The PLUG Word Aligner - User's Guide

Jörg Tiedemann
Department of Linguistics
Uppsala University
joerg@stp.ling.uu.se

September 15, 2000

Contents

1 NAME

PLUG Word Aligner — User's-Guide

2 What is PWA?

The PLUG Word Aligner, PWA, is a collection of tools for finding word correspondences in a bilingual parallel text, a bitext. If the parallel text is a translation bitext, i.e. a source text with a translation into another langugage, word correspondences, typically, represent translation equivalents. The basic building blocks of PWA are modules for knowledge-lite word alignment. Knowledge-lite is to be understood as opposed to knowledge-intense; a minimum of linguistic knowledge is engaged making the tools fairly language independent. The word alignment modules may be combined into different systems and configurations and adapted to different text types and language pairs. The input to the system is sentence aligned bitext and the output is a list of word occurrence correspondences, link instances, and a lexicon of word type correspondences, link types. Words may consist of one or more text segments.

PWA incorporates modules for two word alignment systems, Linköping Word Aligner, LWA, and Uppsala Word Aligner, UWA. Both systems were developed within the PLUG project at the Department of Computer and Information Science at Linköping University and the Department of Linguistics at Uppsala University. Both systems are integrated in a modular corpus toolbox, which, in addition, contains tools for the generation of monolingual collocations, and for the evaluation of alignment results, the PLUG Scorer — PLS.

The PWA modules are integrated in the Uplug system which is the corpus toolbox for processing textual data. The system includes three main components:

```
* UplugIO - a transparent I/O interface
* UplugSystem - system for combining re-usable modules
* UplugGUI - a graphical user interface
```

PWA was compiled as one of the main deliverables of the PLUG project. It is provided free of charge for research and teaching purposes to academic institutions. It is implemented in Perl and Perl/Tk and compiled into binary format using the Perl2Exe compiler by DynamicState. PWA is made available for Microsoft Windows, Linux, and Sun Solaris according to the license agreement that can be found on the PLUG home page at http://stp.ling.uu.se/~corpora/plug/pwa/.

PLUG refers to the project "Parallel Corpora in Linköping, Uppsala and Göteborg". It is a co-operative project aimed at the development, evaluation and application of computer programs for alignment and data generation from parallel corpora with Swedish as either the source or the target language. Applications include machine translation, computer-aided translation, translation databases, multi-lingual web dictionaries and translator's training. The participating departments were Computer and Information Science, Linköping university, Linguistics, Uppsala university, and, during the initial phase of the project, Swedish language, Göteborg university.

The project ran for 24 months starting in March 1998. The project was part of the Swedish Language Technology Programme funded by the Swedish Council for Research in the Humanities and Social Sciences, HSFR, and the

Swedish National Board for Industrial and Technical Development, NUTEK. For further information about the project, see http://stp.ling.uu.se/~corpora/plug/.

3 How to install PWA

3.1 How to install PWA on Windows

PWA is distributed as self-extracting setup program. Save the file 'setup.exe' on your system and start it (e.g. by double clicking in the file manager etc.). Follow the instructions during the setup program. The installation routine offers default configurations that might be sufficient for most systems.

All necessary files will be installed on your system and 'PWA' will be added to the 'Program' sub-menu in the start-up menu. Here, you can find shortcuts for starting PWA and for removing it from your system ('uninstall').

3.2 How to install PWA on Linux

PWA is distributed as compressed tar-archive. Use 'gunzip' for decompressing and 'tar -xf' for creating files and directory structures (?? represents the version of your PWA-copy):

```
gunzip pwa??.tar.gz
tar -xf pwa??.tar
```

The commands above will create a sub-directory 'pwa' in the current directory. In this sub-directory, all PWA-related files, except temporary files, will be stored. Temporary files will be stored in '/tmp/' by default. You have to change the default-settings for temporary files in './ini/UplStrm.ini' in your PWA home directory in order to change the location of temporary files:

```
...
temporary files
    PLUG XML = '/temporary/files/'
    general = '/temporary/files/'
...
```

4 How to start PWA

4.1 How to start PWA on Windows

Start PWA by double clicking on the 'pwa' shortcut. PWA has been added to the 'Program' sub-menu in the start-up list as well. You can also use this shortcut for starting the system. Alternatively, you can always start PWA by double-clicking on the 'pwa.exe' file in the PWA directory from Explorer (File manager).

Note!!

PWA needs some time to start the interface on your desktop. The implementation of the system requires a command-line window to pop up for a very short time. This window will disappear directly again. Don't panic! The PWA interface will appear after the command-line window was destroyed.

4.2 How to start PWA on Linux

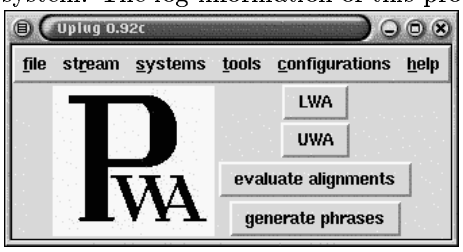
Start PWA with the 'pwa' program in your PWA home directory. The system expects to find necessary files relative to the directory 'pwa/' in your home-directory. Use the '−h' parameter for setting the PWA-directory if your PWA-installation is installed at a different location:

```
pwa -h /my/pwa/directory
```

The 'pwa'-program starts the graphical user interface (GUI) with its main window. The PWA main window will be displayed when starting PWA for the first time. The interface consists of the Uplug 'menu-bar' at the top of the window and the Uplug 'system-frame' below. It is possible to start PWA sub-systems without starting the graphical user interface. Specify the name of the sub-system (such as LWA) as additional parameter when calling the 'pwa'-program:

```
pwa -h /my/pwa/directory sub-system-name
```

PWA will use the current configurations for executing the specified sub-system. The log-information of this process will be printed to STDOUT.



5 The menus and commands

The menu-bar includes six pull-down menusto be found in the menu bar: 'file', 'stream', 'systems', 'tools', 'configurations' and 'help':

5.1 The 'file' menu

- The 'view' command can be used to open text files and edit them.
- The 'quit' command can be used to exit the program.

5.2 The 'stream' menu

The Uplug system provides a transparent I/O interface to sequential data collections (streams).

- Use 'show stream data' to open data streams, to read the first 100 records and to display them.
- Use 'open stream' to open data streams without reading.
- Use 'add stream specification'in order to define a new stream specification (look at the section for advanced users).

5.3 The 'systems' menu

- The 'open system' command contains a sub-menu that can be used to open the system-frame for one of the included PWA sub-systems.
- The 'show/hide input/output stream' flags specify if all input/output streams shall be displayed in the current system frame or not. Look at the sections on 'How to look at input/output streams' .
- The 'close system' command closes the current system frame and returns to the PWA main window.
- See the sections for advanced users for information on how to 'create systems' and how to 'add modules'.

5.4 The 'tools' menu

The tools menu contains user defined tools that have been added to the GUI. Tools can be added with the 'add tool' command in the 'configurations' menu. The 'tools' menu contains 5 UWA-specific tools by default:

- edit target phrases
- edit UWA:target.ini
- edit UWA:target.mor
- edit source phrases
- edit UWA:source.ini
- edit UWA:source.mor
- edit UWA:basic dictionary

Check the section about UWA for more information.

5.5 The 'configurations' menu

- See the sections for advanced users for information on how to 'add tools' to the 'tool' menu and to the text editor window.

5.6 The 'help' menu ...

... will hopefully include some help soon. Right now, only the version number and the Uplug-logo will be displayed by clicking on the 'about Uplug' command.

6 The interface

The system frame in the PWA main window contains four buttons for choosing one of the included PWA sub-systems. These are **LWA**, **UWA**, **Evaluate alignments** and **Generate phrases**. Click on corresponding buttons for opening the system frame of the desired sub-system.

6.1 How to start sub-systems

Each sub-system frame contains a frame with its module sequence and a command-button frame at the bottom. The simplest way to start the current system is to press the 'start system' button and the system will start the process. On Unix systems, a separate Xterm window will be opened in order to run the system in the background. On Windows, the Uplug interface will be blocked until the process in finished. It is possible to run parts of Uplug sub-systems. Press one of the 'generate' buttons in order to run the system from the initial module up to the module that corresponds to the chosen 'generate' button.

6.2 How to look at input/output data (streams)

Input and output data collections (streams) are listed in the right-most columns for each of the modules in the sub-system frame. As the default, only the output data collections are shown. You can use the commands in the 'system' menu for hiding/showing data collections. Data collections can be opened by simply clicking on the corresponding name of the collection. The system displays data in separate windows in form of tables. The data collection window contains a command button frame at the bottom. The 'search' button can be used for searching all entries that match the search pattern in the row of input fields above the command buttons. The 'read' function is used to read the first 100 data entries from the collection. The 'count' function counts the number of data records in the collection and displays this value on top of the window. The 'save as' button is used to save the data in another collection (look at section 'How to specify a data stream' for information on how to specify such collections). The 'close' function closes the data collection and destroys the window. Additionally there will be a 'next' button if the collection contains more than 100 entries. Use this button for reading the next 100 data records. There is a row of input

fields below the data table. Here one can specify a pattern (in form of regular expressions) for each of the columns in the data collection. The current pattern will be searched by pressing 'search' and the result will be displayed in a separate data window. Use always the 'clear' button for removing the previous pattern!!! Even empty strings will be considered to be search patterns!!

6.3 How to change settings

Settings are stored in special configuration files for each of the modules in the system. Settings can be modified by using the module parameter dialog which can be opened by clicking on the 'settings' button of the specific module. The dialog window contains three pull-down menus:

```
* input (streams)
* output (streams)
* parameter
```

Select the parameter you want to set from the corresponding list. Now, corresponding attributes will be displayed in the dialog window. Input and output parameter correspond to stream specification forms. Other parameter can be set by selecting check boxes, modifying scale values, or changing input entries depending on the configuration file. Additionally, there might be button that display 'show array' in the parameter frame. In this case, the attribute values are complex structure that can be modified by clicking on these buttons. A separate dialog window will be displayed for modifying thess attributes. Press always 'cancel' for discarding your changes and 'ok' for storing your modifications in the configuration file. Note: All changes will be stored in configuration files that will be used by processes that run the current sub-system. Changes that have been made may influence processes that run at the same time in the background! Note: Input and output attributes will be set automatically to the same values as in earlier modules of the module sequence if they have been specified with the same parameter name! This feature was added for supporting iterations.

6.4 How to switch to sub-systems

A module in Uplug systems might be a sub-systems itself. One can switch to the configuration frame of the sub-system by pressing the 'settings' button of the corresponding module. Use the 'close' button to return to the parent frame.

6.5 How to skip modules

Modules can be skipped by selecting the 'skip' check-box for the corresponding module. Note that this might influence the process such that some necessary input data is not available for the preceding module.

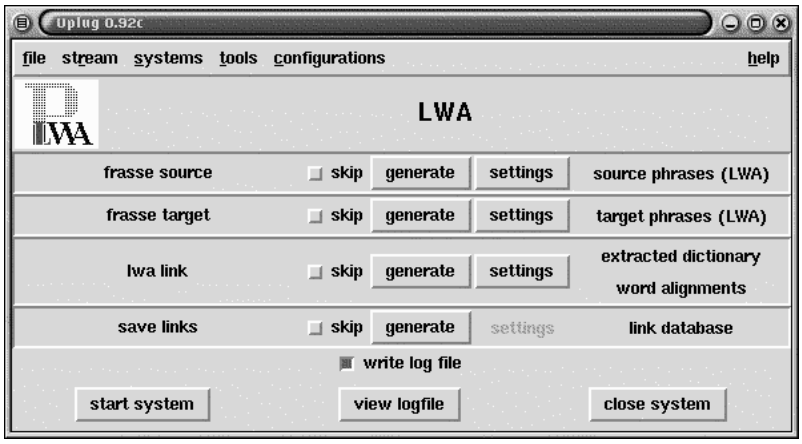
6.6 How to view log-file information

Uplug systems produce some output with information about the ongoing process. This information will be stored in the systems log-file if the check-box 'write logfile' is selected. The log-file can be inspected by pressing the 'view log file' button.

7 LWA

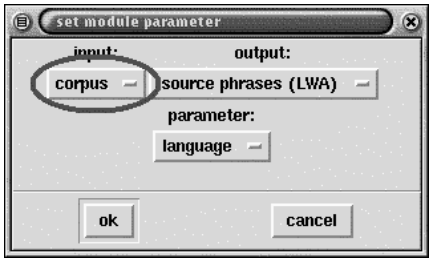
7.1 How to run LWA

- Select the 'LWA' system from the 'open system' menu or press the 'LWA' button in the PWA main window. The LWA frame will then be displayed.
- If you want to run the system with all sub-processes included, just click on 'start system'. PWA under Windows will be blocked until the process is finished. Unix systems will open a new window in order to run the process in the background. If you want to omit any of the steps in the process, tick the appropriate 'skip' checkbox.
- If you want to run only a sub-process of the LWA module, press one of the 'generate' buttons in order to run the system from the initial module up to the module that corresponds to the chosen 'generate' button.

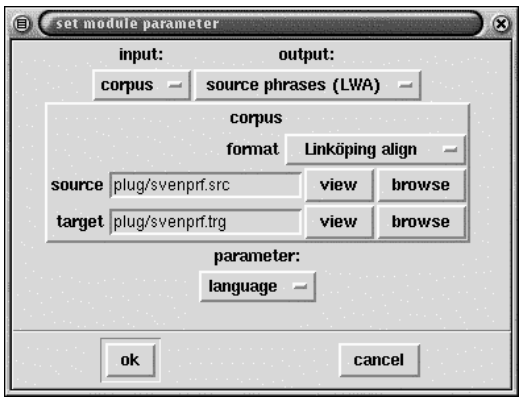


7.2 How to select a different input text

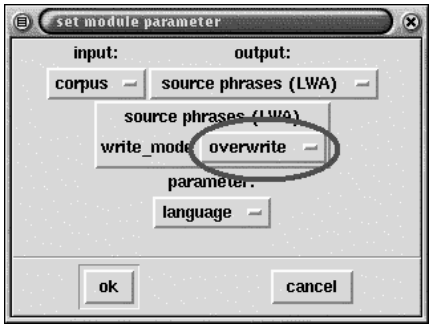
1. Press 'settings' for the 'frasse source' module.
2. Chose the input attribute 'corpus' (select it again even if displayed already).



3. Select the data format of your input text (currently 'Linköping align' and 'plug XML' are available) and set the required attributes for your text. Data in 'plug XML' format will be taken from the file that has been specified with the 'file' attribute; data in 'Linköping align' format will be taken from the source language file and the target language file that have been specified with the 'source' and 'target' attributes.

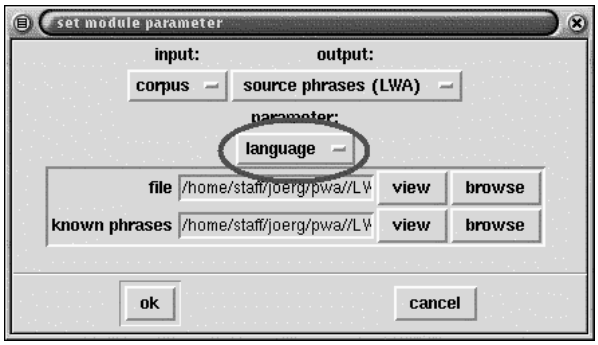


4. Now, choose the output stream 'source phrase (LWA)' and change the 'write_mode' to 'overwrite' (otherwise the old phrase file will be re-used if it exists).



5. Adjust the language settings for the source phrase generation. Select the 'language' parameter and choose corresponding files for the source language of your corpus. Language files are available for Swedish (swe.lang, general_swe_phrases.txt), English (eng.lang, general_eng_phrases.txt), and

French (fre.lang, general_fre_phrases.txt) in the sub-directory 'LWA/LanguageData/' in your PWA home directory.



6. Do the same modifications for the 'frasse target' module corresponding to the target language in your corpus.

   Note: you don't have to change the 'corpus' stream specifications again because the system will use the settings from the previous module (frasse source) anyway!!

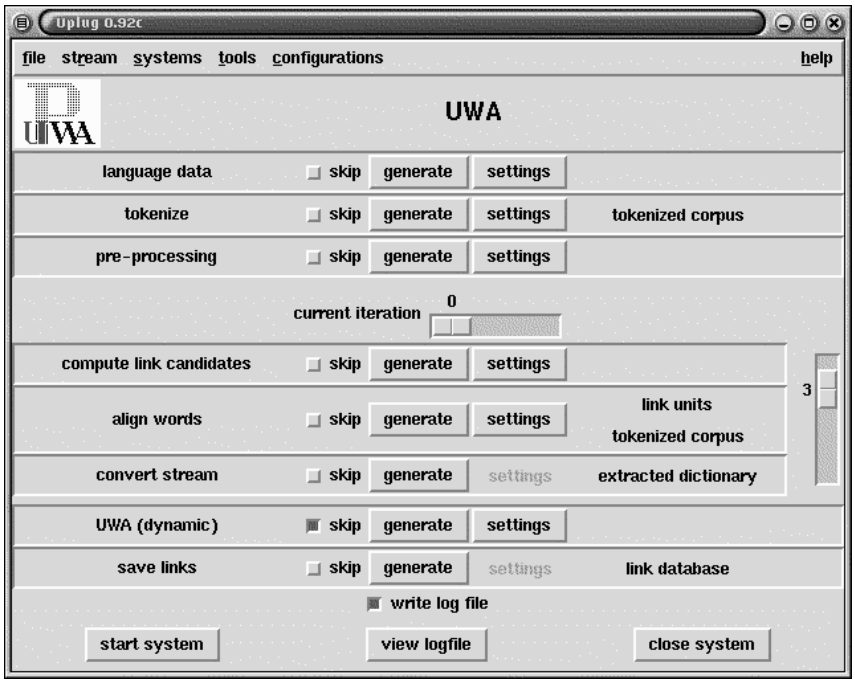
   Now, you have to set the source and target language in the 'lwa link' module:

7. Select 'settings' for this module and change the 'language' parameter for the 'source' and the 'target' parameter (currently 'Swedish', 'English', and 'French' are available).

8 UWA

8.1 How to run UWA

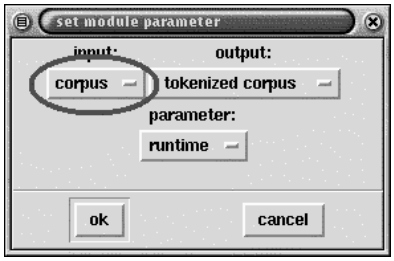
- Select the 'UWA' system from the 'open system' menu or press the 'UWA' button in the PWA main window.
- If you want to run the system using all sub-processes, just click on 'start system'. PWA under Windows will be blocked until the process is finished. Unix systems will open a new window in order to run the process in the background. If you want to omit any of the steps in the process, you just tick the appropriate 'skip' checkbox.
- If you want run only a part of the UWA module, press one of the 'generate' buttons in order to run the system from the initial module up to the module that corresponds to the chosen 'generate' button.



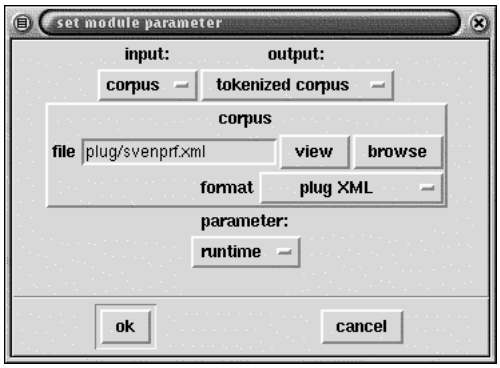
8.2 How to select a different input text

1. Press 'settings' for the 'tokenize' module.

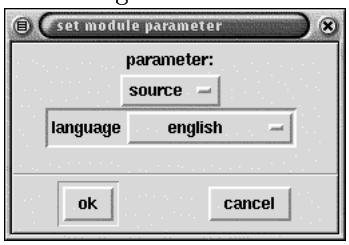
2. Choose the input attribute 'corpus' (select it again even if displayed already).



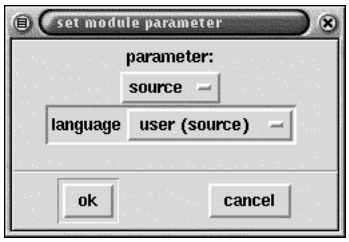
3. Select the data format of your input text (currently 'plug XML', 'Linköping align' and 'Uppsala align' are available) and set the required attributes for your text. Data in 'plug XML' and 'Uppsala align' format will be taken from the file that has been specified with the 'file' attribute; data in 'Linköping align' format will be taken from the source language file and the target language file that have been specified with the 'source' and 'target' attributes.



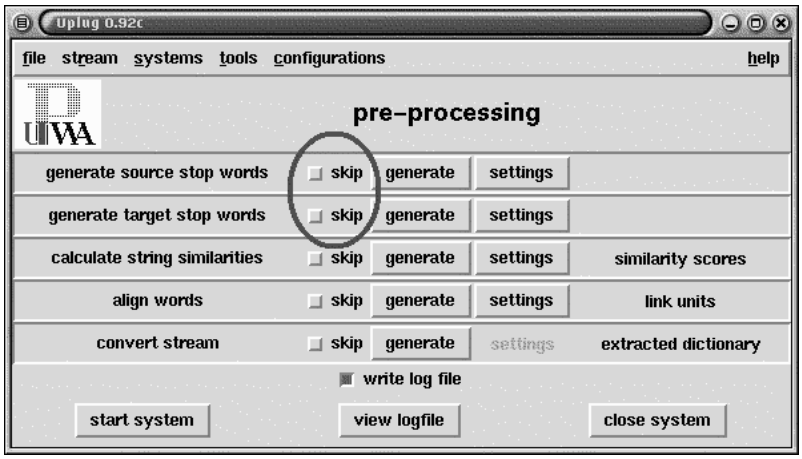
4. Choose 'settings' for the 'language data' module. Set the 'language' parameter corresponding to the source and the target language of your corpus using the 'source' and the 'target' category.



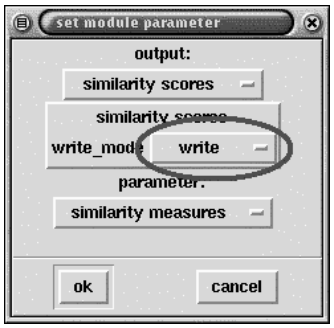
5. Select 'user (target/source)' if the language is not included in the list. In such cases, the language specific information will be taken from the 'source.ini' and the 'target.ini' file. You can edit them by selecting 'edit UWA:source.ini' and 'edit UWA:target.ini' from the 'tools' menu.



6. If you want the system to select a list of stop words by inspecting the current corpus you can change the settings for the 'pre-processing' module: Simply click on 'settings' for the pre-processing module and remove the skip-marker for the first two modules ('generate source/target stop words').



7. You should also change the settings for calculating string similarity scores in the pre-processing module. Click on 'settings' for the module 'calculate string similarities', select 'similarity scores from the output stream list, and change the 'write_mode' to 'overwrite'. This will force the system to compute a new list of candidates instead of re-using older ones.



8. Go back to the UWA system frame ('close' the pre-processing sub-system) and start the system by pressing 'start system'.

9 Evaluate alignments with PLS

The PLUG Link Scorer can be used to evaluate word alignments by comparing them to manually created gold standards. Such standards have to be stored in PWA gold format (see the section 'What is the PWA gold format?' below).

9.1 How to run PLS

PWA comes with a sample gold standard for the included Swedish/English text.

1. Choose 'settings' for the 'evaluate word alignments' module. Select 'word alignments' from the list of 'input' data. Set the 'stream name' parameter

for word alignments to one of the possible alternatives: **link database** — all links in the database; **word alignments (UWA)** — word alignments from the recent UWA process; **word alignments (LWA)** — word alignments from the recent LWA process. Confirm the new settings by clicking the 'ok'-buton.

2. Start the evaluation by clicking on the 'start system' button within the sub-system frame.

3. Open the log-file after the evaluation process terminated by clicking on 'view logfile'. The log-file represents the evaluation protocol including the results of each individual comparison, a summary of the evaluation, and calculations of different evaluation scores (precision, recall, F-value). Look at the small sample below:

```
================================================================
correct:  svenprf110   full    full     step 1
incorrect:svenprf185   skall (skall)    be (will)       step 1
missed:   svenprf220   dagar   days
partial:  svenprf178   kommunernas      local(the local authorities,1(1)/3)
...
================================================================
number/step              all   1   5   7
number gold     :        100
number returned :         49  41   1  10
================================================================
number correct  :         24  21   0   3
number partial  :         18  15   1   5
number incorrect:          7   5   0   2
================================================================
precision (ARCADE):                    44.000%
precision (PLUG):                      68.627%
precision (PWA):                       70.859%
================================================================
recall (ARCADE):                       33.503%
recall (PLUG):                         50.000%
recall (PWA):                          34.835%
================================================================
F-measure (ARCADE):                    38.041%
F-measure (PLUG):                      57.851%
F-measure (PWA):                       46.708%
================================================================
```

9.2 How to select a different Gold Standard

Gold standards have to be specified using the 'PWA gold' format. Choose a new gold standard as follows:

1. Choose 'settings' for the 'evaluate word alignments' module.
2. Select 'facit' from the list of 'input' data and set the 'stream name' parameter for 'facit' to 'new'.
3. Select 'PWA gold' as 'format' parameter.
4. Specify the file with the gold standard in the input field or choose the file by browsing the file system ('browse').
5. Confirm your changes by pressing the 'ok' button

10 Generate phrases

The generate phrases module includes LWA and UWA modules for the automatic generation of word collocations.

10.1 How to run the phrase generator

Simply press the 'start system' button in order to run the phrase generator. Results can be inspected by clicking on the corresponding output data collections after the process was terminated.

10.2 How to select a different input text

The phrase generator is adjusted to compile word collocations for Swedish/English bitext as the default. Phrase generation for texts in other languages requires language specific files that have to be included in the generator. These modifications and extensions are not topic of this manual.

1. Select 'settings' from the 'frasse source' module.
2. Select 'corpus' from the list of input data (do it even if corpus is the selected item of the list already).
3. Specify the new text by selecting corresponding files. **Linköping align** format expects two plain text files for sentence aligned parallel text; **plug XML** format requires bitext in XML format according to the specifications from the plugXML.dtd.
4. Confirm your changes by pressing the 'ok' button.

11 LWA for advanced users

11.1 How to set additional LWA parameters

Each module contains a set of parameters that can be modified with the 'settings' dialog.

```
-----------------------------------------------------
frasse source
       language
               file                    <filename>
               known phrases           <filename>
       token
               skip separators         yes|no
frasse target
       language
               file                    <filename>
               known phrases           <filename>
       token
               skip separators         yes|no
lwa link
       alignment
               link window             yes|no
       source
               language                SWEDISH|ENGLISH
                                              |FRENCH
       target
               language                SWEDISH|ENGLISH
                                              |FRENCH
       token
               minimal frequency       2-50
-----------------------------------------------------
```

12 UWA for advanced users

12.1 How to add phrases

User defined phrases can be added for both, the source and the target language. Use the command 'edit source/target phrases' to open a text editor with the corresponding phrase file. Just put one phrase on each row in order to specify additional phrases. Now, you have to change the settings for the 'text segmentation' module in the sub-system 'identify phrases' (select the sub-system 'identify phrases' from the 'open system' menu or go via UWA->compute link candidates->identify phrases). Open 'settings' for the 'text segmentation' module. Choose 'source/target phrase' from the input stream list. The following streams are available:

```
source/target phrase (UWA)         - phrases generated by UWA only
source/target phrase (LWA)         - phrases generated by LWA only
source/target phrase               - user defined phrases only
source/target phrase collection    - all of the above
```

Select the stream of your choice.

12.2 How to add basic translations

UWA allows you to add basic translations to a bilingual lexicon that will be used in the alignment process. Use the command 'edit UWA:basic dictionary' from the tools menu in order to specify such dictionary entries. Put each pair on one row and separate source and target language token by exactly one TAB-character.

12.3 How to set additional UWA parameter

Each module contains a set of parameters that can be modified with the 'settings' dialog. UWA includes three sub-systems, 'pre-processing', 'compute link candidates', and 'UWA (dynamic)'. Furthermore, the pre-processing sub-system includes two further sub-systems for generating stop word lists and the sub-system for computing link candidates includes a sub-system for identifying phrases.

```
* UWA
-------------------------------------------------------
language data
       source
              language                swedish|english
                                      |german|user (source)
       target
              language                swedish|english
                                      |german|user (target)
tokenize
       runtime
              print progress          yes|no
       source
              digit                   <set of digits>
              semi word delimiter     <set of character>
              word delimiter          <set of character>
       target
              digit                   <set of digits>
              semi word delimiter     <set of character>
              word delimiter          <set of character>
align words
       align sequence
              step                    <array of align steps>
       co-occurrence statistics 1
              minimal value           <threshold>
       co-occurrence statistics 2
              minimal value           <threshold>
       similarity scores 1
              minimal value           <threshold>
       similarity scores 2
```

```
              minimal value           <threshold>

* pre-processing
-----------------------------------------------------
calculate string similarities
       similarity measures
              minimal score           <threshold>
       source
              maximal phrase length   1-10
              minimal token length    1-10
       target
              maximal phrase length   1-10
              minimal token length    1-10
align words
       align sequence
              step                    <array of align steps>

* compute link candidates
-----------------------------------------------------
count co-occurrence freq
       token pair
              maximal distance        1-20
              minimal frequency       1-10
              minimal length difference 0-1
calculate association scores
       statistics
              method                  Dice|MutualInformation
                                          |t-score
              minimal score           <threshold>
align low frequency tokens
       frequencies
              frequency threshold     1-20
              maximal frequency       1-100

* identify phrases
-----------------------------------------------------
generate phrases
       phrase
              maximal phrase length   1-10
              minimal frequency       1-100
       statistics
              method                  Dice|MutualInformation
              minimal score           <threshold>
text segmentation
       source
              maximal phrase length   1-10
       target
```

```
               maximal phrase length   1-10

* get source stop words
-------------------------------------------------------
tokenize
       runtime
               print progress          yes|no
       source
               digit                   <set of digits>
               semi word delimiter     <set of character>
               word delimiter          <set of character>
       target
               digit                   <set of digits>
               semi word delimiter     <set of character>
               word delimiter          <set of character>
get N best
       data
               IC lower than           <threshold>
               larger than             <frequency>
               top n                   <rank>

* get target stop words
-------------------------------------------------------
tokenize
       runtime
               print progress          yes|no
       source
               digit                   <set of digits>
               semi word delimiter     <set of character>
               word delimiter          <set of character>
       target
               digit                   <set of digits>
               semi word delimiter     <set of character>
               word delimiter          <set of character>
get N best
       data
               IC lower than           <threshold>
               larger than             <frequency>
               top n                   <rank>
-------------------------------------------------------
```

UWA (dynamic) is another variation of the Uppsala Word Aligner. Here, the system does not rely on previously compiled phrases but tries to find the best alignment even for phrases iteratively. It applies mainly the same modules with similar parameter settings except two additional modules, 'create inverted file' and 'co-occurrence statistics (idx)'. The following parameter can be set for the latter:

```
-----------------------------------------------------
co-occurrence statistics (idx)
       source
               maximal phrase length   0-10
               minimal frequency       1-100
       statistics
               method                  Dice
               minimal score           <threshold>
       target
               maximal phrase length   0-10
               minimal frequency       1-100
       token pair
               maximal distance        0-60
               minimal frequency       1-100
               minimal length difference 0-1
-----------------------------------------------------
```

13 The Uplug interface for advanced users

13.1 How to specify a stream

Streams are specified by setting appropriate attributes in the stream specification dialog. This dialog pops up e.g when using the 'show stream data' and 'open stream' commands from the 'stream' menu. Known streams can be selected by using the corresponding 'stream name' from the pull-down menu in the stream specification dialog. Unknown streams can be specified as 'new' streams. Select always 'new' from the pull-down menu for stream names (even if it is displayed already) in order to specify an unknown stream. Now, one has to select the stream data 'format'. The dialog window will show additional attributes according to the format you have chosen. Look at the description of stream formats in the 'WHAT-IS?' guide for additional information.

13.2 How to add stream specifications

- Use 'add stream specification'from the 'stream' menu in order to define a new stream specification that can be accessed by stream name. See further the section on 'how to specify a stream' above.

13.3 How to create new sub-systems

New sub-systems can be added to the Uplug interface. Use the commands from the 'systems' menu as follows:

- The 'create system' command can be used to create a new Uplug subsystem. Sub-systems have to have a unique identifier in form of a system name. Select the name of the system and press 'ok' for creating a new

system with the specified name. The following dialog window can be used to specify the sub-system. The basic attributes ('configdir', 'logfile', 'logfiledir', 'write to log file') are set by default. Please, check if these specifications are valid. Uplug systems are defined by sequences of modules. Press the 'show array' button in order to open the module sequence. The initial list is empty. Press 'add' in order to add a new module to the end of the current sequence, 'shift' to remove the first module in the sequence, and 'pop' to remove the last module in the sequence. Each module in the list can be selected from pull-down menus that include all available modules. Press 'ok' to finish your specifications or 'cancel' to discard your changes. The create-system dialog includes also an 'add' button for additional system attributes (the attribute name has to be unique!) and a 'delete' button for removing a specific attribute. New systems won't be displayed in the PWA main window in your current session. However, you can reach the new system by selecting it from the 'open system' menu.

- The 'add module' command can be used to include additional modules in the Uplug system. Such modules can be included in Uplug sub-systems later on. The procedure of adding modules is quite similar to creating systems. First, specify a unique name for the new module. Secondly, the following dialog can be used to specify the necessary attributes of this module. The definition of supported modules is very open. There is mainly two different kinds of modules: Perl libraries ('perl lib') and executable files ('bin' and 'perl script'). Choose the corresponding 'type' of the module you want to add. Perl libraries have to include the 'filename' of the library, which will be included with the '#require' command, and the 'command' attribute that specifies the sub-function in this library that will be called. Executable files require the 'command' attribute only. The file/tool will be searched in the specified 'directory' only. The 'configuration' attribute specifies the name of the parameter file in Uplug format for the current module. It will be searched in the configuration directory of the Uplug sub-system that includes the current module. The 'add' and 'delete' button have the same function as in 'create system'. Be aware that the Uplug system does not validate or integrate new modules such that they fit automatically to other modules. New modules do not have to use the I/O interface or the configuration mechanism of the Uplug system. Each module has to be implemented explicitely such that it can co-operate with other modules in Uplug sub-systems.

13.4 How to add user defined tools

Tools can be added to the 'tools' menu and to the in-build text editor. Use the commands from the 'configurations' menu as follows:

- The 'add tool' command is used to add a graphical tool to the 'tools' menu. A graphical tool is simply a sub-function in a common Perl/Tk-script that takes a reference to the main window as its first argument.

The sub-function has to create its own window by calling the 'Toplevel' function of Perl/Tk. The 'file' attribute specifies the name and location of the Perl script, the 'command' attribute specifies the name of the sub-function in the script, 'additional parameters' specify a list of comma-separated argument-values that have to be passed to the sub-function, and the 'label' attribute specifies the label in the 'tools' menu.

- The 'add file tool' command is used to add tools that can be called when text-files are displayed with the 'view' command from the 'file' menu. Each file tool will be accessible by additional buttons that will be displayed in the text-edit window. The 'command' attribute specifies the tool that will be called with the current file name as its first parameter. The 'parameter' attribute specifies the string of additional parameters that will be passed to the specified tool. The 'file pattern' specifies a pattern of filenames for which the tool shall be accessible. The 'label' attribute specifies the name that shall be displayed on the button for calling the tool. The system expects the result to be printed to STDOUT by the specified command. The result will be displayed within the text-edit frame if it includes less than 5 rows. Larger output will be displayed in a separate text-edit window.

14 Data formats

14.1 What is PLUG XML?

PLUG XML is an encoding scheme that was especially defined for the bilingual sentence-aligned PLUG corpus using the eXtensible Markup Language (XML). The document type definition for PLUG XML can be found in the 'lib' sub-directory of the PWA distribution. The following picture illustrates a small sample from a typical PLUG bitext:

```
-----------------------------------------------------
<?xml version="1.0"?>
<!DOCTYPE plug SYSTEM "/corpora/PLUG/plugXML.dtd">
<PLUG>
<header creator="joerg@power.ling.uu.se"
        date.created="Thu Jun 18 12:47:34 CEST 1998">
  <fileDesc name='svenprf.xml'/>
  <profileDesc type="political text"
               src.lang="sv"
               trg.lang="en"/>
</header>
<corpus>
  <subcorpus>
    <titleStmt>
      <title>Regeringsf\"oklaringen</title>
    </titleStmt>
```

```
    <publicationStmt>
      <publisher>svenska regeringen</publisher>
    </publicationStmt>
    <profileDesc src.lang='sv' trg.lang='en'/>
    <alignStmt aligned.by='Erik Tjong Kim Sang'
               aligned.at='Uppsala University'/>
    <document>
      <doc.header lang='sv' no.words='1876'>
        <fileDesc name='1988svA.tei' size='31986'/>
      </doc.header>
      <doc.header lang='en' no.words='2474'>
        <fileDesc name='1988enA.tei' size='33890'/>
      </doc.header>
      <doc.body>

<align id='svenprf3' link='1-1'>
  <seg lang='sv'>
    <s>
    Sveriges neutralitetspolitik \"ar av avg\"orande
    betydelse f\"or v{\"aa}rt lands fred och oberoende.
    </s>
  </seg>
  <seg lang='en'>
    <s>
    Sweden's policy of neutrality is of decisive
    importance for our peace and independence.
    </s>
  </seg>
</align>

</doc.body>
</document>
</subcorpus>
</corpus>
</PLUG>
-----------------------------------------------------
```

PLUG XML files contain segments from both bitext parts. Aligned sentences are stored in *align* structures as sequential *seg* structures. This makes PLUG XML easy to process for our purposes were word correspondences have to be found in sentence aligned bitexts.

14.2 What is Linköping align?

This data format is used by the LWA system. The sentence aligned bitext is stored in 2 separate files with aligned segments on corresponding lines. Each

line starts with an identifier enclosed in '#' characters. Comment lines start with '%'. Aligned sentences are stored on corresponding lines.

```
-----------------------------------------------------
% date_created: Mon May 22 22:02:34 2000
##svenprf5## There is wide popular support for this policy.
##svenprf6## It will be pursued with firmness and consistency.
-----------------------------------------------------
```

14.3 What is Uppsala align?

The Uppsala align format was used in earlier versions of the UWA alignment system. In this format bilingual sentence alignments are combined in one text file. Each alignment starts with a header followed by the actual text segments.

```
-----------------------------------------------------
 1X:8:8:8:8:
      (1) Den kommer att fullf\"oljas med kraft och konsekvens.
      (2) It will be pursued with firmness and consistency.
  1X:7:8:7:8:
      (1) Syftet \"ar att stimulera arbete och sparande.
      (2) The aim is to stimulate work and saving.
-----------------------------------------------------
```

14.4 What is UWA tab?

The 'UWA tab' format stores stream entries in tabulator-separated data fields. Column names are specified in the header line.

```
-----------------------------------------------------
# columns: (id,source,target)
# date_created: Mon May 22 22:19:48 2000
# fields: (id,source,target)
svenprf2       talman  speaker
svenprf2       ledam\"oter       members
svenprf2       sveriges        swedish
svenprf2       riksdag parliament
svenprf3       sveriges        sweden's
-----------------------------------------------------
```

14.5 What is UWA dic?

The 'UWA dic' format is used to store the extracted dictionary from UWA alignments in alphabetically sorted order. The source language item, enclosed in curly brackets, starts a new data record followed by a list of corresponding target language items which is also enclosed in curly brackets. Target language items are sorted by their frequency. Look at the small example below:

```
-----------------------------------------------------
# created_by: UWA
# date_created: Mon May 22 17:26:33 2000
{Centraleuropa}
{
    1X:Central Europe
}
{De}
{
    6X:The
    3X:the
}
-----------------------------------------------------
```

14.6 What is the DBM format?

The 'DBM' format applies database management libraries that can be tied to Perl hash structures. The GNU database manager (GDBM) will be chosen as the default. However, other database manager are supported such as SDBM and NDBM. The file format depends on the database manager that has been used and data in such files are usually stored in binary format. DBM streams may use certain fields as unique key values in the database. The system will generate numeric values if no key is selected.

14.7 What is the PWA gold format?

The 'PWA gold' format was defined for the storage of manually created reference data (gold standards). Currently, UplugIO supports only read access for PWA gold streams. The PWA gold format corresponds to the output format of the PLUG Link Annotator (PLA) in which data are stored as attribute value pairs on separated lines.

```
-----------------------------------------------------
align ID:   svenprf206
sample:     22|2
word:       en
link:       en -> a
link type:  regular
unit type:  single -> single
source:     22|2
target:     23|1
source text:##svenprf206## Det \"ar en framg{\"aa}ng.
target text:##svenprf206## This is a success for our policies.
-----------------------------------------------------
```

14.8 What are Collections?

The 'Collection' format is used to combine streams. Collections may include streams that store data in different formats and at different locations. Collections are organized as a sequence of streams that can be read continuously.

Index