is one utterance for which it took the parser 973 seconds to produce a parse as a result of which the reported average time is much worse than it is for the two other texts. Moreover, the average times obscure the fact that as many as 44 (i.e. 73.2 per cent) of the 60 utterances were actually parsed within 5 seconds (cf. Table 11.10.1).

Finally, in Table 11.5.1 the accuracy of the TOSCA parser has been determined by computing the percentage of the number of instances in which the parser could identify various constructions. Here it should be pointed out that for utterances for which the parser failed to produce an analysis, none of the constructions that occurred in it could contribute towards the success (in other words, if for some reason, the input could not be parsed it was assumed that the identification of all constructions that occurred in it failed).

11.5 Analysis II: Original Grammar, Additional Vocabulary

A separate second analysis in which the system was used with its original grammar and additional vocabulary was, in view of the design of the TOSCA system, not deemed to make any sense. As was explained above, the system requires non-ambiguous and fully correct input, that is, with each token the contextually appropriate tag must be associated in order for the parser to yield the correct analysis. In fact, one can say that the proposed second analysis was already carried out during the first analysis as incorrect tags were being replaced by correct ones.

What would have been interesting in the TOSCA setting is to have the tagger train on texts from this specific domain. As the relative probabilities of the wordform-tag combinations are adjusted, the performance of the tagger for this type of text is bound to improve. However, for lack of time as well as of proper test materials this experiment was not undertaken.

11.6 Analysis III: Altered Grammar, Additional Vocabulary

With regard to the grammar, a number of alterations can be envisaged. Basically these are of two types: alterations that, on implementation, will extend the coverage of the grammar/parser, and alterations that will contribute towards an improvement of the efficiency of the parser. Alterations of the first type are constituted by, for example, a revision or adaptation of the descriptive model, and the formulation of additional rules. Since there appeared to be few omissions in the grammar, and a
revision of the descriptive model is not something that can be carried out overnight, we did not pursue this any further. The efficiency of the parser can be improved by adapting the grammar so that it contains only those rules that are relevant to this specific domain. The yield, however, will be relatively small: the automatic lookahead component already serves to bypass parts of the grammar. One small experiment that we did carry out consisted in removing the syntactic markers from the grammar. The motivation for undertaking this experiment was that the insertion of syntactic markers considerably increases the need for intervention and therefore the amount of manual work one has to put in. Removal of the syntactic markers from the grammar can be viewed as a step towards further automating the system as a whole. Unfortunately, the outcome of the experiment only confirmed the present need for syntactic markers. An attempt at running the parser without requiring any syntactic markers failed: a test run of an adapted version of the parser on the LOTUS text showed the parser to be extremely inefficient so that for a great many utterances a conclusive result could no longer be obtained.

11.7 Converting Parse Tree to Dependency Notation

At the IPSM'95 workshop, it was decided to attempt a translation of the parse of a selected sentence into a dependency structure format for all parsing systems that do not already yield such a format. It was argued that dependency structure analyses are better suited for the calculation of precision and recall measures (cf. Lin, this volume). An attempt to translate the analyses produced by different systems would test the feasibility of a general translation and measurement mechanism. A translation from a constituent structure analysis, however, is certainly not straightforward. One problem is that, in any marked relation between two items in the dependency analysis, one item must be defined as dominating and the other as dependent. For several objects in a constituent structure analysis this is not normally the case, so that rules have to be defined which determine dominance for the translation and which thus add (possibly gratuitous) information. On the other hand, information may be lost since constituent structure analysis allows the binding of information to units larger than a single word, whilst in the translation there is not always a clear binding point for such information. We have attempted to provide a translation of the analysis yielded by the TOSCA system into a rather richer dependency structure analysis than agreed, in order to prevent the loss of information we deem important. The original analysis and the translation of the example sentence
are shown in Figures 11.6 and 11.7.

Since we are not currently interested in the translation other than for the purpose of this experiment, we have not implemented a translation algorithm but have translated the analysis by hand. However, we have tried to adhere to the following general rules. For all headed phrases, the head becomes the dominating element and the other constituents become dependent upon it. For unheaded phrases or clauses, one constituent is chosen to fill the role of the head: the main verb for verb phrases (i.e. auxiliaries, adverbials and main verb), the preposition for prepositional phrases, the main verb (the translation of the verb phrase) for clauses, the coordinator for coordination and the subordinator in subordination. For multi-token words, the last token becomes the dominating element and the preceding tokens become dependent. Each dependency link is marked with an indication of the type of the relation which is derived from the function label of the dependent element at the level of the phrase or clause which is collapsed. In the case of multi-token words, the link
is labelled as a premodifier of whatever the word as a whole represents. In addition, each element by itself is marked with information derived from all constituents in which it dominates.

Apart from extending the format to include extra information, we also deviate from the original proposal in that we assume an additional level dominating the utterance as a whole, called "< discourse >", on which the main elements of the utterance are dependent. Alternatively, we could make punctuation and connectives dependent on the verb or vice versa. Neither alternative, however, is particularly appealing.

Our rules lead to a consistent but not always satisfactory translation. The choice of the dominating element is sometimes arbitrary, as is illustrated in the treatment of "that is". Another problem is constituted by the lack of binding points for information. This is illustrated in "than the TM sentence", where clausal information has to be bound to the subordinator and "the TM sentence" has to be called a subject of "than". Both problems make themselves felt in "longer or shorter", where "or" is chosen as dominating rather than having "or shorter" dependent on "longer" or "longer or" on "shorter". In the current solution "longer" and "shorter" are equal, as we think they should be, but "or" has to be labelled with adjectival information.

From this experience we conclude that the use of this translation is undesirable except for the express purpose of comparison with other systems. The problems that occur within this single example sentence
demonstrate clearly that even a translation for comparison has its pitfalls. Only when the designers of the comparison define very clearly how each construction (e.g. coordination) is to be represented will it be possible for all participants to come to a translation which is comparable. If all information in the translation needs to be compared, it can only contain elements from the greatest common divider of all systems, which means that a lot of richness of the original is lost. We at least would be disappointed if we were to be judged on such a meagre derivative of our analyses.

11.8 Summary of Findings

If we take the three texts under investigation to be representative of the domain of computer/software manuals, application of the TOSCA system to this domain can, on the whole, be considered successful. There were not too many problems in getting the parser to parse the utterances. The coverage of the grammar proved satisfactory: only few structures were encountered that pointed to omissions in the grammar. While we had expected the grammar to fall short especially with structures that so far in our own test materials had been relatively underrepresented (such as imperatives), this was not the case.

The experiment once more confirmed what earlier experiences with other types of text had already brought to light: the system clearly shows the effects of a compromise between efficiency and robustness on the one hand, and coverage and detail on the other. As we pointed out above, both efficiency and robustness are considered to be only of secondary importance. In the design of the system linguistic descriptive and observational adequacy are taken to be the primary objectives. However, practical considerations force us to somehow optimize the analysis process. This explains the present need for intervention at various points in the process, viz. tag selection, syntactic marking and selection of the contextually appropriate analysis. As a result, operation of the present system is highly labour-intensive.

Judging from our experiences in carrying out the experiment it would appear that the performance of the TOSCA system when applied to a specific (restricted) domain could be improved by a number of domain-specific adjustments to some of its components. For example, the development and implementation of a domain-specific tokenizer and tagger would yield a better performance of the tagger and reduce the need for tag correction. The efficiency of the parser might be increased by restricting the grammar to the specific domain. While the present grammar/parser seeks to cover the whole of the English language and must allow for all possible structural variation, a parser derived from a domain-
specific grammar could be much more concise and far less ambiguous. As for the automatic filter component that is used to automatically filter out the intuitively most probable analysis, this appears to be domain-dependent: while usually in running text the sentence analysis of an utterance is intuitively more likely than its analysis as a phrase or clause, in instructive types of text this is not necessarily true.

11.9 References


Appendix I

The 60 IPSM Test Utterances
Dynix Test Utterances

If your library is on a network and has the Dynix Gateways product, patrons and staff at your library can use gateways to access information on other systems as well.

For example, you can search for items and place holds on items at other libraries, research computer centers, and universities.

Typically, there are multiple search menus on your system, each of which is set up differently.

The search menu in the Circulation module may make additional search methods available to library staff.

For example, an alphabetical search on the word "Ulysses" locates all titles that contain the word "Ulysses."

Displays the records that have a specific word or words in the TITLE, CONTENTS, SUBJECT, or SERIES fields of the BIB record, depending on which fields have been included in each index.

For example, you can use an accelerated search command to perform an author authority search or a title keyword search.

The search abbreviation is included in parentheses following the search name:

A system menu

Any screen where you can enter "SO" to start a search over

Certain abbreviations may work at one prompt but not at another.

To perform an accelerated search, follow these instructions:

That item's full BIB display appears.

Write down any information you need, or select the item if you are placing a hold.

Alphabetical Title Search.

Enter the line number of the alphabetical title search option.

A BIB summary screen appears, listing the titles that match your entry.

When you access the BIB record you want, you can print the screen, write down any information you need, or select the item if you are placing a hold.

The cursor symbol (>) appears on the alphabetical list next to the heading that most closely matches your request.

Byzantine empire
Lotus Test Utterances

When you are editing a document, you want to be able to move quickly through the pages.

Scrolling changes the display but does not move the insertion point.

To use keyboard shortcuts to navigate a document

Move the mouse pointer until the I-beam is at the beginning of the text you want to select.

For information, refer to "Undoing one or more actions" in this chapter.

Ami Pro provides three modes for typing text.

In Insert mode, you insert text at the position of the insertion point and any existing text automatically moves.

If you press BACKSPACE, Ami Pro deletes the selected text and one character to the left of the selected text.

You can disable Drag & Drop.

If you want to move the text, position the mouse pointer anywhere in the selected text and drag the mouse until the insertion point is in the desired location.

The contents of the Clipboard appear in the desired location.

To move or copy text between documents

Choose Edit/Cut or Edit/Copy to place the selected text on the Clipboard.

If the document into which you want to paste the text is already open, you can switch to that window by clicking in it or by choosing the Window menu and selecting the desired document.

Press SHIFT+INS or CTRL+V.

Select the text you want to protect.

Permanently inserts the date the current document was created.

You can select Off, 1, 2, 3, or 4 levels.

When you want to reverse an action, choose Edit/Undo.

Modifying the Appearance of Text
Trados Test Utterances

The following are suggestions on how to proceed when using the Translator’s Workbench together with Word for Windows 6.0.

Another important category of non-textual data is what is referred to as "hidden text."

Alternatively, choose the Options... menu item from Word’s Tools menu.

Thus, you will make sure that you see all the information that the Workbench manages during translation:

Always put one abbreviation on a line, followed by a period.

During the translation of this example, the Workbench should ignore the second sentence when moving from the first sentence to the third one.

In Word, you can immediately recognize a 100% match from the color of the target field.

The TWB1 button, also labeled Translate Until Next Fuzzy Match, tells the Workbench to do precisely this.

You can also use the shortcut [Alt] + [x] on the separate numeric keypad to start this function.

That is, these words make the source sentence longer or shorter than the TM sentence.

Likewise, if something has been left out in the source sentence, you will have to delete the corresponding parts in the suggested translation as well.

Automatic Substitution of Interchangeable Elements

If the Workbench cannot find any fuzzy match, it will display a corresponding message ("No match") in the lower right corner of its status bar and you will be presented with an empty yellow target field.

Then go on translating until you want to insert the next translation.

Select the text to be copied in the Concordance window, usually the translation of the sentence part that you have searched for.

The same goes for formatting:

Making Corrections

If you would like to make corrections to translations after their initial creation, you should always do this in TM mode so that the corrections will be stored in Translation Memory as well as in your document.

But consider the following example where text is used within an index entry field:
If a perfect or fuzzy match is found, the Workbench will again automatically transfer its translation into the target field in WinWord.
Appendix II

Sample Parser Outputs

The following pages contain the output from the eight parsing systems for five test utterances drawn from the set of 60 in Appendix I. Very slight modifications had to be made to the format of some parse trees to allow them to fit on the page. Those trees affected are marked with a star.
Enter the line number of the alphabetical title search option.

("Enter" "the" "line" "number" "of" "the" "alphabetical" "title" "search" "option")

; ; ; GC: 292490 words [1169960 bytes] of dynamic storage in use.
; ; ; 1665260 words [665040 bytes] of free storage available before a GC.
; ; ; 625010 words [2500040 bytes] of free storage available if GC is
; ; ; disabled.

Parsing time: 34.832098 seconds.
String length: 10
Chart size: 9
spanning edges are: NIL
Chart size: 9

Fragment No. 1
>From 0 To 1
(NOUN "Enter")

Fragment No. 2
>From 1 To 4
(NP (DET "the") (NOUN (ADJ "line") (NOUN "number")))

Fragment No. 3
>From 4 To 10
(NP-MOD (NP-COMJ "of")
(NP (DET "the")
(NOUN (ADJ "alphabetical")
(NOUN (ADJ "title") (NOUN (ADJ "search") (NOUN "option")))))
ENGCG: D20 (Analysis I)

Enter the line number of the alphabetical title search option.

"<Enter>"
"<the>" "the" <Def> DET CENTRAL ART SG/PL @DM>
"<line>"
"line" @ NOM SG @M>
"<number>"
"number" @ NOM SG/PL @B>
"<of>"
"of" PREP @#OF
"<the>"
"the" <Def> DET CENTRAL ART SG/PL @DM>
"<alphabetical>"
"alphabetical" <DER:ic> <DER:al> A ABS @A>
"<title>"
"title" @ NOM SG @M> @P
"<search>"
"search" @ NOM SG @M>
"<option>"
"option" @ NOM SG @P
"<.>"
LPARSER: D20 (Analysis I)

Enter the line number of the alphabetical title search option.

Accepted
Unique linkage, cost vector = (0, 0, 13)

```
+-----------------------------+-----
 | +-----------------------------+-----
 | +-----------------------------+-----
 +-----------------------------+-----
 | +-----------------------------+-----
 | +-----------------------------+-----
 +-----------------------------+-----
 | +-----------------------------+-----
 | +-----------------------------+-----
 +-----------------------------+-----
```

Enter the line number of the alphabetical title search option
PRINCIPAR: D20

Enter the line number of the alphabetical title search option.

;; time: 0.3 sec.
;; Fragments: 1
(S
  (VP (Vbar (V (V[IP]
    (V[IP] Enter)
  )
  (IP
    (Det the)
    (Nbar
      (N line number)
      (PP (Pbar (P
        (P of)
      )
      (IP
        (Det the)
        (Nbar
          (AP (Abar (A alphabetical)))
          (N title)
          (N search option))))))))
))
RANLT: D20 (Analysis I)

Enter the line number of the alphabetical title search option.

(VP/NP enter

(N/2+/DET) a the

(P1/PPMOD (N/1/ (N/COMPOUND line number))

(P2/NP a of

(N/2+/DET) a the

(P2-N

(N/COMPOUND alphabetical

(N/COMPOUND title (N/COMPOUND search option)))))))))))
SEXTANT: D20 (Analysis I)*

Enter the line number of the alphabetical title search option.

162 VP 101 Enter enter INF 0 0
162 NP 2 the the DET 1 1 3 (number) DET
162 NP 2 line line NOUN 2 1 3 (number) NN
162 NP* 2 number number NOUN 3 1 0 (enter) DOBJ
162 NP 2 of of PREP 4 1 9 (option) PREP
162 NP 2 the the DET 5 1 9 (option) DET
162 NP 2 alphabetical alphabetical ADJ 6 2 7 (title) ADJ 9 (option) ADJ
162 NP 2 title title NOUN 7 2 8 (search) NN 9 (option) NN
162 NP 2 search search NOUN 8 1 9 (option) NN
162 NP* 2 option option NOUN 9 2 3 (number) NN PREP 0 (enter) IOBJ-of
162 -- 0 . . 10 0
DESPAR: D20 (Analysis I)

Enter the line number of the alphabetical title search option.

```
#016
VB  enter 1  --> 11  -
DT  the 2  -->  4  [
NM  line 3  -->  4  +
NM  number 4  -->  1  OBJ
IN  of 5  -->  4  ]
DT  the 6  -->  10  [
JJ  alphabetical 7  -->  8  +
NM  title 8  -->  9  +
NM  search 9  -->  10  +
NM  option 10  -->  5  +
.  .  .  11  -->  0  ]
```
TOSCA: D20 (Analysis I)*

Enter the line number of the alphabetical title search option.

1 analysis in 2 seconds with TOSCA=1CE/W0.3.950102

Enter
ALICE: D22*

Displays the records that have a specific word or words in the TITLE, CONTENTS, SUBJECT, or SERIES fields of the BIB record, depending on which fields have been included in each index.

("Displays" "the" "records" "that" "have" "a" "specific" "word" "or" "words" "in" "the" "TITLE" "," "CONTENTS "," "SUBJECT "," "or" "SERIES" "fields" "of" "the" "BIB" "record" "," "depending" "on" "which" "fields" "have" "been" "included" "in" "each" "index")

;; GC: 297762 words [191048 bytes] of dynamic storage in use.
;; 160988 words [643962 bytes] of free storage available before a GC.
;; 619738 words [2470952 bytes] of free storage available if GC is disabled.

Parsing time: 101.074216 seconds.
String length: 36
Chart size: 267
spanning edges are: NIL
Chart size: 267

Fragment No. 1
>From 0 To 1
(NP "Displays")
Fragment No. 2
>From 1 To 3
(NP (DET "the") (NOUN "records"))
Fragment No. 3
>From 3 To 4
(DET "that")
Fragment No. 4
>From 4 To 5
(UNK-CAT "have")
Fragment No. 5
>From 5 To 8
(POSTMOD (UNK-CAT "a") (NOUN (ADJ "specific") (NOUN "word")))
Fragment No. 6
>From 8 To 10
((NOUN MOD) (UNK-CAT "or") (NOUN "words"))
Fragment No. 7
>From 10 To 11
(PREP "in")
Fragment No. 8
>From 11 To 12
(DET "the")
Sample Parser Outputs

Fragment No. 9
>From 12 To 13

(*IP "TITLE")
Fragment No. 10
>From 13 To 14

(*UNK-CAT ",")
Fragment No. 11
>From 14 To 15

(*IP "CONTENTS")
Fragment No. 12
>From 15 To 16

(*UNK-CAT ",")
Fragment No. 13
>From 16 To 17

(*IP "SUBJECT")
Fragment No. 14
>From 17 To 18

(*UNK-CAT ",")
Fragment No. 15
>From 18 To 21

((NOUN MOD) (UNK-CAT "or") (NOUN (ADJ "SERIES") (NOUN "fields")))
Fragment No. 16
>From 21 To 25

(*NP-MOD (MP-COMJ "of")
 (*IP (DET "the") (NOUN ((NOUN MOD) "BIB") (NOUN "record")))
Fragment No. 17
>From 25 To 36

(*UNK-CAT ",")
Fragment No. 18
>From 36 To 27

(PRESP "depending")
Fragment No. 19
>From 27 To 28

(*UNK-CAT "on")
Fragment No. 20
>From 28 To 31

(*UNK-CAT (MP (DET "which") (NOUN "fields") (UNK-CAT "have"))
Fragment No. 21
>From 31 To 33

(*VP-PASS (AUX "been") (PPART "included"))
Fragment No. 22
>From 33 To 36
(SENT-MOD (UNK-CAT "in") (NP (DET "each") (NOUN "index")))
ENGCG: D22 (Analysis I)

Displays the records that have a specific word or words in the TITLE, CONTENTS, SUBJECT, or SERIES fields of the BIB record, depending on which fields have been included in each index.

"<Displays>
"display" <SVO> V PRES SG VFIN @+FMAINV
"<the>
"the" <Def> DET CENTRAL ART SG/PL @DBJ
"<records>
"record" @ NOM PL @OBJ
"<that>
"that" @<ComMod> @<CLB> @Rel> PROM SG/PL @SUBJ
"<have>
"have" <SVO> <SVO/A> V PRES -SG3 VFIN @+FMAINV
"<a>
"a" <Indef> DET CENTRAL ART SG @DBJ
"<specific>
"specific" <DEr:ic> A ABS @AN
"<word>
"word" @ NOM SG @OBJ
"<or>
"or" CC @CC
"<words>
"word" @ NOM PL @OBJ
"<in>
"in" PREP @<Nom @ADVJ
"<the>
"the" <Def> DET CENTRAL ART SG/PL @DBJ
"<TITLE>
"title" @ NOM SG @P
"<\,>
"<CONTENTs>
"content" @ NOM PL @SUBJ @APP @P @<Nom
"<SUBJECT>
"subject" @ NOM SG @APP @P @<Nom
"<\,>
"<or>
"or" CC @CC
"<SERIES>
"series" @ NOM SG/PL @DBH
"<field>
"field" @ NOM PL @SUBJ @APP @P
"<of>
"of" PREP @<Nom @OF
"<the>
"the" <Def> DET CENTRAL ART SG/PL @DBJ
"<BIB>
"bib" @ NOM SG @DBH
"<record>
"record" <SVO> V PRES -SG3 VFIN @+FMAINV
"record" @ NOM SG @P
"<§,>
"<depending>"
"<depend" <SV> PCP1 @-FMAINV
"<on>"
"on" PREP @ADVL
"<which>"
"which" DET CENTRAL WH SG/PL @DM>
"<fields>"
"field" N NOM PL @SUBJ
"<have>"
"have" <SVO> <SVOC/A> V PRES -SG3 VPFIN @+FAUXV
"<be>"
"be" <SVC/A> <SVC/A> PCP2 @-FAUXV
"<included>"
"include" <SVD> @/in> PCP2 @-FMAINV
"<in>"
"in" PREP @ADVL
"<each>"
"each" <Quant> DET CENTRAL SG @QM>
"<index>"
"index" N NUM SG @P
"<§,>"
LPARSER: D22 (Analysis I)

Displays the records that have a specific word or words in the TITLE, CONTENTS, SUBJECT, or SERIES fields of the BIB record, depending on which fields have been included in each index.

Accepted (4424 linkages, 192 with no P.P. violations)
Linkage 1, cost vector = (0, 1, 72)

+-----------------------X-----------------------+
| +-----------------------+                        |
| |                       |
| |                       |
| |                       |
+-----------------------+                        |

/+//+/ it displays the records that have a specific word or words in the TITLE, CONTENTS, SUBJECT, or SERIES fields of the BIB record, depending on which fields have been included in each index.
PRINCIPAR: D22

Displays the records that have a specific word or words in the TITLE, CONTENTS, SUBJECT, or SERIES fields of the BIB record, depending on which fields have been included in each index.
(VP (Vbar (V (V_CP depending on))
 (CP
  (NP[2]
   (Det which)
   (Vbar (V fields)))
   (Ibar (IP
    (Ibar
     (Have have)
     (Be been)
     (VP (Vbar
      (V (V_CP included)
      t[2])))
      (PP (Pbar (P
       (P in)
       (P
        (Det each)
        (Vbar (V index))))))
      ))))))))))))))))))))))))))}))

.)
RANLT: D22 (Analysis I)*

Displays the records that have a specific word or words in the TITLE, CONTENTS, SUBJECT, or SERIES fields of the BIB record, depending on which fields have been included in each index.

\[
\begin{align*}
\text{(11/PPMOD) (11/BELMOD1 (11/ displays) (S/THATLESSREL) (Stb (12+/DET1a the (12- (11/ records))) (A2/ADVMOD1/- (A1/A that)) (VP/MOD1 (VP/MP have (TRACEE1 E))) (X2/NMOD3 (12+/PARTS (12+/1PR0a (11/PPMOD (11/COORD2A (11/PP a) (11/AMOD1 (A2/ADVMOD1/- (A1/A specific)) (11/ word)) (CONJ/1 or (11/ words)))) (P2/P1 (P1/MP a in (12+/APP05/2 (12+/DET1a the (12+/COORD3A (12- (11/BELMOD1 (11/ title) (S/THATLESSREL (Sta (12+/W2-a (12- (11/ contents))) (VP/MP subject (TRACEE1 E)))))) (CONJ/2 or (12- (11/ series))))) (12+/PW fields)))) of (12+/ADJ1 the (A2/ADVMOD1/- (A1/A (A/COMPOUNDB bib record)))))))) (VP/INTR depending) (S/ADVBla (12+/MOD2 (P2/P1 (P1/MPb on (12+/PRO which)))) (S2 (12+/W2-a (12- (11/ fields))) (VP/HAVE have (VP/BE_PRD been (PRD2 (A2/ADVMOD1/- (A1/A included))))))))) (P2/P1 (P1/MP a in (12+/QUA (A2/ADVMOD1/- (A1/A each)) (12- (11/ PW index))))))]
\end{align*}
\]
SEXTANT: D22 (Analysis I)*

Displays the records that have a specific word or words in the TITLE, CONTENTS, SUBJECT, or SERIES fields of the BIB record, depending on which fields have been included in each index.
DESPAR: D22 (Analysis I)

Displays the records that have a specific word or words in the TITLE, CONTENTS, SUBJECT, or SERIES fields of the BIB record, depending on which fields have been included in each index.

VBZ displays 1 --> 37 -
DT the 2 --> 3 [
NNS records 3 --> 1 +OBJ
WDT that 4 --> 3 "
VBZ have 5 --> 1 ]
DT a 6 --> 8 [
JJ specific 7 --> 8 +
NNS word 8 --> 5 +OBJ
CC or 9 --> 6 ]
NNS words 10 --> 9 [
IN in 11 --> 8 ]
DT the 12 --> 13 [
NNS title 13 --> 11 +
, , 14 --> 13 ]
NNS contents 15 --> 14 [
, , 16 --> 15 ]
NNS subject 17 --> 16 [
, , 18 --> 17 ]
CC or 19 --> 17 -
NNS series 20 --> 21 [
NNS fields 21 --> 19 +
IN of 22 --> 21 ]
DT the 23 --> 25 [
NNS bib 24 --> 26 +
NNS record 25 --> 22 +
, , 26 --> 28 ]
VBG depending 27 --> 5 -
IN on 28 --> 27 -
WDT which 29 --> 30 [
NNS fields 30 --> 20 +
VBZ have 31 --> 33 ]
VBZ been 32 --> 33 -
VBZ included 33 --> 27 -
IN in 34 --> 33 -
DT each 35 --> 30 [
NNS index 36 --> 34 +
, , 37 --> 0 ]
TOSCA: D22 (Analysis I)*

Displays the records that have a specific word or words in the TITLE, CONTENTS, SUBJECT, or SERIES fields of the BIB record, depending on which fields have been included in each index.

<tpar>
47 analyses in 93 seconds with TOSCA-ICE/W0.3.950102
</tpar>

<tparn fun=- cat=TXTU>
<tparn fun=UTT cat=SU att=(su,act,decl,indic,motr,pres,unm)
<tparn fun=VP cat=VP att=(act,indic,motr,pres)
<tparn fun=MVB cat=MV att=(indic,motr,pres)> Displays
</tpar></tpar>

<tparn fun=OD cat=NP>
<tparn fun=DT cat=DTP>
<tparn fun=DTCE cat=ART att=(def)> the
</tpar></tpar>

<tparn fun=NPPO cat=CL att=(act,indic,motr,pres,rel,umr)
<tparn fun=SU cat=NP>
<tparn fun=MVB cat=MV att=(indic,motr,pres)> have
</tpar></tpar>

<tparn fun=OD cat=NP>
<tparn fun=DT cat=DTP>
<tparn fun=DTCE cat=ART att=(def)> a
</tpar></tpar>

<tparn fun=NPPO cat=CL att=(act,indic,motr,pres,rel,umr)
<tparn fun=SU cat=NP>
<tparn fun=MVB cat=MV att=(indic,motr,pres)> specific
</tpar></tpar>

<tparn fun=NPPO cat=CL att=(act,indic,motr,pres,rel,umr)
<tparn fun=SU cat=NP>
<tparn fun=MVB cat=MV att=(indic,motr,pres)> word
</tpar></tpar>

<tparn fun=NPPO cat=CL att=(act,indic,motr,pres,rel,umr)
<tparn fun=SU cat=NP>
<tparn fun=MVB cat=MV att=(indic,motr,pres)> words
</tpar></tpar>

<tparn fun=A cat=PP>
<tparn fun=P cat=PREP> in
</tpar></tpar>

<tparn fun=NP cat=NP>
<tparn fun=DT cat=DTP>
<tparn fun=DTCE cat=ART att=(def)> the
</tpar></tpar>

<tparn fun=MVB cat=MV att=(indic,motr,pres)> TITLE, CONTENTS, SUBJECT, or SERIES fields
</tpar></tpar>

<tparn fun=NPPO cat=CL att=(act,indic,motr,pres,rel,umr)
<tparn fun=SU cat=NP>
<tparn fun=MVB cat=MV att=(indic,motr,pres)> of
</tpar></tpar>
depending on which fields have been included in each index.
ALICE: L20

Press SHIFT+INS or CTRL+V.

("Press_SHIFT" "+" "INS" "or" "CTRL" "+" "V")

; ; ; GC: 341468 words [1365972 bytes] of dynamic storage in use.
; ; ; 117282 words [469128 bytes] of free storage available before a GC.
; ; ; 570032 words [2304128 bytes] of free storage available if GC is
; ; ; disabled.

Parsing time: 30.606719 seconds.
String length: 7
Chart size: 99
spanning edges are: NIL
Chart size: 99

Fragment No. 1
>From 0 To 3

(HP (HP-MOD "Press_SHIFT") (HP (HP-MOD "+") (HP "INS")))

Fragment No. 2
>From 3 To 7

(HP-MOD (HP-CODE "or") (HP (HP-MOD "CTRL") (HP (HP-MOD "+") (HP "V"))))
ENGCG: L20 (Analysis I)

Press SHIFT+INS or CTRL+V.

"<Press>"
"press" <-> <SVO> <SV> V IMP VFIM @FMAINV
"<SHIFT+INS>"
"SHIFT+INS" ? > <NoBaseformNormalisation> NOM SG/PL @OBJ
"<or>"
"or" CC @CC
"<CTRL+V>"
"CTRL+V" ? > <No NOM SG @OBJ
"<§>"
LPARSER: L20 (Analysis I)

Press SHIFT+INS or CTRL+V.

Accepted

Unique linkage, cost vector = (0, 0, 1)

+-------@-------+
| +---ψ----+     +-------+-----+
|       |       |       |
///// press v SHIFT+INS or CTRL+v
PRINCIPAR: L20

Press SHIFT+INS or CTRL+V.

;; time: 0.05 sec.
;; Fragments: 1
(S
  (VP (Vbar (V (V NP
    (V NP Press)
    (NP
      (NP (Nbar ([# SHIFT+INS]))
      or
      (NP (Nbar ([# CTRL+V]))))))))
.)
RANLT: L20 (Analysis I)

Press SHIFT+INS or CTRL+V.

(VP/NP press
  (N/2/COORD (N/2/-a (N/1/ shift ins)))
  (COORD/2 or (N/2/-a (N/1/ ctrl v))))
SEXTANT: L20 (Analysis I)*

Press SHIFT+INS or CTRL+V.

<table>
<thead>
<tr>
<th>93</th>
<th>VP</th>
<th>101</th>
<th>Press</th>
<th>press</th>
<th>INF</th>
<th>0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>93</td>
<td>MP*</td>
<td>2</td>
<td>SHIFT-INS</td>
<td>SHIFT-INS</td>
<td>NOUN</td>
<td>1 1 0 (press) DOBJ</td>
</tr>
<tr>
<td>93</td>
<td>MP</td>
<td>2</td>
<td>or</td>
<td>or</td>
<td>CC</td>
<td>2 0</td>
</tr>
<tr>
<td>93</td>
<td>MP*</td>
<td>2</td>
<td>CTRL-V</td>
<td>CTRL-V</td>
<td>NOUN</td>
<td>3 1 0 (press) DOBJ</td>
</tr>
<tr>
<td>93</td>
<td>--</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>4 0</td>
</tr>
</tbody>
</table>
DESPAR: L20 (Analysis I)

Press SHIFT+INS or CTRL+V.

 VB  press 1  --> 5  
 MP  shift+ins 2  --> 1  [ OBJ
 CC  or 3  --> 2  ]
 MP  ctrl+4  --> 3  [  
 5  --> 0  ]
TOSCA: L20 (Analysis I)

Press SHIFT+INS or CTRL+V.

<tparc>
2 analyses in 2 seconds with TOSCA-ICE/W0.3.950102
</tparc>

<tparn fun=- cat=TEXT>
<tparn fun=UTT cat=S att=(-su,act,imper,motr,pres,umn))>
<tparn fun=VP cat=VP att=(act,imper,motr,pres))>
<tparn fun=VB cat=V att=(imper,motr,pres))>
</tparn></tparn>

<tparn fun=ID cat=COORD>
<tparn fun=CJ cat=NP>
<tparn fun=PHD cat=# att=(prop,sing))>
</tparn></tparn>
<tparn fun=DM cat=COORD att=(coord))>
</tparn></tparn>

<tparn fun=PHD cat=# att=(prop,sing))>
</tparn></tparn>
<tparn fun=PUNC cat=PM att=(per))>
</tparn>
ALICE: L22*

Select the text you want to protect.

("Select" "the" "text" "you" "want" "to" "protect")

; ; ; GC: 311968 words [1247672 bytes] of dynamic storage in use.
; ; ; 146762 words [567128 bytes] of free storage available before a GC.
; ; ; 605552 words [2422128 bytes] of free storage available if GC is
; ; ; disabled.

Parsing time: 30.436594 seconds.
String length: 7
Chart size: 86
spanning edges are: NIL
Chart size: 86

Fragment No. 1
>From 0 To 5

(SENT (SENT-MOD (UNK-CAT "Select") (NP (DET "the") (NOUN "text")))))

(SENT (VP-ACT (NP "you") (V-TR "want") (NP NULL-PHON)))

Fragment No. 2
>From 5 To 7

(SENT-MOD (UNK-CAT "to") (NP "protect"))
ENGC: L22 (Analysis I)

Select the text you want to protect.

"<Select>"
"<select> ↔ <SVO> <SV> $/for$ V IMP VFIN @+FMAINV
"<the>"
"the" <Def> DET CENTRAL ART SG/PL @DM>
"<text>"
"text" ∈ NOM SG @OBJ
"<you>"
"you" <NonMod> PRON PERS NOM SG2/PL2 @SUBJ
"<want>"
"want" <SVOCA/> <SVO> <P/for> V PRES -SG3 VFIN @+FMAINV
"<to>"
"to" INFMARK @INFMARK
"<protect>"
"protect" <SVO> V INF @-FMAINV
"<$.>"
LPARSER: L22 (Analysis I)

Select the text you want to protect.

Accepted
Unique linkage, cost vector = (0, 0, 4)

+-----0-+--------+-+-+
|       |       |       |
|       |       |       |

//////// select the text you want to protect.
PRINCIPAR: L22

Select the text you want to protect.

; ; time: 0.13 sec.
; ; Fragments: 1
(S
  (VP (Vbar (V (V_MP
    (V_MP Select)
    (MP
      (Det the)
      (Nbar
        (N text)
        (CP
          (Op[
            (CP (IP
              (MP (Nbar (N you)))
            (Ibar (VP (Vbar (V (V_CP
              (V_CP want)
              (CP (Cbar (IP
                (Ibar
                  (Aux to)
                  (VP (Vbar (V (V_HP
                    (V_HP protect)
                    t[1]))))))))))))))))))))))}}){
RANLT: L22 (Analysis I)

Select the text you want to protect.

(VP/NP select
 (N/DET a the
  (N/INFMOD (N/RELMOD (N/REL (N/text)))))
(S/THATLESSREL (S/a (N/PRO you) (VP/NP want (TRACE E))))
(VP/TO to (VP/NP protect (TRACE E))))
SEXTANT: L22 (Analysis I)*

Select the text you want to protect.

134 VP 101 Select select INF 0 0
134 NP 2 the the DET 1 1 2 (text) DET
134 NP* 2 text text NOUN 2 1 0 (select) SBJ
134 NP* 3 you you PRON 3 0
134 VP 102 want want INF 4 0
134 VP 102 to to TO 5 0
134 VP 102 protect protect INF 6 1 3 (you) SBJ
134 -- 0 . . . 7 0
DESPAR: L22 (Analysis I)

Select the text you want to protect.

```
VB  select 1 --&gt; 8 -
DT  the 2 --&gt; 3 [  
NN  text 3 --&gt; 1 + OBJ  
Pp  you 4 --&gt; 5 " SUB  
VBp  want 5 --&gt; 3 ]
To  to 6 --&gt; 7 -
VB  protect 7 --&gt; 5 -
  .  8 --&gt; 0 -
```
TOSCA: L22 (Analysis I)

Select the text you want to protect.

Cannot be parsed due to raising.