Flexible Language Analysis Tools

FLAT

User Manual

LanA Consulting
Madvigs Allè, 9, 2
DK-1829 Copenhagen, Denmark
Tel.: +45 33 25 04 41
Fax: +45 33 22 38 22
e-mail: lanaconsult@mail.dk
www.lanaconsult.com

Content

What is FLAT

FLAT stands for **Flexible Language Analysis Tool,** and is a multipurpose interactive tool for developing NLP systems and/or training computational linguists

FLAT includes the following programs:

1. ***Lexicon Creator,***
2. ***Tagger (look-up and disambiguation)***
3. ***Interactive Tag Disambiguation Interpreter,***
4. ***FLAT Control*** (an invisible Active X Control, based on the MFC library) ***for integration with a foreign application,***

And sample patent domain knowledge:

1. ***Patent domain Lexicon***
2. ***Rules for Tag Disambiguation.***

FLAT

- can be used for any language based on ANSI character set without reengineering
- can easily be integrated in any NLP application, by thus dramatically reducing the complexity and costs of producing multilingual applications.
- is equipped with control and interactive interfaces for updating linguistic knowledge and tracing processing steps and does not require programming skills to create and experiment with different depths (and sizes) of lexical and grammatical knowledge.

This version of FLAT is a 32-bit Windows application developed to run in a number of operating environments: Windows 95/98/Me or Windows NT 4.0/2000/XP.

Technical support

LanA Consulting, an IT company located in Copenhagen, Denmark, reserves all rights for the FLAT application. All registered users will receive up-to-date information on new versions of this program as well as full support for this system (consulting, training, versions upgrade, FAQ answers etc.).

Contact address:

Lana Consulting, Madvigs alle 9, 1829 Copenhagen, Denmark
Ph.: +45 33250441
Fax: +45 22332822
E-mail: lanaconsult@mail.dk
URL: http://www.lanaconsult.com

System requirements

Your system must fulfill the following minimum requirements if you want to run FLAT:

PC: Pentium processor, 32 MB RAM, 100MB free hard disk space, CD-ROM drive
Operating system: Windows 95/98/Me or Windows NT 4.0/2000/XP

Installation

To install FLAT on your personal computer, proceed as follows:

1. **Start the setup program:** Insert the CD-ROM in the appropriate drive and start the **setup.exe** program on the CD-ROM. To do this, choose the **Run** command on the Start menu and type the following in the command line: **d:setup.exe** (replacing d: with the letter for your CD-ROM drive, if different).
2. **Select the installation directory:** Confirm the suggested installation location (default: C: \Program Files\FLAT). You may also choose a different directory.
3. **Select the Program Folder:** Confirm the suggested Program Folder (default: FLAT). You may type a new folder name, or select one from the existing Folder list.
4. **Run FLAT:** Click on the icon of FLAT in Program Folder or on the icon on the Desktop, or check the check box "Yes, Launch the program file".
5. **Register:** After you ran the program FLAT, you will get a pop-up window in which you will see your **user code** and the **serial number** of FLAT. You would be suggested to enter your name and **registration key.** To get your registration key please **mail** your serial number and your user code to ***lanaconsult@mail.dk***. We will reply your message.

Uninstallation

If you want to uninstall FLAT, proceed as follows:

1. Open the dialog box **Add/Remove Program Properties** by selecting **Settings-Control Panel-Add/Remove Programs** from the Windows Start menu.
2. Select FLAT for removal.

Registration

Run FLAT, after you ran the program FLAT, you will get a pop-up window in which you will see your **user code** and the **serial number** of FLAT. You will be suggested to enter your name and **registration key.**

To get your registration key please **e-mail** your serial number and your user code to ***lanaconsult@mail.dk***. We will send you your registration key in reply to your message.

Starting and resuming FLAT sessions

To start a FLAT session for the first time double click on the Lexicon Creator icon. You should have a Flat lexicon created before using the FLAT Tagger.

To resume a Flat session double click either on the Lexicon Creator icon or the Tagger icon depending upon your task.

Quitting FLAT and saving files

Before quitting FLAT Lexicon Creator do not forget to save your lexicon as **a lexicon file** to be able to resume the FLAT session to work with this lexicon later. To save your lexicon as a lexicon file select options "File -> Save" or "File -> Save as..." . By default the lexicon you created will be saved in the "Lexicons" subdirectory of the FLAT directory.

Trouble shooting

- If you get an unexpected system error message, which might be caused by misuse of the program click "OK" and continue working. In case it does not help exit FLAT and open it again.

Lexicon Creator

Functionalities

FLAT Lexicon Creator is a program for lexicon acquisition that allows for different depth descriptions of lexical items in any language based on ANSI character set. The entry of the lexicon contains two fields, the name of the lexeme and a tag or supertag that can code more knowledge than a regular part-of-speech. The basic principle for this tool is that the user can easily update both the list of words and tags (supertags) making the tag set larger or smaller in number and as "shallow" or "deep" as required. See a sample set of supertags in the section "FLAT Patent domain knowledge").

Lexicon Creator is pipelined to the FLAT tagger (see the section Look-up Tagger), so as to automatically import words that were not recognized by the system after tagging a certain amount of text. The coverage of the lexicon thus improves incrementally.

With Lexicon Creator you can do the following.

- Create lexicons of any size from scratch, customizing them as necessary and save them in files
- Update an existing lexicon
    - Add new lexical items without duplication the same lexemes with the same codes by
        - Importing them from external text file
        - Pasting words recognized as unknown from the tagger
        - One by one, manually
    - Edit lexical items
    - Remove lexical items (one by one or in groups)
    - Add, edit or remove code tags
    - Assign/Edit code tags for lexemes (one by one or in groups)
    - Search any lexical item
    - Look through the word list by
        - Using the scroll bar (you can only scroll 10,000 lexical items)
        - Using the "Back" or "Forward" buttons that make it possible to scroll every next/previous 10,000 lexical items in turn
    - Filter lexical items
        - By prefixes, suffixes, code tags
        - Without code tags (e.g., newly imported uncoded words)
    - Undo an action

Interface

The Lexicon Creator **interface** displaying a sample lexicon and set of code tags is shown in Figure 1. The main menu in the right top corner of the interface has "File", "Edit", "Language", and "Configure" selections. You will be explained how to use these in the following sections. The left pane shows an interactive "Find" window, the buttons "New", "Undo" and "Delete" (for updating a word list) and a scrollable list of lexical units, including multiword prepositions, adverbs, idiomatic phrases, etc. Under the list of words there are two buttons "Back" and "Forward" which will be enabled in case your lexicon contains more than 10,000 units. The right pane contains an interactive editing window and a number of code tags next to check boxes. A checked box indicates a code tag assigned to a highlighted word. At the very bottom of the interface there are two status bars. One bar shows the location and name of your lexicon, another displays the total number of lexical entries.

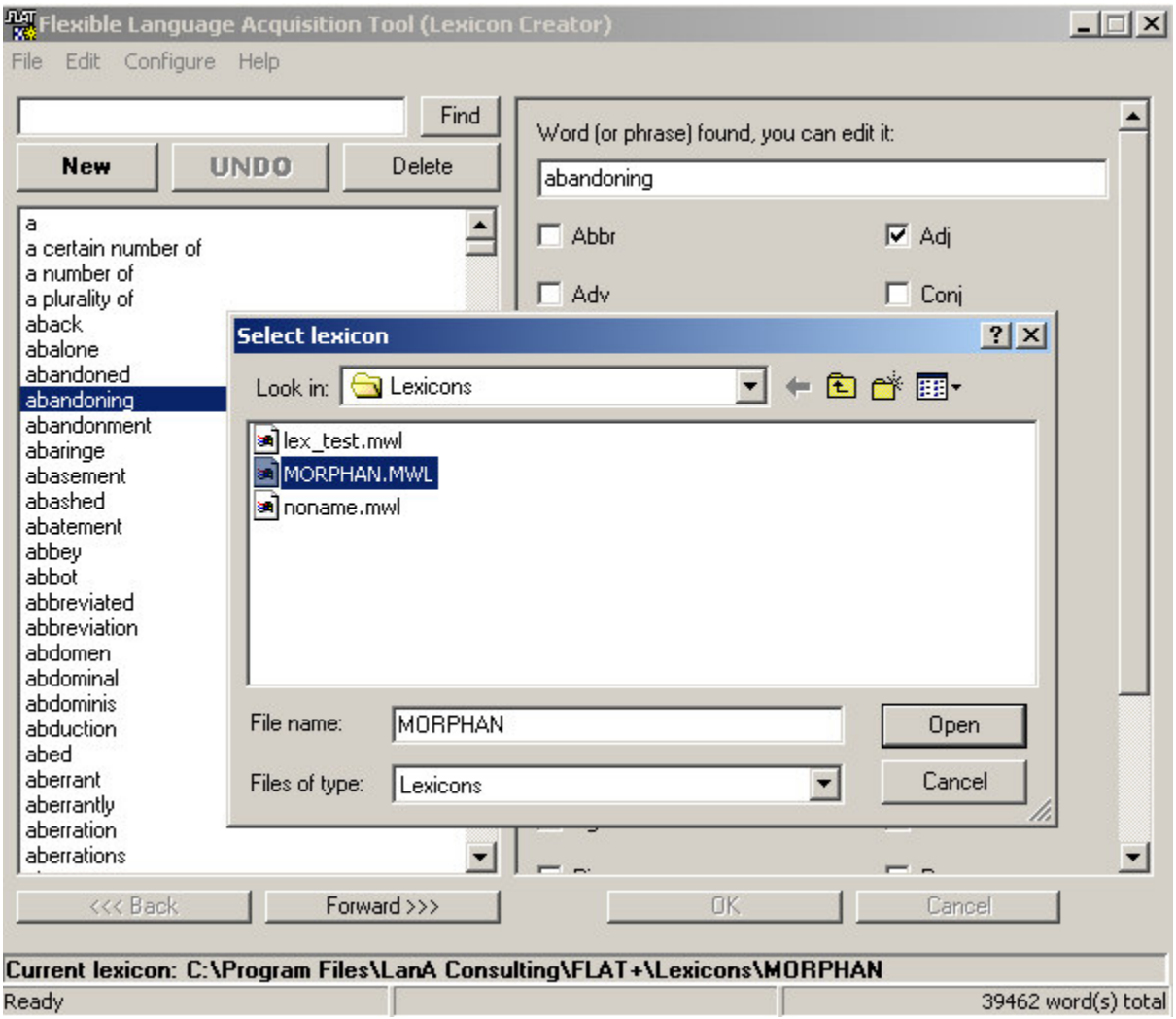

Figure 1. A screen shot of the Lexicon Creator interface with the Flat lexicon open.

How to

Start Lexicon Creator

To start the Lexicon Creator double-click on its icon.

Create a new lexicon

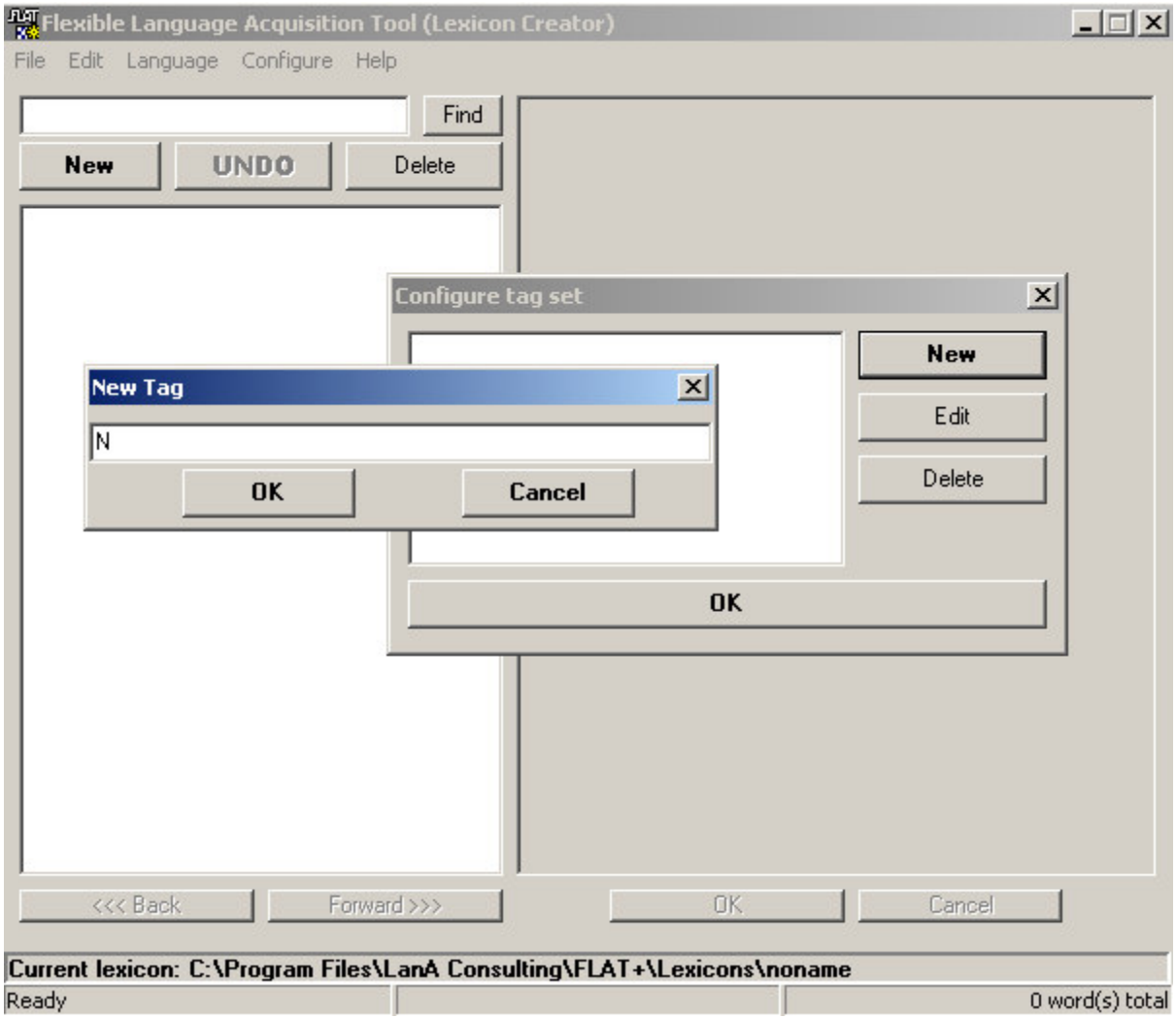

Figure 2. A screen shot of the Lexicon Creator interface at the beginning of lexicon acquisition session. The new tag "N" is being added to the lexicon tag set (currently empty)

To create a new lexicon

1. Select "New" from the "File" menu, you will get an "empty" screen of the interface.
2. **Create a tag set**. For correct program **performance it is highly recommended to have at least one tag created before you input a first lexeme in your lexicon**. Tags will always appear in Latin alphabet.
    a. Click "Configure" in the top menu, the "Configure tag set" dialogue box will appear.
    b. Click the button "New" in this box, the "New Tag" box will appear (see Figure 2).

c. Type a new tag in the active text area and click "OK" to close this window. Note that though created it will not appear in the interface until at least one lexeme is input. Proceed to create other tags or start inputting lexemes (see the section "Add new lexical items").
d. Click "OK" to close the "Configure tag set" dialogue box when finished with tags.

3. Create lexical items (see the section "Add new lexical items")

You can reuse the set of tags of any lexicon you have created before the current one.

1. Open any lexicon created earlier using the "Open" selection in the "File" menu. A password box will appear.
2. Fill in the password and click "OK". In case you have no password for the lexicon you want to open leave the type in area empty and click "OK".
    a. If you type in a wrong password a new "empty lexicon" will open.
3. Select "New" from the "File" menu, you will get a set-up interface screen with a message asking you whether you want to keep the tags from the old open lexicon.
4. Click "Yes" to get a set-up interface screen with the old set of tags but empty word list.
5. Click "No" to get an "empty" set-up interface screen.

Protect your lexicon

You might want to have your lexicon inaccessible for other users, for example, when integrating it into a different application. You can control the access to your lexicon with a password. By default any lexicon is considered to have an empty (no) password.

To create a password

1. Select "Change password" in the "Edit" menu, a dialogue box will appear
2. Leave the "Old password" text area empty and type in your password in the "New password" and "Confirm new password" text areas.

You can change your password calling "Change password" as many times as you want.

Update an existing lexicon

With "Lexicon Creator" you can easily update your lexicon by adding/editing words and tags or/and "retagging" the lexemes. Read the sections below to learn how to do this. Any variant of your lexicon thus updated can be saved and re-opened for further work.

Add new lexical items

When adding lexemes Lexicon Creator will take care of not duplicating identical items (the same lexemes with same tag set). If you add the same lexeme with different tag sets, which might happen, for example, when uploading lists of lexemes from different files, every lexeme will appear in the lexicon list just once associated with a tag set which is a unification of tags for the lexeme in the input lists.

There are several ways to **add new items to the lexicon.**

1. **Manual input.**

    a. Click the button "New" above the word list in the right pane of the interface. You will get a pop-up window with a type in text area.
    b. Type in your new lexeme (up to 4 words)
    c. Click "OK". The new lexeme will be added to the lexicon list.
        i. If you add a multiword lexeme its first word will appear as a separate item in the lexicon. Do not forget to tag it.
    d. Assign tags to this lexeme (see the section "Assign/Edit word coding for lexemes in a lexicon").
    e. Click OK in bottom of the right pane (under the tag list).

2. **Import lists of lexemes from external text file.**

    a. Create lists of lexemes in text files.
        i. **Important**. Every line of the imported text file should contain one word only positioned at the beginning of this line. You can easily do this with any external sorting program.
        ii. **Hint.** You may put different types of lexemes, for example, parts of speech, into different files to tag lists imported from the files in one take.
    b. Select "Import…" from the "File" menu. You will get the "Import text file" pop-up window.
    c. Type in the path to your file or click the "**…**" button next to the type in area to get a regular browsing window.
    d. Follow the dialogue to get the path to the file to import in the type in text area.
    e. Assign tags to all of the lexemes from the file or leave the tag boxes unchecked if the lexemes from this file require different coding. You can assign tags to them later.
    f. Click the button "Import" at the bottom of the pop-up window.

3. **Paste words recognized as unknown from the FLAT Tagger**

    a. Tag a text with FLAT Tagger (see the section "Look-up Tagger").
    b. Select the words you want to paste to your lexicon in the "Unknown words" pop-up window in the tagger interface.
    c. Right-click on any spot in the "Unknown words" window. You will get a pop-up menu of possible actions.
    d. Select "Copy selected" to paste all selected words to the Lexicon Creator or
    e. Select "Copy all" to paste all of the words without selecting any of the words.
    f. Return to Lexicon Creator.
    g. Select "Paste from tagger" in the "Edit" menu. You will get the "Paste words from tagger" pop-up window. The words to be pasted will be shown in the "Words from the buffer" window. The set of tags of your lexicon will be shown below.
    h. Check the tag boxes in case you want to assign checked tags to all the words from the buffer or leave the boxes unchecked to code the words later.
    i. Click the button "Paste" at the bottom of the pop-up window.

4. **Paste lexemes from a user lexicon**

A user lexicon can be created in your own application and paste into a FLAT lexicon (see the section "Integrating FLAT in your own application"). Use the selection "User lexicon" in the "Edit" menu.

Edit lexical items

To edit a lexical item (to correct the spelling or reassign tags)

1. Highlight a lexical item in the list of lexemes in the left pane of the Lexicon Creator interface. The lexeme will appear in the active text area at the top of the right pane of the interface.
2. Edit the word in the text area.
3. Click the "OK" button at the bottom of the right pane.
4. Recheck tag boxes to recode the lexeme
5. Click the "OK" button at the bottom of the right pane.

Remove lexical items

To remove lexical items from your lexicon

1. Select the lexemes to be removed in the list in the left pane
2. Click the "Delete" button.
    i. In case a word coincides with the first word of a phrase, which is in your lexicon you will not be allowed to delete it, you should delete the phrase first.

Add, edit or remove code tag sets

Selecting "Configure" in the main menu gives access to the "Configure tag set" pop-up window through which one can delete, edit or add new code tags to your current lexicon.

To add a new code tag

1. Click the button "New" in the "Configure tag set" pop-up window. You will get the "New tag" box (see Figure 2).
2. Type a new tag in the active text area and click "OK" to close this window. Note that though created it will not appear in the interface until at least one lexeme is input onto your lexicon.
3. Proceed to create other tags.
4. Click "OK" to close the "Configure tag set" dialogue box when finished with tags.

To edit a code tag

1. Select a tag to edit in the "Configure tag set" pop-up window.
2. Click the button "Edit" in this window. You will get the "Edit tag" box with the tag in question displayed in the active text area.
3. Edit the tag in the active text area and click "OK" to close this window.

4. Proceed to edit other tags.
5. Select a tag to edit in the "Configure tag set" pop-up window.

To delete a code tag

1. Select a tag to delete in the "Configure tag set" pop-up window.
2. Click the "Delete" button in this window.
3. Follow the dialogue.
4. Proceed to delete other tags.
5. Select a tag to edit in the "Configure tag set" pop-up window.

Assign/Edit word coding of the lexemes in the lexicon list

You can assign/edit tag coding to a single lexical item or to a group of lexemes that are either in a lexicon list or are imported from a text file or the tagger.

To assign/edit a tag of **a single lexeme in the lexicon list**

1. Highlight the lexeme.
2. Check/Uncheck tag boxes in the right pane of the interface
3. Click the "OK" button.

To assign/edit tag of **a group lexemes in the lexicon list**

1. Highlight the lexemes.
2. Right-click on any spot in the left pane to get a pop-up menu.
3. Select "Set tags for selected" in the pop-up menu. You will get the "Set or clear tags for selected words" pop-up window with the selected words displayed at the top. Note that when dealing with a group of words you cannot clear and assign tags to them in one take.
4. To clear the tags from the words coding
    i. Check the "to clear" radio button to delete or to assign tags from the coding of the group of words in question
    ii. Check tag boxes next to the tags you want to delete in the pop-up window
    iii. Click the "OK" button in the pop-up window
5. To assign new tags to the words coding
    i. Check the "to set" radio button to delete or to assign tags from the coding of the group of words in question
    ii. Check tag boxes in the pop-up window
    iii. Click the "OK" button in the pop-up window

Assign lexeme coding when importing them form a file or the tagger

When importing lexemes from a text file or pasting them from the tagger you can either assign the same tag coding to all the lexemes shown at the top of the corresponding pop-up window (see the

section “Add new lexical items”) or put them in the lexicon uncoded and then proceed as described in the section above.

To assign tag coding to all the imported/paste lexemes

1. Check tag boxes in the pop-up window
2. Click the “OK” button

Look through the word list

To look through the list of lexemes

1. Use the scroll bar (you can only scroll 10,000 lexical items)
2. Use the “Back” or “Forward” buttons that make it possible to scroll every next/previous 10,000 lexical items in turn.

Search/Filter lexical items

Depending upon the selection in the “Edit” menu the user can get either a full word list of the lexicon or sub lists of words sorted by their suffixes, prefixes, or tags. It is also possible to get a list of untagged words and tag them using the Lexicon Creator interface.

To search a single lexical item

1. Type a lexeme in the text area at the top of the left pane of the interface
2. Click the button “Find”.
    i. In case the lexeme is in the lexicon it will appear in the active window at the top of the right pane of the interface. You can further edit it.
    ii. In case the lexeme is not in the lexicon a lexeme that follows it in the alphabet list will appear

To search any lexical item by prefixes, suffixes or by code tags

1. Select “Filter” from the “Edit” menu. The “Select filter” pop-up window will appear.
2. Check a radio button corresponding to your parameter of search.
3. In case you have checked the “prefix” or “suffix” radio button type in a corresponding string of characters in the active text area at the top of the window. Click the button “Filter” to get a list of lexemes meeting your search parameters in the left pane of the interface.
4. If you checked the “tag(s)” radio button check tag boxe(s) corresponding to your search parameters. Click the button “Filter” to get a list of lexemes meeting your search parameters in the left pane of the interface.

To search new (uncoded) words select “New words” from the “Edit” menu.

To restore the list of all lexemes in the lexicon select “Show all words” from the “Edit” menu.

Tagger Look-up

Tagger Look-up assigns the lexemes all code tags from a particular lexicon (created with FLAT Lexicon Creator). The tagger is pipelined to FLAT Lexicon Creator, so that you can, for example, tag the same text based on different lexicons, thus defining how large, "deep" or "shallow" a lexicon should be for your application.

Any changes you might make in the lexicon are immediately traced in the tagger thus allowing for operative testing of lexicon coverage. The tagger reports immediately whether the text is covered by a lexicon listing unknown words (if any) in a pop-up window. The text to tag can be either typed in the active text area of the tagger control interface or downloaded from a text file. Both the input text and results of any tagging session can be traced in the interface and saved. The names of the tagged files will automatically be marked with the suffix "Tag". This makes it very easy to compare different traces based on different lexeme/tag sets.

One of the essential features of the tagger is that it can directly be used for lexical acquisition from relevant corpora due to its functionality to import lexemes recognized as unknown to a FLAT lexicon. (see the section "Add new lexical items"). The coverage of the lexicon is thus updated incrementally.

Interface

The interface screen contains the main top menu consisting of "File", "Configure" and "Help" selections, and a control screen divided into two windows (see Figure 3).

The upper window will show a text to be tagged. The lower window of the screen will show the traces of look-up tagging. At the set up stage both windows are empty. The upper window is active. The user can either download a text from an external file or type it directly into the upper interactive window. Inscription above both windows remind you what file you are working with. Assigning a new name to the file using "Save as" from the "File" menu will change it in the interface.

How to

Start the tagger

To start the tagger double-click on the tagger icon.

Configure the tagger for a tagging session

Important. Before starting a tagging session you should **configure the tagger** to a particular FLAT lexicon.

To configure the tagger to a particular FLAT lexicon.

1. Select "Tagging…" in the "Configure" menu. You will get the "Tagging configuration" pop-up window.'
2. Click the "Browse" button to browse for a lexicon you want to work with to get its name in the "Lexicon" text area.
3. Enter the password (see section "Protect your lexicon")
4. Click the "OK" Button

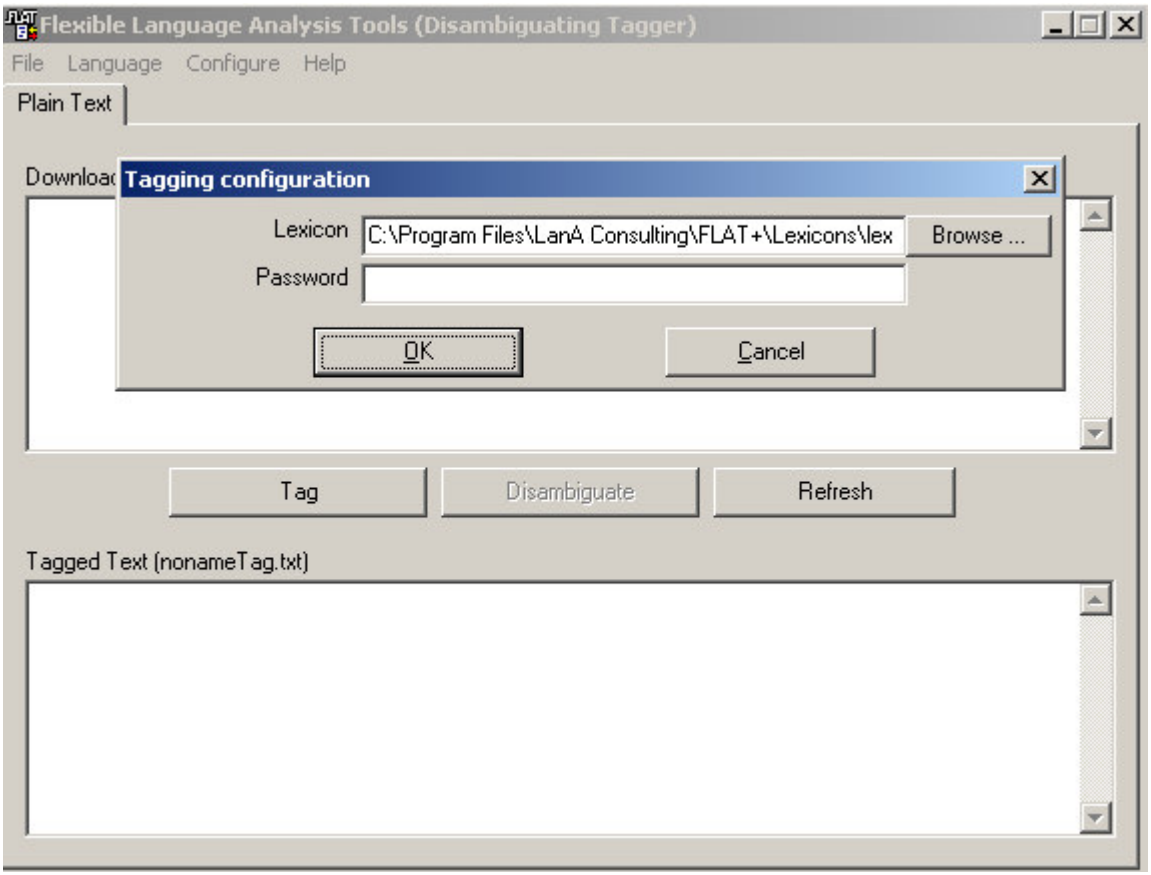

Figure 3. A screenshot of the tagger interface, which is being configured to a FLAT text lexicon

The tagger is set to work. The last configuration (the name of the lexicon and password) you worked with will be remembered by the tagger so that next time you start the tagger it is considered as a default configuration, thus making it unnecessary to configure the tagger every time.

Download, tag and save

You can either type a text to be tagged directly into the upper interactive window or download a text from an external file. Note that you can only download a plain text. The text cannot exceed 16 Kb. Symbols **{, }, ~** are not allowed in the input text.

1. Select "Download" in the "File" menu.
2. Follow the usual dialogue until you see your text in the upper window.
3. Assign a name to the file by selecting "Save as" from the "File" menu. The name of the file will appear in the interface.
4. To tag a text click the button "Tag" after the text appears in the upper window.
5. To save traces of tagging select "Save tagged" or "Save tagged as" from the "File" menu. The names of the tagged files will automatically be marked with the suffix "Tag".

Improve the coverage

One of the essential features of the FLAT tagger is that it can directly be used for lexical acquisition from relevant corpora due to its functionalities to show, save and/or import unknown words to FLAT Lexicon Creator (see Figure 4). The coverage of the lexicon is thus updated incrementally.

If the text you tagged is not covered by your lexicon unknown words will appear in the "Unknown words" pop-up window (see Figure 4).

You can

- Close this window by clicking on the "OK" button.
- Save all the words from this window in a file to further work with them.

    1. Right-click on any spot in the "Unknown words" window You will get a pop-up menu of possible actions.
    2. Select "Save all words as" to save them in a text file that you can import to a FLAT Lexicon Creator lexicon later.

- Paste the new words to a FLAT lexicon

    1. Select the words you want to paste to your lexicon in the "Unknown words" pop-up window in the tagger interface.
    2. Right-click on any spot in the "Unknown words" window You will get a pop-up menu of possible actions.
    3. Select "Copy selected" to paste all selected words to the Lexicon Creator or
    4. Select "Copy all" to paste all of the words to the Lexicon Creator without selecting any of the words.
    5. Open (or return to) Lexicon Creator.
    6. Select "Paste from tagger" in the "Edit" menu. You will get the "Paste words from tagger" pop-up window. The words to be pasted will be shown in the "Words from the buffer" window. The set of tags of your lexicon will be shown below.
    7. Check the tag boxes in case you want to assign checked tags to all the words from the buffer or leave the boxes unchecked to code the words later.
    8. Click the button "Paste" at the bottom of the pop-up window.

You can always re-open the "Unknown words" window in the tagger by selecting "Show unknown words" in the "Configure" menu.

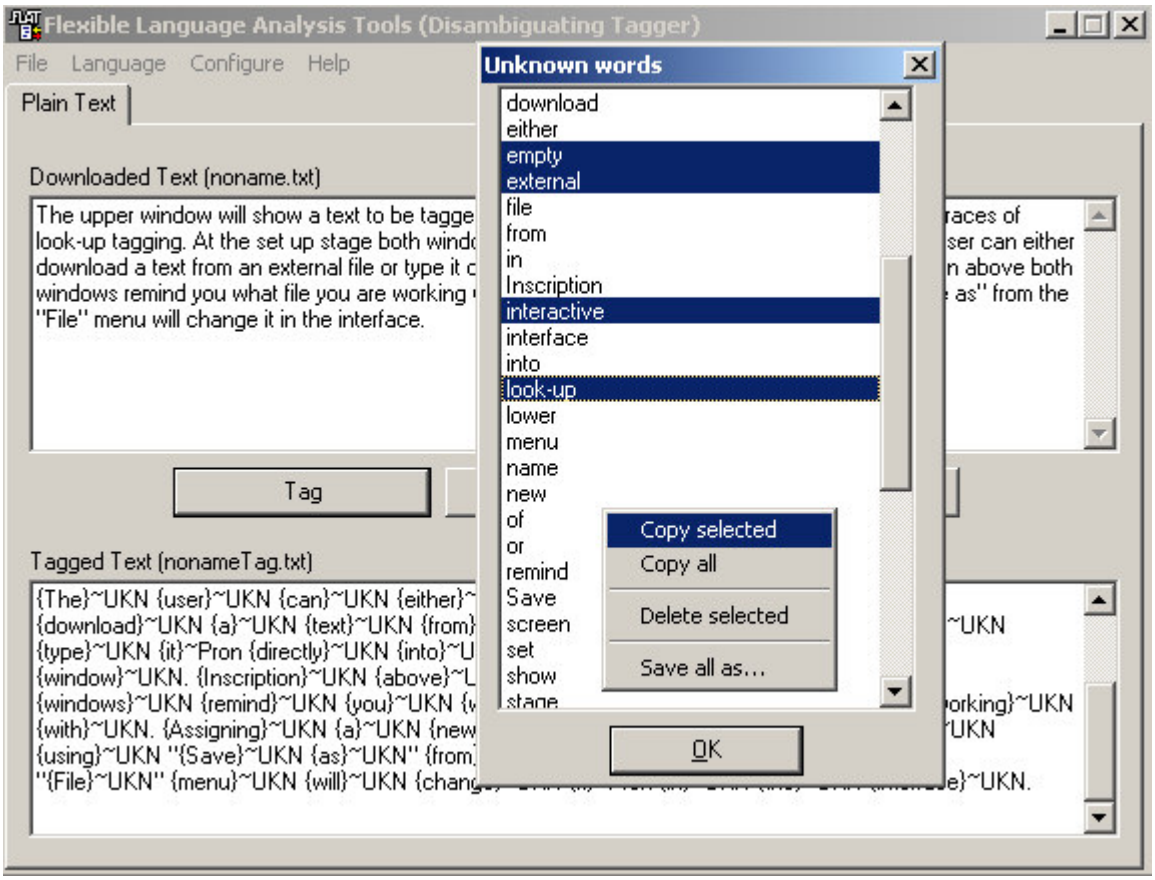

Figure 4. Import of words from the tagger

Tag Disambiguation

FLAT Tagger includes a rule-based tag disambiguator that disambiguates multiple tags. For tag disambiguation you will need the buttons “Disambiguate” and “Refresh” and one more selection in the “Configure” menu, - “Tag disambiguation”, which opens an interactive interpreter for writing or updating tag disambiguation rules. Immediately after saving new rules an updated trace can be displayed in the interface.

The rules are linked to a particular FLAT lexicon. A set of disambiguation rules tuned to a patent domain are included in FLAT together with a domain lexicon (See section FLAT lexicon).

FLAT is provided with a disambiguation rule interpreter so that you can create and test different sets of disambiguation rules based on the same or different FLAT lexicons/tags. In spite of formalism simplicity the rules have quite a rich and flexible inventory of “right hand side” conditions that can provide for rather fine (vs. coarse) disambiguation. For example, one can check a context within a five-string window with the tag in question in the middle. The context could be checked either in terms of tags and/or word strings. It is also possible to check whether a context tag/word belongs to a certain list, etc. The disambiguation rules can be both “reductionistic” and “substitution” ones. Any changes you might make in the rule set are immediately traced in the tagger interface thus allowing for operative testing of rule coverage capacity.

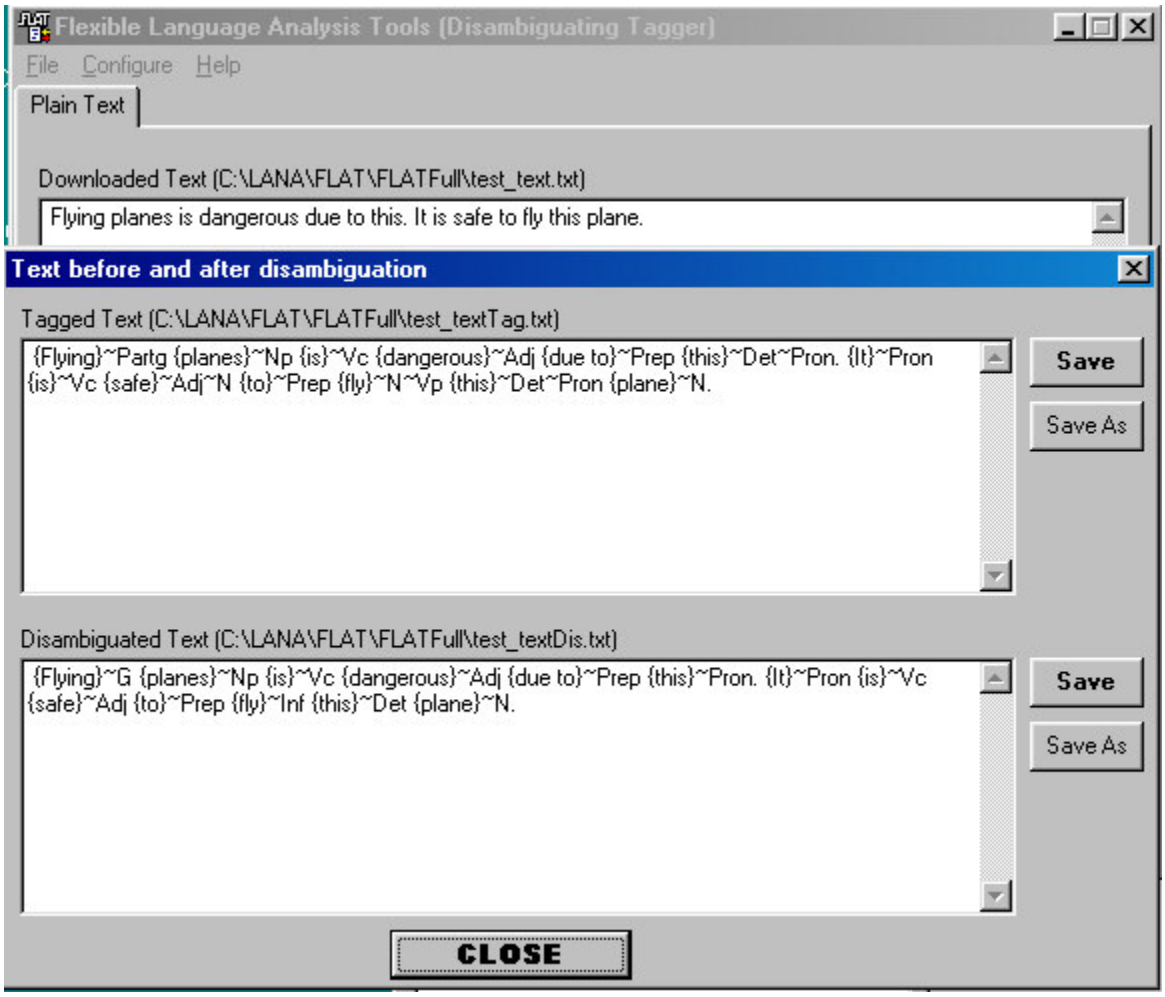

Figure 5. A screen shot of the Tagger interface with the pop-up disambiguation window. It shows the results of disambiguation of a test text based on test lexicon (see the section "Disambiguation rule formalism")

How to work with FLAT to Disambiguate Tags

1. Create a lexicon with FLAT Lexicon Creator.
2. Open the tagger by double-clicking on its icon.
3. Configure the tagger to a lexicon for which you will write the disambiguation rules
4. Write disambiguation rules (read the section "Write disambiguation rules")
5. Tag the text, (update your lexicon with new words in case there are any and tag the text again),
6. Click the button "Disambiguate". You will get the pop-up window with two control screens. The upper screen shows a tagged text before disambiguation, the lower screen displays the same text with tags disambiguated according to your set of disambiguation rules. You can compare both texts to brush up your rules. The analysis traces can be saved for further use.

Interface for disambiguation rules

FLAT Tagger is provided with a special **interactive interface** for writing disambiguation rules in a simple formalism, which will be described further. Open this interface by selecting "Tag disambiguation" in the "Configure" menu of the tagger interface.

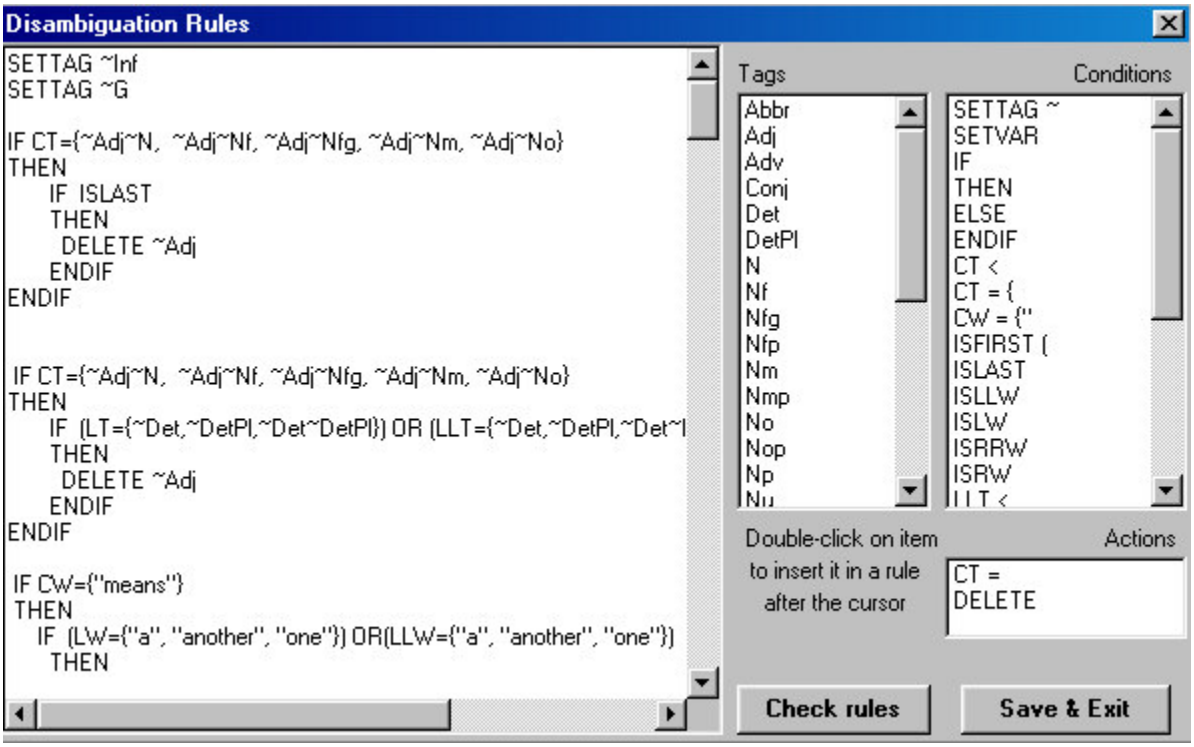

Figure 6. A screen shot of the interactive interface for writing disambiguation rules filled with FLAT rules (see the section "FLAT knowledge" for the meaning of supertags)

The right pane of the interface is a type-in area in which you can write the rules. To make rule writing less tedious and time consuming the right pane of the interpreter contains two clickable menus. The first menu lists all the tags from the FLAT lexicon you configured your tagger to; the second menu contains expressions used in rules. After you wrote a rule click the "Check it" button. This button triggers a rule check and in case of a mistake displays an error description message. After closing the message box on the "OK" button click you will find the cursor right in the place where a correction should be made. Immediately after saving new rules an updated trace can be displayed in the interface.

Disambiguation rule formalism

Disambiguation rules are always based on a certain FLAT lexicon. In spite of the IF-THEN-ELSE-ENDIF formalism simplicity the rules have quite a rich and flexible inventory of "right hand side" conditions that provide for rather fine (vs. coarse) disambiguation. For example, one can check a context within a five-string window with the tag in question in the middle. The context could be checked either in terms of tags and/or word strings. It is also possible to check whether a context tag/word belongs to a

certain list, etc. The disambiguation rules can be both "reductionistic" and "substitution" ones. Rules are **case sensitive**.

Disambiguation formalism includes a ***declaration*** (optional) part and ***rules*** (obligatory)**.** We will first describe the structure of disambiguation rules and then the declaration part of the formalism.

Rule structure

The formats of disambiguation rules allowed by the interpreter are listed below (see also Figure 6).

1. The upper level rule has always the IF-THEN-ENDIF structure as follows.

   Rule format 1:
   IF *condition*
   THEN
   Action1
   Action2
   ENDIF

2. The structure IF-THEN-ENDIF can be embedded in the upper (and next) level structure (as many times as you need):

   Rule format 2:
   IF *condition1*
   THEN
   IF condition2
   THEN
   Action1
   ENDIF
   ENDIF

   You can also embed the structure IF-THEN-ELSE-ENDIF in the rules in format1 and format2 and get a rule in format3.

   Rule format 3:

   IF *condition1*
   THEN
   IF condition2
   THEN
   Action1
   ELSE
   Action2
   ENDIF
   ENDIF

It is possible to have any number of embedded structures in one rule, for example, like the following:

Rule format 4:

IF condition1
THEN
    IF condition2
    THEN
        Action list or IF-THEN-ELSE-ENDIF structure

    ELSE
        Action list or IF-THEN-ELSE-ENDIF structure

    ENDIF
ENDIF

Attention: There **cannot be ELSE in the upper level of the rule**, that is there cannot be a rule like this:

***IF** *condition*
THEN
Action list or IF-THEN-ELSE-ENDIF structure
ELSE
Action list or IF-THEN-ELSE-ENDIF structure
ENDIF

Below all actions and conditions are described

Conditions

Conditions can be **simple** or **complex.**

Complex conditions

Complex conditions are Boolean expressions combining other conditions as follows (mind the brackets):

(*condition1*)**AND**(*condition2*)
(*condition1*)**OR**(*condition2*)
NOT(*condition*)

Using Boolean operators you can write conditions of any complexity (**mind the brackets**), for example:

((*condition1***)AND(***condition2***))OR(NOT(***condition3***))**

Any rule can contain any number of imbedded IF-THEN-ELSE structures and Boolean expressions combining several rules into one. Thus though formally there can be up to 50 separate rules, in practice you can cover as much disambiguating procedures as you can think of within the suggested formalism.

The order of the rules is relevant. The program linked to the rule interpreter performs a disambiguation action after the context meets the rule condition and moves to process next tag.

Simple conditions

There are several groups of simple conditions.

Tag context conditions help tag disambiguation depending upon the tags context. The tag context is a five-tag window with the tag in question in the middle.

TAG ::
- CT – a current tag
- LT – the tag left of the current tag
- LLT – the tag two steps left of the current one
- RT – the tag right of the current tag
- RRT – the tag two steps right of the current one

TAG *values* can be *single* (~N, ~V) or *multiple* (~N~V, ~Adj~V, ~Adj~V~N). The order of single tags in a multiple tag is not relevant. You can write the tag context condition as follows.

TAG = {*value1, value2, …, valueN}*, for example,

CT = {~N}
LLT = {~N, ~N~V}
RRT = {~Adj~N~V, ~Adj~N…}
RT = {~V}
LT = {~Adj, ~N}

There can be up to 15 values listed for one TAG in a rule.

You can also write a more flexible tag context condition in the form

TAG < {*single_value*}, which means that TAG can have any set of multiple values, that necessarily include said single_*value*, for example,

CT < ~N

means that a current tag can have any multiple value including ~N (e.g., ~N~V, ~Adj~N or ~Adj~N~V), but CT cannot have a *single* value like just ~N , or any multiple value not including ~N (e.g. ~V, ~N or ~Adj~V)

Word context conditions disambiguate tags depending upon the word context. The word context covers 5 words with the word having a current tag to disambiguate in the middle.

WORD:

CW – a current word that has a current tag
LW – the word left of the current word
LLW – the word two steps left of the current word
RW – the word right of the current word
RRW– the word two steps right of the current word

You can write the tag context condition as follows

WORD = {"*word1*" , "*word2*" , …}, for example,

CW = {"comprises"}
LW = {"measuring"," checking", "improving" }

There can be up to 15 words listed like this.

You can write the same conditions in a more compact form:

WORD = *Variable_name,*

where *Variable_name* is, for example, a name of a list of words, which you declare in the declaration part of the formal description of disambiguation rules (see how to write the declaration part further in the text). You will thus avoid listing the same sets of words in rules, for example, a condition can be written like

CW = ListA

Other conditions are as follows:

ISLAST – means that a current word is positioned before a *period, comma, colon or semicolon.*

Conditions listed below may be specially useful when analyzing phrases, i.e., when you know the phrase boundaries

ISLW– means that there is a word left to the current one in a text (the current word is not the first one in the text)
ISRW- means that there is a word right to the current one in a text (the current word is not the last one in the text)
ISLLW means that there are two words left to the current one in a text (the current word is not the first one in the text)
ISRRW means that there are two words right to the current one in a text (the current word is not the last one in the text)

Actions

There are only two actions in tag disambiguation rules:

CT = ~tag_any-value
DELETE ~tag_single-value,

for example,

CT = ~N
CT = ~N~V
DELETE ~N
DELETE ~Adj

The first action is used in "substitution" rules. It will substitute the tag of a current word (or phrase) with a tag stated in a rule.
The second action is used in "reductionistic" rules. This action will try to **delete** a single value tag from the tag of a current word. This action will not be executed in two cases.

- If there is no stated single tag assigned to the current word this action will do nothing.
- If the current word has only one single-value tag, no deletion will take place to prevent the occurrence of untagged words.

The rules can be further augmented by using the **declaration part** of the rule formalism.
You can declare new tags and lists of words.

By **declaring tags** you can enrich your tag set at the disambiguation step without changing your lexicon. For example, you can detect gerunds, if your lexicon only codes such words as *measuring* as -ing forms of the verbs. Tags are declared one by one as follows (mind the "~" sign).

SETTAG ~G
SETTAG ~Inf

You can declare as many tags as you want. Maximum length of a tag is 15 symbols (in case of multiple tags). Tags thus declared will be valid in rules, but will not be added to the lexicon.

You can also **declare lists of words** as follows.

SETVAR Var_name = { "*word1*", "*word2*", … }, for example,

SETVAR preps = {"for", "by", "in order to"}
SETVAR nums = {"one", "two", "three", "four", "five", "six", "seven"}
SETVAR oneword = {"one"}

Var_name – is the name of the word list. It must only consist of letters; **digits in Var_name are not allowed** in the current formalism. There can be up to 20 words in one list and up to 20 Var_names declared. You can further use these Var_names as values of WORDs (see the section "Examples of rules).

Examples of disambiguation rules

In this section we give several examples of disambiguation rules. The rules are purely illustrative and only cover the following text:

Flying planes is dangerous due to this. It is safe to fly this plane.

Configure the tagger to the test lexicon lex_test. Download the text_test file to the tagger. Tag the text to get

*{Flying}~**Partg** {planes}~Np {is}~Vc {dangerous}~Adj {due to}~Prep {this}~**Det~Pron.***
*{It}~Pron {is}~Vc {safe}~**Adj~N** {to}~Prep {fly**}~N~Vp** {this}~**Det~Pron** {plane}~N.*

Disambiguate and compare the text before and after disambiguation:

*{Flying}~**G** {planes}~Np {is}~Vc {dangerous}~Adj {due to}~Prep {this}~**Pron**. {It}~Pron {is}~Vc {safe**}~Adj** {to}~Prep {fly}~**Inf** {this**}~Det** {plane}~N.*

Open the rule interpreter and see what rules were used for disambiguation. Pay attention to the format. At the top of the interactive interface pane declared are two new tags (G and Inf) and one list (SETVAR list). We hope these examples will help you to write your own "real world" rules.

FLAT Patent domain knowledge

Lexicon and tag disambiguation rules

The FLAT lexicon included in this product has been semi-automatically acquired from a 5 million-word on-line corpus of complete patent disclosures. It is a list of supertagged lexemes. In our model a supertag codes morphological information (such as POS and inflection type) and semantic information,- an ontological concept, defining a word membership in a certain semantic class (such as object, process, substance, etc.). For example, the feature structure of a noun supertag is as follows:

Tag [POS
[Noun
[Object [plural, singular]
Process [-ing, other [plural, singular]]
Substance [plural, singular]
Other [plural, singular]]]]]

The "depth" of supertags is specific for every part of speech and codes only the amount of knowledge that is believed to be sufficient for patent texts analysis procedure. We do not assign equally "deep" supertags for every word in this lexicon. For example, supertags for verbs include such morphological features as verb POS, and verb forms (-ing form, -ed form, irregular form, finite form). For finite forms we further code the number (plural or singular).

The following notations are used in the FLAT patent domain lexicon and tagger:

Abbr-abbreviation; Adj – adjective; Adv – adverb; Conj – conjunction; Det - determiner, singular; DetPl – determiner plural; N-noun, singular, object; Nf – noun, singular, action, does not end in -ing; Nfp –noun, plural, action, does not end in –ing, Nfg - noun, singular, action, ends in - ing; Nm - noun, singular, substance; Nmp - noun, plural, substance; No - noun, singular, other; No - noun, plural, other; Num –numeral;
P – verb, finite, singular, present; Pd – verb form, ends in –ed; Pg – verb form, ends in –ing; Pi – Verb form, irregular; Pis – verb form "is"; Pare – verb form "are"; Pp – verb form, finite, plural, present; Prep – preposition; PronPs – possessive pronoun; Qu –quantifier; Wh – Wh-word.

Tag disambiguation rules are based on these supertags. You can see them in the interpreter interface.

Integration of FLAT into a different application

With FLAT Control**,** based on Active X technology, you can create your own Windows-applications that integrate all FLAT features.

The FLAT Control main features are:

- Tagging of a given text based on a lexicon created in FLAT Lexicon Creator
- Tagging and disambiguating of a given text based on a lexicon and rules, created with FLAT Lexicon Creator and Tagger
- Spell checking
- Dynamic input of new lexical items to a FLAT lexicon from a user (your) lexicon of a different application

FLAT Control is compatible with all development environments that comply with Active X technology (can use Active X controls).

FLAT Control is an invisible Active X Control, based on the MFC library. It has several methods:

- AddWord
- GetDict
- GetTagList
- GetUntagged
- SetDict
- SpellCheck
- Start
- TagAndDismbiguate
- TagIt
- CheckDisRules (*reserved for internal use in FLAT components*)
- GetLastDisError (*reserved for internal use in FLAT components*)
- GetLastDisErrorPosition (*reserved for internal use in FLAT components*)
- LoadDisRules (*reserved for internal use in FLAT components*)
- SaveDisRules (*reserved for internal use in FLAT components*)

FLAT Control is delivered with two sample projects (for Visual Basic 6.0 and Borland C++ Builder 4.0) for you to learn how to use FLAT Control. However, before you can use them FLAT must be installed.

The sample project for Visual Basic 6.0 is located in the directory Samples/VB on installation CD.
The sample project for Borland C++ Builder 4.0 is located in the directory Samples/BCB.

NB: In order to use FLAT Control in Borland C++ Builder, you must install the package provided for FLAT Control. The package is located in the directory Samples/BCB/Package.

To create an installation package for your applications that uses FLAT Control, you must include the files *MorphAnControl.ocx* and *DLLMorphAnNew2.dll* in the installation package from System32 or System sub-folder in the Windows folder.

All methods not reserved for internal use are described below.

AddWord

Return type:
Nothing

Parameters:

Name	Type	Description
Word	String	Word (or phrase) to add into a user lexicon
Tag	String	One valid tag (that already exists in your lexicon) without ~ as the first character

Description:
This method is used to dynamically add words or phrases to the user lexicon of your own application. A new word (or phrase) will be recognized and tagged without restarting the **FLAT** system.

Restrictions:
Phrases should not consist of more than four words. As every new word/phrase is transferred from a particular text where it has just one meaning it will be but input into the **FLAT** lexicon with the tag corresponding to that meaning (chosen by the user. See section Look-up tagging). You can always add other tags to this lexical item by pasting them to the Lexical Creator using the functionalities of the Lexical Creator (see section “Add tags”).

GetDict

Return type:
String

Parameters:
No

Description:
Returns a current lexicon path and filename.

Restrictions:
None

GetTagList

Return type:
String

Parameters:
None

Description:
Returns all tags in a current lexicon as a string separated by a period (.), for example, "Adj.N.P.Prep."

Restrictions:
None

GetUntagged

Return type:
String

Parameters:
None

Description:
Returns a list of unknown words (words that were not found in a current lexicon and tagged as "~UNK") as a string separated by comma (,), for example. *"device, word, developed"*.

Restrictions:
None

SetDict

Return type:
Short integer (1 or 0)

Parameters:

Name	Type	Description
Filename	String	Full filename (path and filename) of a lexicon
Password	String	Valid password for this lexicon

Description:
This method is used to set a current lexicon. You must provide a full filename and valid password to use a lexicon. In case of success (filename and password are OK) the system will return 1, otherwise 0. You should check the return value before using other methods. **Using other methods after returning 0 can lead to unpredictable results.**

Restrictions:
None

SpellCheck

Return type:
Short integer (1 or 0)

Parameters:

Name	Type	Description
Word	String	Word to check whether it can be found in a lexicon (you can use this method for your own spell checking)

Description:
Tries to match a text word against a current lexicon. In case of success, it will return 1, otherwise 0.

Restrictions:
Only single words can be matched.

Start

Return type:
Nothing

Parameters:
No

Description:
This method is used to open a current lexicon, initialize it and prepare it for a working session. You **must** call this method once after each successful **SetDict** method. After the **Start** you can call other methods anytime. It is not necessary to call the **Start** method before the TagIt or **TagAndDisambiguate** methods. **Use of other methods (excluding GetDict and SetDict) before the Start method can lead to unpredictable results**.

Restrictions:
None

TagIt

Return type: String

Parameters:

Name	Type	Description
Text	*String*	Lexicon look-up

Description:
This method is used match text words against a current lexicon. For every matched lexical item it assigns all tags found in the item entry. No disambiguation is done at this stage of analysis. You can develop your own tag disambiguator or use one of our FLAT.

Restrictions:
The text cannot exceed **16 Kb**. Symbols **{, }, ~** are not allowed in the input text. After this method has been used you can call **GetUntagged** to get a list of unknown words.

TagAndDisambiguate

Return type: String

Parameters:

Name	Type	Description
Text	*String*	Text to tag and disambiguate

Description:
This method is used to tag a text based on current lexicon tags and disambiguation rules. It returns a tagged text with completely or partially disambiguated tags (depending upon the disambiguation rules).

Restrictions:
The text cannot exceed 16 Kb. Symbols **{, }, ~** are not allowed in the input text. After this method has been used you can call the **GetUntagged** method to get a list of unknown words.

FLAT in a classroom

Extreme user-friendliness of the tool interfaces makes the software rather suitable for the classroom when teaching NLP. It allows the students, computational linguists to concentrate on the linguistic issues of developing NLP applications (e.g. machine translation) without augmenting the problems with programming issues. Actually, the description of tools in the previous sections gives some hints about for what and how to use FLAT for the teaching of NLP. First of all an instructor might use the tools to familiarize students with the problems of NLP software, linguistic error analysis, specificity of the sublanguage approach to NLP, etc. Another way is to use FLAT is to participate in building an NLP system. For example, based on the tagging lexicon acquisition tool and interactive rule interpreter, exercises can be developed, using a specially designed test suit, to investigate the problem of coverage and knowledge necessary for disambiguation. A student can experiment with changing (inventing) tags to see whether a "deeper" description of lexical units gives better resolution. Interpreter can also be used to teach students to write formal (programmable) language descriptions, etc.