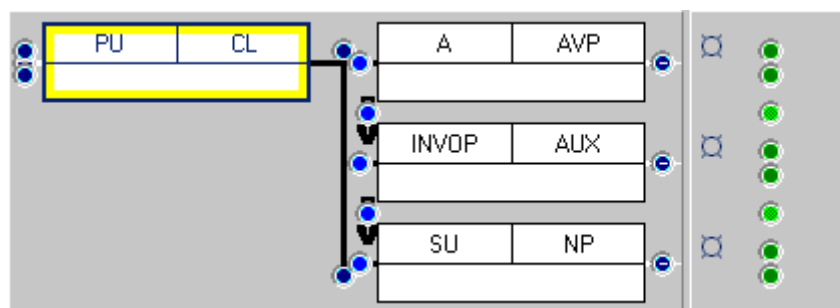


Figure 29: An FTF produced by the Wizard

This is a snapshot taken from the corpus itself. You could begin the search now, though we suggest you first remove the PU (parse unit) label. If you don’t, the FTF will retrieve the construction only if it appears at the top of a tree, so it will miss other cases. To clear the function label:

- Right-click on PU and select ‘< any >’ from the menu list that appears (rather misleadingly, this was labelled ‘< none >’, i.e., with no limit, in ICECUP 3.0).

The node should now just contain the CL label (Figure 31 overleaf). This query is a little more general than before.

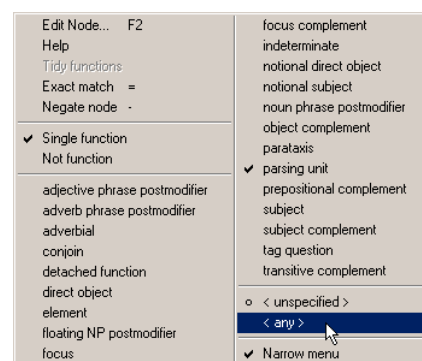


Figure 30: Clearing the function by selecting ‘< any >’

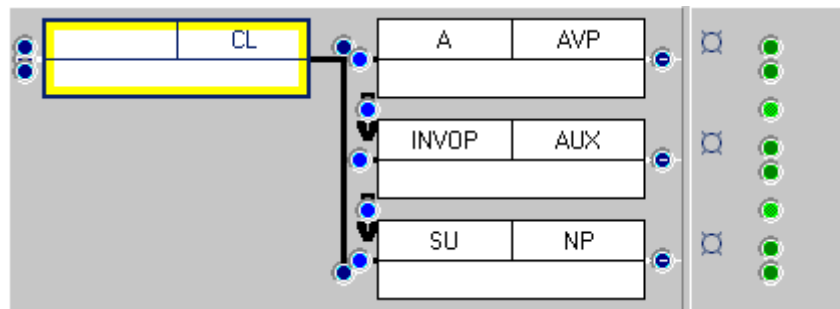



Figure 31: The completed FTF

### 3.8 The FTF Wizard (Version II)

ICECUP 3.1 contains a new way of generating FTFs from syntactic trees. The idea is you first mark nodes to include them in your FTF and then convert them into a query. If you return to the tree we started with (see Figure 27 on page 24), you can try this out.

- Click on the tree window for *Never can it* (S1B-010 #107) to bring it to the front.
- Click with the *right* mouse button to mark the following nodes:
  - **adverbial phrase** (A, AVP);
  - **inverted operator** (INVOP, AUX); and
  - **pronoun NP head** (NPHD, PRON).
- Alternatively you can select the node and then press <Insert> or hit the  (select node for wizard) button.
- Hit the big 'Wizard' button again. This time (Figure 32) there are fewer options.

The clever bit is how the new wizard makes decisions about how to deal with nodes in the tree *you did not mark*.

If you tick the box to 'make tree links **immediate**', the FTF retains any intervening blank nodes (Figure 33, left). Otherwise, depending on the topology, these nodes may be removed and white 'eventual' links are added (right).

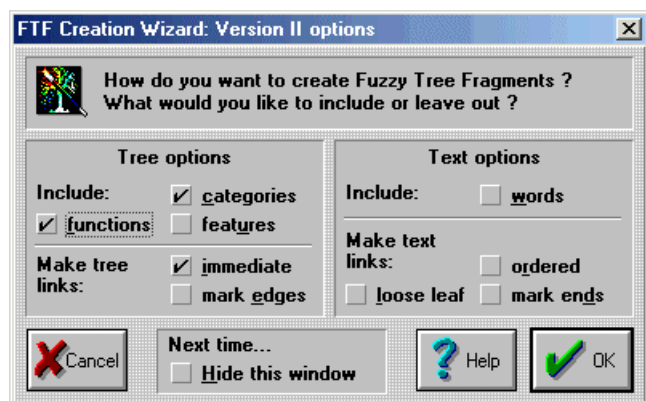


Figure 32: The new wizard window

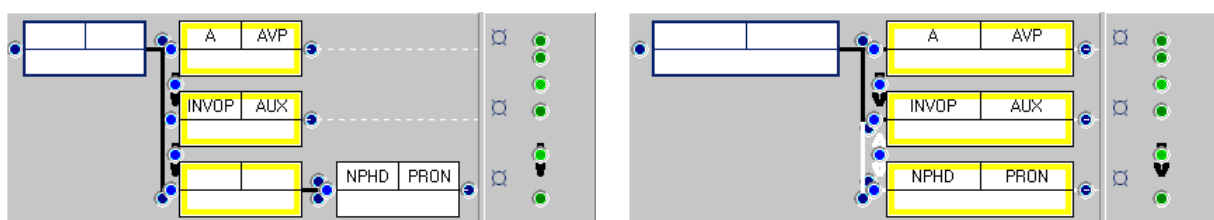


Figure 33: Two different FTFs, with (left) and without (right) immediate tree links and empty nodes.

Note there must still always be one common parent node.

We strongly recommend you experiment with the Wizard. ICECUP will remember the nodes you've marked in the tree unless you close the tree window or continue browsing the original text. So if your initial results are not very useful, you can close the FTF you just created and simply try again with different settings.

**Tip:** If you find yourself applying the same Wizard parameters to every tree you can even skip going through the process of hitting 'OK' in the Wizard window. Just click on the **Next time... Hide this window** option. (To alter settings go to **Query | Wizard II options...**).

### 3.9 Saving FTFs



Once you have created an FTF, you can save it for future use as follows:

- With the FTF open, click on the large **Save** button.

The **Save** dialog box will appear.

- Select the drive and directory where you wish to save the file.
- Type in the name of the FTF you wish to save, say '**dobject**' for an FTF designed to find direct objects (you can use a maximum of 8 characters in the filename). The filename will be 'dobject.ftf'.
- Click on **OK**.

To open a saved FTF, click on the large **Open** button on the button bar and select the appropriate file.

**NOTE:** All the FTFs illustrated in *Getting Started* are available on the CD, in the *Examples* directory.

## PART 4: Subcorpora

Searches in ICECUP are carried out across the whole corpus unless you specify otherwise. If you are only interested in carrying out research into a particular subcorpus, or wish to contrast one part of a corpus with another, you must first define these subcorpora.

What if we wanted to restrict a search to, say, just the London-Lund part of DCPSE, or the ICE-GB written texts?

The **Text Fragment**, **Node**, **Variable** and **Random Sample** query windows all contain a panel (right) which lets you apply the query to a given subcorpus.

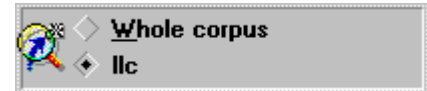



Figure 34:  
The 'apply to subcorpus' panel

This panel lets you apply the search to the *currently visible subcorpus*. This means either the *currently selected subvalue* in the Corpus Map (Section 4.1 below), or the *current query results* (Section 4.3).

### 4.1 Searching within a given subcorpus

In our first example we will apply the simple Text Fragment query 'play' to the LLC-sourced part of DCPSE. This involves two stages:

#### Stage 1: define the subcorpus

- Open the Corpus Map, if closed, by hitting the large **Map** button.
- Select **Source Corpus** using the variable selector. To do this, double-click on the title line and locate **Source Corpus** in the selector. Alternatively, press <F2> and then 'End'.
- Next, open the **Source Corpus** variable in the corpus map to show its subvalues, **ice-gb** and **llc**. You can click on the 'Expand to values' button (  ) or press <Ctrl> and '1' together. Finally, select 'llc'.

#### Stage 2: apply the search

- Click on the large **Text** button on the button bar. The 'apply to' panel should now show 'Whole corpus' and 'ice-gb'.
- Type 'play' and select 'llc' in the panel (Figure 35). Click on **OK**.

### 4.2 Opening and browsing a subcorpus

We have looked at a typical subcorpus in DCPSE. Our second example concerns a typical contrast in ICE-GB: between speech and writing. We will define and browse a subcorpus of written texts in ICE-GB, and then apply our queries to this subcorpus.



Figure 35: Applying a search for 'play' to ICE-GB part of DCPSE

- If the Corpus Map is closed, open it again by clicking on the **Map** button. **Text Category** should be visible in the Variable selector – if not, press <F2> and ‘Home’.
- Selectively expand the map by double-clicking on the icons marked with a ‘plus’ (Figure 36, right).

Text Category is the *principal sampling variable* for the corpus. The major subdivision in ICE-GB is between speech and writing. The **spoken** category is then split into dialogue, monologue and mixed. DCPSE contains spoken material only, and is subdivided into different types of speech transcript, including formal and informal dialogue and other types of specialized speech.

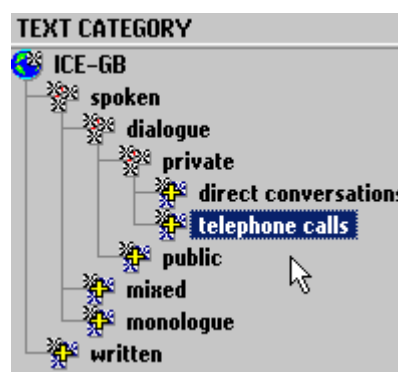


Figure 36: Part of the ICE-GB Corpus Map, organised by Text Category

The map expands first to show the two major subdivisions of the corpus: spoken and written.

- Select **written** and click on the large **Browse** button at the top right (or hit <F4>).

All the written texts appear in a new window (Figure 37), starting with the first, W1A-001:

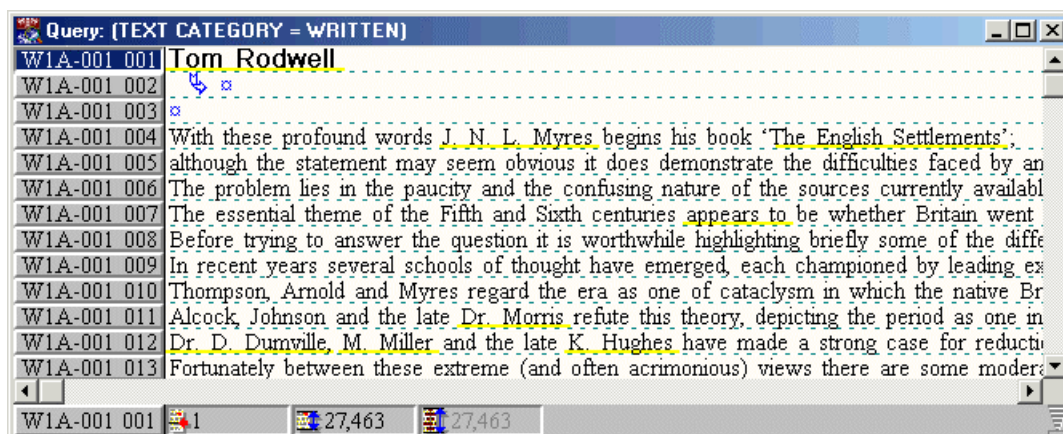


Figure 37: A subcorpus of written texts in ICE-GB

Notice that this window is headed:

**Query: (TEXT CATEGORY = WRITTEN)**

This is a view of the subcorpus of all the written texts in ICE-GB.

### 4.3 Searching in an open subcorpus

Now let us conduct a search within this subcorpus. The search procedures are exactly the same as those outlined in ‘Stage 2’ in the previous section. Again, before you launch each search, you specify whether it should include the whole corpus or your defined subcorpus.

Suppose we perform a simple lexical search for the word *apple*:

- Click on the **Text** button. In the **Text Fragment** box that opens, type ‘apple’.

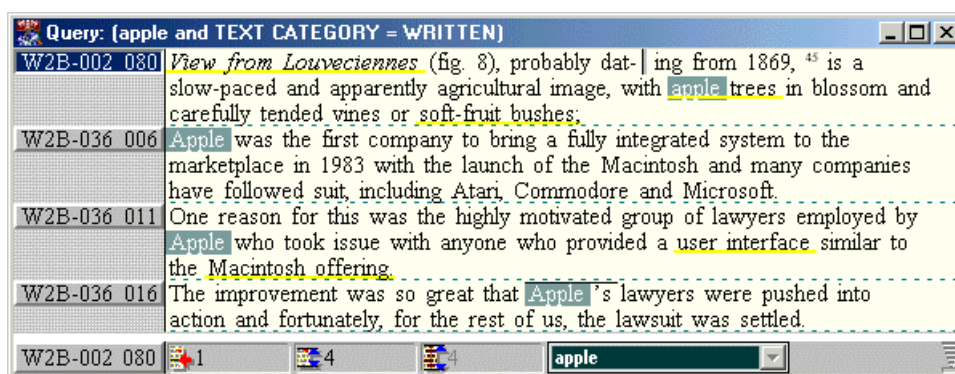
The ‘apply to’ panel of this dialog box (see Figure 34) offers two choices, ‘Whole corpus’ and ‘Query: (TEXT CATEGORY = WRITTEN)’ (part of this will be hidden).

- Select the second option to search within the current query window.
- Click on **OK** to launch the search.

The search retrieves four instances of *apple* in the written corpus. The results are displayed in a new window (below).

Notice that this results window is headed:

**Query: (apple and TEXT CATEGORY = WRITTEN)**



**Figure 38: Results of the search for “apple” in written texts**

**Tip:** This view is *word-wrapped* to show all the examples.



- To switch to this mode, press <F2> a few times, double-click on the total number of text units in the status line, or select word wrap mode from the button bar (Figure 39).



**Figure 39: Choosing word wrap mode**

#### 4.4 Combining variables to build specialised subcorpora

Subcorpora can be more complex than simple subcategories of the corpus. Text categories can be combined with each other, and with other variables, such as speaker age and gender. In our final example, using ICE-GB we will define a subcorpus of utterances by female speakers in telephone calls.

- Click on **Map** and expand the spoken component by double-clicking on the  icon.
- Expand this branch by double-clicking on the  icon, until you have opened **spoken: dialogue: private**, as in Figure 36 on page 29.

Two subcategories appear under **private**: direct conversations and telephone calls.

- Click on **telephone calls** to highlight this text category.
- Click on the large **Browse** button at the top right.



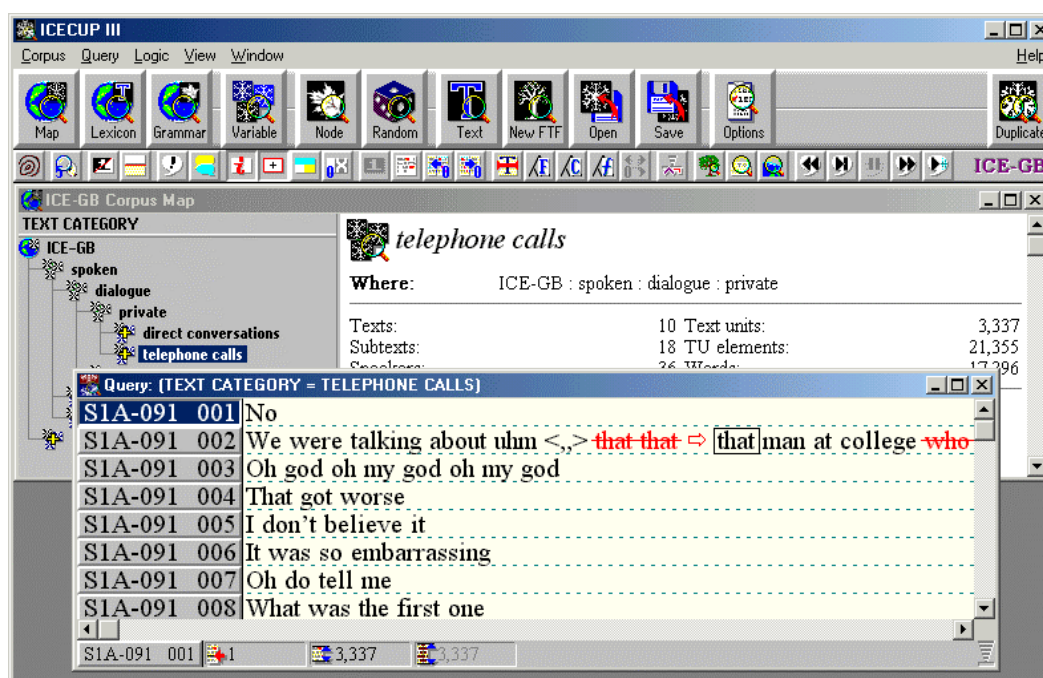


Figure 40: Subcorpus of telephone calls

A subcorpus containing just the telephone calls opens in a new results window. Your screen should now look something like Figure 40. Notice that this results window is headed:

**Query: (TEXT CATEGORY = TELEPHONE CALLS)**

Now we will refine the subcorpus to one of female speakers in telephone calls.

- Without closing the results window, return to the Corpus Map, and select **SPEAKER GENDER** from the Variable Selector (use the mouse to click and scroll down).
- Double-click on the 'world' icon next to '**ICE-GB**'. The map expands to show three values, *female*, *male*, and *co-authored*.
- Click on **female** to highlight this value.

At this point we will introduce one of ICECUP's special features, "drag and drop".

- Press and hold down the left mouse button on the 'world' icon next to **female** in the map.

The query element, '**SPEAKER GENDER = FEMALE**', will expand under the mouse arrow. This element can be dragged around the screen by holding down the mouse button and moving the mouse (Figure 41).

- Drag '**SPEAKER GENDER = FEMALE**' across the screen with the mouse. Drop it into the window containing the subcorpus of telephone calls.

This window is automatically updated. It now displays only those utterances in telephone calls spoken by women. Click on the results window. Your screen



Figure 41: Dragging an element from the Corpus Map

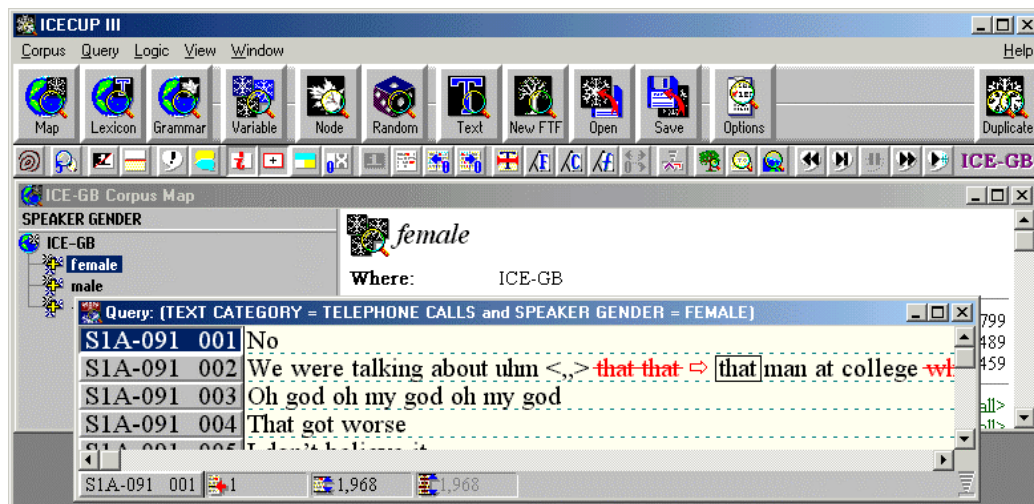


Figure 42: A subcorpus of utterances by female speakers in telephone calls

should now look something like Figure 42. The results window is now headed:

**Query: (TEXT CATEGORY = TELEPHONE CALLS and SPEAKER GENDER = FEMALE)**

This 'drag and drop' method can be used to create any (feasible) subcorpus, by combining any of the variables in the corpus map.

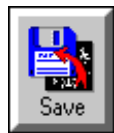
It can also be used to create complex queries by combining FTFs. For more information, see Drag and Drop Logic, p.35.

**Tip:** When dealing with several windows, they will overlap unless you organise them using *tile window* options. Try using **Window | Tile Horizontal** or **Tile Vertical**.



## PART 5: Saving your Results

### 5.1 Saving the results of a search



The results of any search can be saved by clicking on the large **Save** button at the top of the screen.

The **Save to Disk** window will appear (Figure 43).

- From the options offered, select whether you wish to save the **Whole query** (all the citations) or **Just current text unit**.
- If you hit **Save** when viewing a tree, you will see options for just saving that text unit.

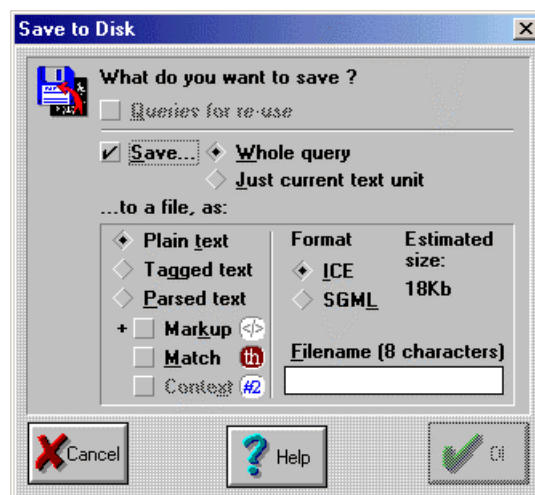


Figure 43: The Save to Disk dialog box

You are offered several choices about the type of information to save: **Plain text**, **Tagged text** and **Parsed text**. These different formats are summarised in Table 3 overleaf.

The matching part of the text can be highlighted in the output file by selecting the **Match** option in the **Save to Disk** dialog box. When this option is selected, matching words are enclosed within marks like this “\*\*[ ... ]\*\*” in the output file.

The **Save to Disk** command also lets you save **Markup** in the output file. Structural markup encodes features of the text such as paragraph boundaries, boldface print, self-corrections, and overlapping speech. In selecting this option, you should be aware that spoken texts in particular contain a great deal of structural markup (although it remains hidden in ICECUP’s default view). Unless you are specifically interested in this annotation, it is usually not very useful to save it in the output.

The final option, **Context**, lets you save context sentences (see Section 2.4) as simple text.

When you have selected all your options:

- Type in the filename (a maximum of 8 characters). Click on **OK**.

**NOTE:** By default all results are saved to a directory called ‘output’ on your hard disk, such as **c:\output**. With a network licence the software may be installed across a shared network but it is a good policy to make sure that exported material is saved in a user’s personal file space. (It is not a good idea to export material directly to a floppy disk because resulting files can be large!)

You can now also output results in an SGML format, which is more verbose than the standard versions. You can also export corpus map and lexicon tables to disk. For more information see the on-line help files.

<b>Plain text</b> (Plain text output files have the file extension <b>.txt</b> )	This will save only the lexical items, pauses, and punctuation. e.g. <ICE-GB:S1A-020 #221:1:B> Does she play tennis
<b>Tagged text</b> (xxx.tag)	This saves all lexical items together with their grammatical tags. e.g. <ICE-GB:S1A-020 #221:1:B> Does <AUX(do,pres)> she <PRON(pers,sing)> play <V(montr,infin)> tennis <N(com,sing)>
<b>Parsed text</b> (xxx.tre)	In this format, the saved text contains all lexical items, grammatical tags, and syntactic labels. The output is in the form of an indented file, in which the indents correspond to the grammatical hierarchy in the tree. e.g. <ICE-GB:S1A-020 #221:1:B> PU,CL(inter,montr,pres) INTOP,AUX(do,pres) {Does} SU,NP() NPHD,PRON(pers,sing) {she} VB,VP(montr,pres,do) MVB,V(montr,infin) {play} OD,NP() NPHD,N(com,sing) {tennis}

Table 3: Format options for saved search results (DCPSE also saves the ICE-GB or LLC text code)

## 5.2 Saving Corpus Map tables and Lexicons

You can also click on the Save button to save material from the corpus map, lexicon or grammaticon. In this case the window offers different options, including the option of saving tables of statistics in a form that can be easily read by a spreadsheet program. Files are saved as plain text, tables with 'tab' characters between columns.

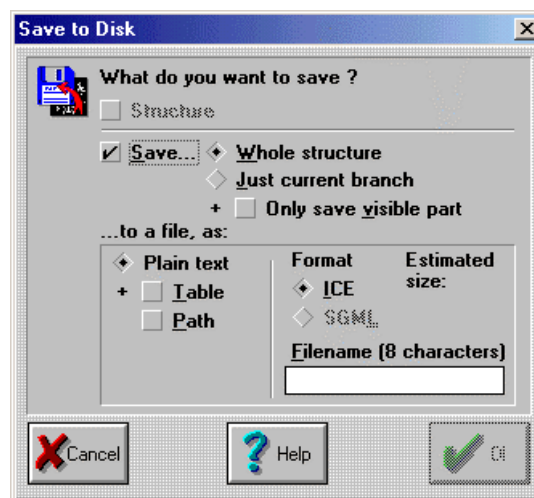


Figure 44: Outputting data from the corpus map, lexicon and grammaticon

## PART 6: Advanced Features

Here we briefly introduce four very useful advanced features of ICECUP. Two of them, Drag and Drop Logic and Random Sampling, were available in ICECUP 3.0. The last two, the lexicon and grammaticon, are new to ICECUP 3.1.

### 6.1 Drag and Drop Logic

In Part 4, Subcorpora, we showed how you can drag a variable from the Corpus Map into another window to combine queries and create a subcorpus.

This “drag and drop” technique has other important applications in ICECUP. You can combine queries with each other, and drag part of one query into another. You can even drop an element into the corpus map to add a column to the table of statistics.

You can edit a combined query, and the logical relationships between elements in the query, by opening the logic editor in the query results window.

Drag and drop logic is explained in detail in the main help manual. For more information, click on **Help | Combining Queries**.

### 6.2 Random Sampling



The **Random** command allows you to create a random sample of the corpus. This is particularly useful when you are investigating high-frequency items, such as nouns or clauses. In many cases you may wish to retrieve just a small sample of these, rather than every instance in the corpus.

- To generate a random sample click on the large **Random** button on the button bar.

This opens the Random Sample dialog box shown in Figure 45:

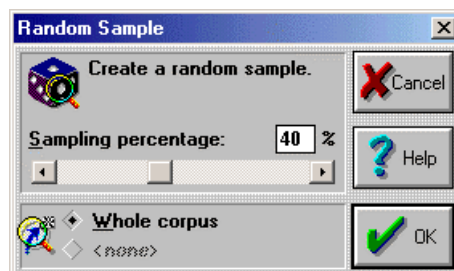


Figure 45: The Random Sample dialog box

- Specify the percentage of the sample. 100% generates the whole corpus, 0% an empty list. A number between 1 and 99% generates a genuine random sample of the corpus.

This random sample can then be combined with other queries using drag and drop. It can also be saved for later retrieval, using the **Save** command. This allows you to compare a number of queries over the same random set of sentences.


### 6.3 The Lexicon



The Lexicon is a new feature in ICECUP 3.1. Every word in the corpus has an entry in the lexicon. Every distinct grammatical tag, including the part of speech, for each word is given a sub-entry.

- Click on the large button marked **Lexicon**, or select **Corpus | Lexicon**.

The Lexicon window will open. It looks like the corpus map (see Section 1.2).

- Click on the ‘expand all’ button (‘’). The display should look like Figure 46.

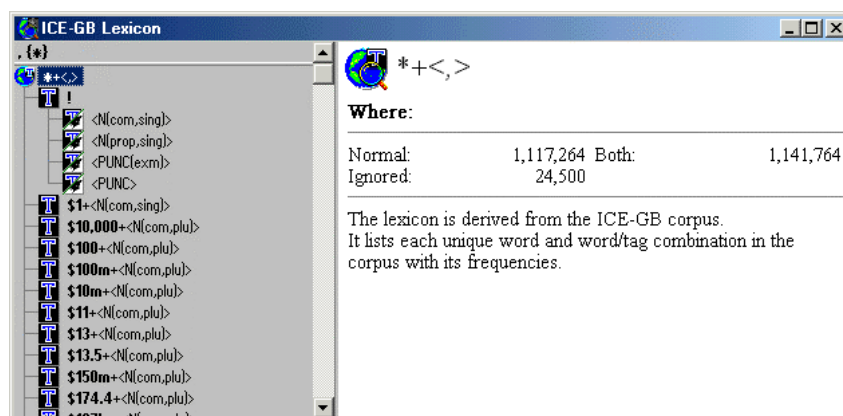


Figure 46: Default Lexicon view

The default view is a flat list of every distinct lexical item and its word class tag.

You can browse the lexicon using the cursor keys and mouse, or using the *Quick Find* command (see Figure 5 on page 12) in the button bar. Figure 47 shows how the exclamation mark is analysed in ICE-GB. Although it is mostly treated as a punctuation symbol, occasionally (4% of the time) it is treated as part of a compound noun.

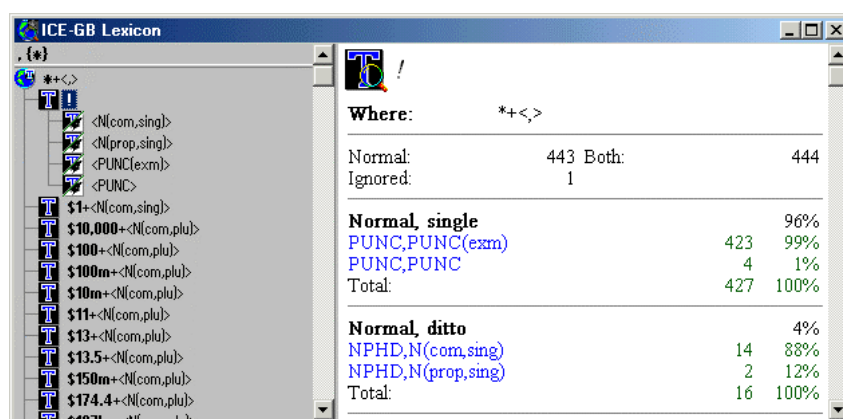



Figure 47: The lexicon entry for a single exclamation mark

It is often useful to organise the lexicon to focus it on a particular task.

You can do this in two main ways: by *restricting* and by *structuring* the lexicon. Let's restrict the lexicon grammatically and lexically.

- Press the ‘lexicon options’ () button.
- Type ‘N’ (noun) into the Node rectangle and ‘fi\*’ (starts with “fi”) into the Word one.
- Hit **OK**.

This will create a lexicon of all nouns starting with the letters *fi*, as in Figure 49.

- Double-click on one of these elements. ICECUP shows matching cases in a new window.

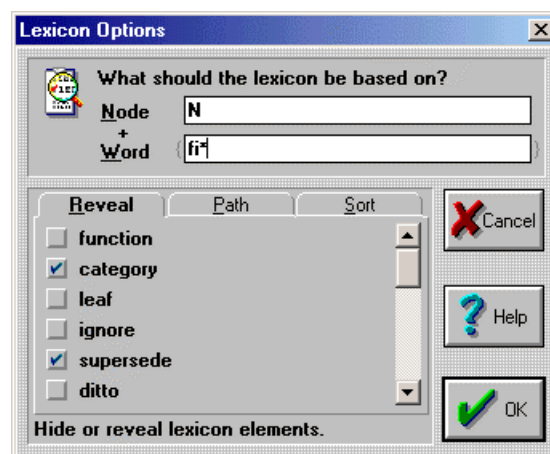



Figure 48: Setting lexicon options

**Tip:** you can make your Lexicon view automatically update this second window as you browse the lexicon, by switching on ‘autoconnect’ () *before* opening the window.

**Tip:** use **Window | Tile Vertical** to organise your view.

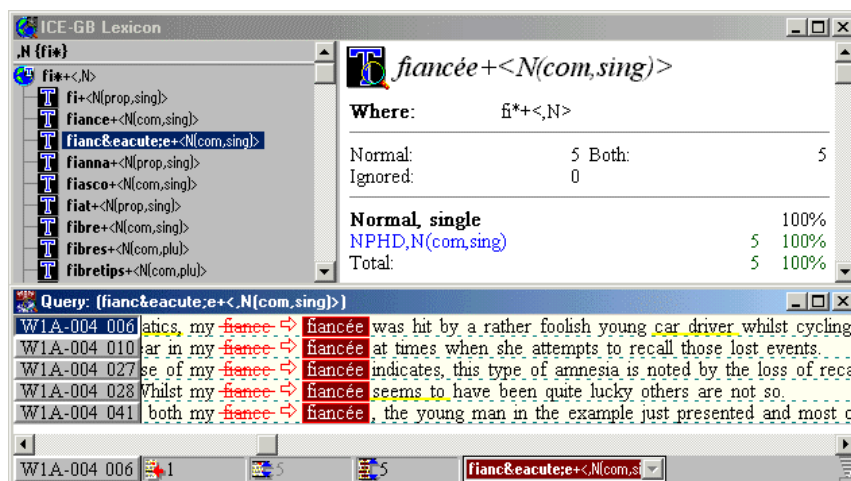



Figure 49: Viewing cases in the lexicon

## 6.4 The Grammaticon



The Grammaticon is similar to the lexicon. Every syntactic node in the corpus has an entry in the grammaticon. By default the structure is flat, so typically you would first have to define its structure. To structure the grammaticon:

- Open the tool by clicking on the big **Grammar** button.
- Click on the ‘grammaticon options’ () button.

The following is a typical example of a structured grammaticon. It is subdivided first by function and second by category.

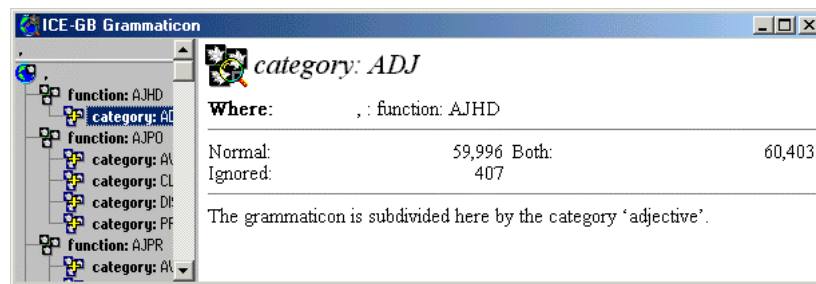


Figure 50: Grammaticon structured by function, then category

The options window (Figure 51) lets you structure and limit the grammaticon. Grammaticon and lexicon options are very similar. You can restrict the grammaticon to just noun patterns, say, by entering ('N') into Node. You can also structure it using the tabbed options.

- Reveal or hide parts of each node.
- Create a tree structure for the grammaticon by making a 'path' (Figure 51, right).
- Specify the sorted order of leaves.

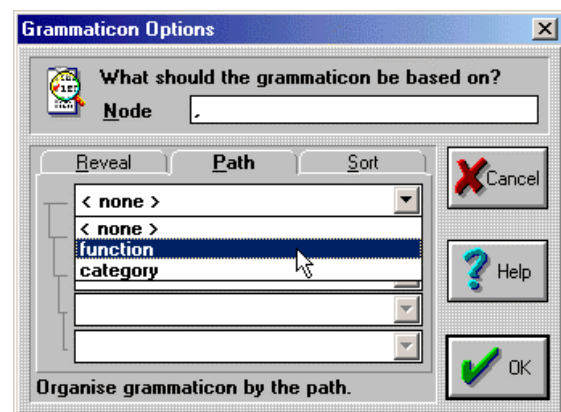
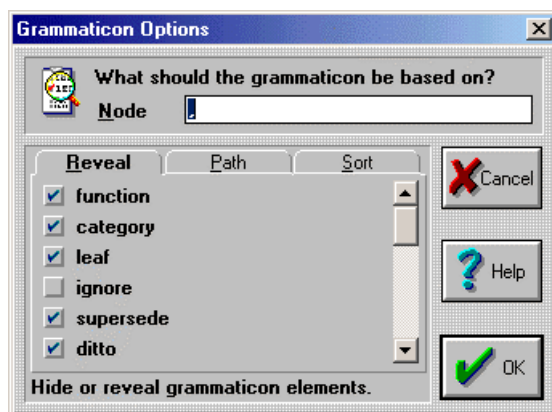


Figure 51: Different grammaticon options

Like the corpus map (see Section 1.2, page 11), the grammaticon and lexicon also support statistical tables calculated from the corpus. As you can see in Figure 52, each column corresponds to a single entry in the middle section of the panel on the right.

- Press the **Show Table** (📊) button to reveal the table.

	Normal	Ignored	Both
function: AJHD	2,056,101	31,582	2,087,683
category: ADJ	59,996	407	60,403
function: AJPO	5,187	30	5,217
category: AVP	171	0	171
category: CL	1,855	12	1,867
category: DISP	11	0	11
category: PP	3,150	18	3,168
function: AJPR	8,521	28	8,549
category: AVP	8,367	27	8,394

Figure 52: A simple grammaticon table



## PART 7: Using Help

*Getting Started* describes only a small number of the features available in ICECUP. To explore the corpus and the software fully, we strongly recommend that you consult the Help file in ICECUP. To do this:-

- Click on **Help** on the menu bar. The drop-down menu in Figure 53 (left) appears. As Fuzzy Tree Fragments are so central to ICECUP, they merit a special subsection (right).

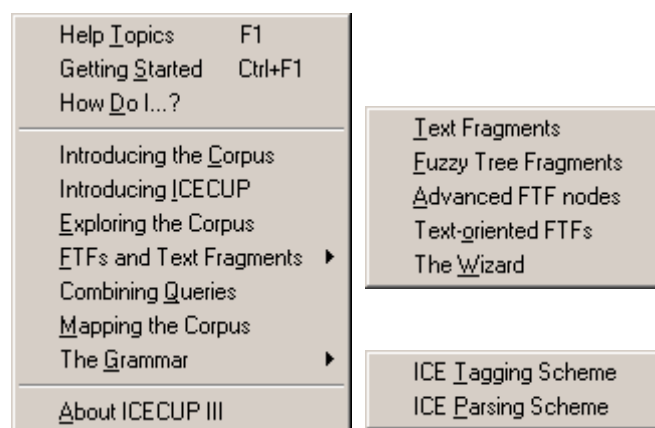


Figure 53: ICECUP's Help menu

The ICE Grammar Reference contains full documentation on the ICE tagging and parsing schemes, and is divided accordingly. The wordclass tags, as well as the function, category and feature labels, are listed alphabetically and cross-referenced where appropriate. The Grammar Reference is full of little 'example' icons, '🔍', which search the corpus using ICECUP to find examples of the element you were reading about.

**Tip:** To find other examples of a selected tree node, hit <F7>. Press <Shift> and <F7> to edit the Node search first.

## Further reading

We hope that this introduction to ICECUP has helped you start exploring some of the possibilities of this software. ICECUP has been developed in the recognition that different users have different needs, but that we all wish to explore the grammar of a parsed corpus.

In order to learn the full functionality of ICECUP and explore ICE-GB and DCPSE we recommend that you read *Exploring Natural Language: Working with the British Component of the International Corpus of English*, published by John Benjamins in 2002 (Nelson, Wallis and Aarts 2002). This book is a reference for ICE-GB, but we would highly recommend it for users of DCPSE as well. The handbook summarises the ICE grammar, and gives many case studies showing how you can use ICECUP to explore the corpus and carry out a range of experimental investigations. It also discusses questions of good experimental design.

You can also visit our website, [www.ucl.ac.uk/english-usage](http://www.ucl.ac.uk/english-usage), for more guidance and reference material, to contact us, and to download the latest version of ICECUP.

## References

- Aarts, Bas and Flor Aarts (2002) 'Relative *Whom*: a 'Mischief-Maker''. In: Andreas Fischer, Gunnel Tottie and Peter Schneider (eds.) *Text Types and Corpora*. Tübingen: Gunter Narr Verlag. 123-130.
- Aarts, Jan, Hans van Halteren, and Nelleke Oostdijk (1996) 'The TOSCA Analysis System'. In: C. Koster and E. Oltmans (eds.), *Proceedings of the first AGFL Workshop*. Nijmegen: CSI. 181-191.
- Denison, David (1998) Syntax. In: S. Romaine (ed.), *The Cambridge History of the English Language*. IV: 1776-1997. Cambridge. 92-329.
- Greenbaum, Sidney (1988) 'A Proposal for an International Corpus of English', *World Englishes* 7. 315.
- Kennedy, Graeme (1998) *An Introduction to Corpus Linguistics*. London: Longman
- Leech, Geoffrey (2000) 'Diachronic linguistics across a generation gap: from the 1960s to the 1990s'. Paper read at the symposium Grammar and Lexis. University College London/ Institute of English Studies.
- Mair, Christian (1995) 'Changing Patterns of Complementation and Concomitant Grammaticalisation of the Verb *help* in Present-Day English'. In: Bas Aarts and Charles F. Meyer (eds.), *The Verb in Contemporary English*, Cambridge. 258-272.
- Mair, Christian (1997) 'Parallel Corpora: a Real-Time Approach to the Study of Language Change in Progress'. In: M. Ljung, M. (ed.) *Corpus-Based Studies in English*. Amsterdam. 195-209.
- Mair, Christian and Marianne Hundt (1995) 'Why is the Progressive Becoming More Frequent in English? A Corpus-Based Investigation of Language Change in Progress'. *Zeitschrift für Anglistik und Amerikanistik* 43.2. 111-122.
- Mair, Christian and Marianne Hundt (1997) 'The Corpus-Based Approach to Language Change in Progress'. In: U. Böker and H. Sauer. (eds.), *Anglistentag 1996*. Dresden. 71-82.
- Mair, Christian and Geoffrey Leech (2006) 'Current changes in English syntax'. In B. Aarts and A. McMahon. (eds.), *The handbook of English Linguistics*. Malden, MA: Blackwell Publishers. 318-342.
- Nelson, Gerald (1996a) 'The Design of the Corpus'. In: S. Greenbaum (ed.), *Comparing English Worldwide: The International Corpus of English*, Oxford: Clarendon Press. 27-35.
- Nelson, Gerald (1996b) 'Markup Systems'. In: S. Greenbaum (ed.), *Comparing English World wide: The International Corpus of English*, Oxford: Clarendon Press. 36-53.
- Nelson, Gerald, Sean Wallis and Bas Aarts (2002) *Exploring Natural Language: Working with the British Component of the International Corpus of English*. Amsterdam: John Benjamins.
- Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech and Jan Svartvik (1972) *A grammar of contemporary English*, London: Longman.
- Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech and Jan Svartvik (1985) *A comprehensive grammar of the English language*. London: Longman.
- Smith, Nicholas and Geoffrey Leech (2001) 'Grammatical Change in Recent Written English, Based on the FLOB and LOB Corpora'. Paper read at the ICAME conference. Louvain-la-Neuve, Belgium.
- Svartvik, Jan (1990) *The London-Lund Corpus of Spoken English: Description and Research*. Lund Studies in English 82. Lund: Lund University Press.
- Svartvik, Jan and Randolph Quirk (1980) *A Corpus of English Conversation*. Lund: Gleerup.
- Wallis, Sean, Bas Aarts, and Gerald Nelson (2000) 'Parsing in reverse: exploring ICE-GB with Fuzzy Tree Fragments and ICECUP'. In: John M. Kirk (ed.) *Corpora Galore: analyses and techniques in describing English*. Amsterdam: Rodopi. 335-344.

