Adelheid 1.0 Demonstration Scenarios

Introduction

Adelheid is tagger-lemmatizer system for historical Dutch. Its current state, version 1.0, is the result of the Clarin-NL (call 1) project Adelheid, in which an experimental tagger-lemmatizer was transformed into a system usable by the general public, specifically via the Clarin infrastructure.

In this manual we give you a guided tour showing how to use the system. During this tour, you will need to access a number of other documents and data files, also available on the Adelheid website (http://adelheid.ruhosting.nl):

- The Adelheid 1.0 Tagger-Lemmatizer Manual, explaining how the Adelheid itself can be used to process texts.
- The Adelheid 1.0 Annotation Tool Manual, explaining how the system output can be inspected and (to some degree) adjusted.
- The Adelheid 1.0 Tagset Manual, explaining how the tags and lemmas added by the system should be interpreted.
- A number of data files, contained in Adelheid10Examples.zip :
    - P065p34101.xml An input file containing a single manuscript.
    - demoallxml .xml An input file containing five manuscripts.
    - demoallraw.txt An input file containing the same five manuscripts, but in text format rather than XML format.
    - demolex.xml An example of an additional lexicon.

Preparations

Before you can play out the scenarios below, you will need to set up the proper environment.

Documentation

Download the manuals and examples listed above from the Adelheid website and, if you so desire, print them.

Data

Create a working folder on your machine. Then download and unpack Adelheid10Examples .zip into that folder.

System access

To access the system you will need an internet browser. Any recent browser ought to work. If you encounter difficulties, please let us know (hvh@let.ru.nl). Older browsers are more likely to run into trouble; before contacting us, please first try to update to a newer browser.

You will also need permission to access Adelheid's web applications. When approaching http://lux17.mpi.nl/adelheid/main/, you will be asked to choose an institution and then provide your username and password for that institution. If you do not have any such username and password, you can also apply for a so-called "homeless" account. An official procedure for obtaining such an account is still being developed. However, for now, we advise you to go to http://www.clarin.eu/user/register and apply for a Clarin account there (even if you are not part of an institution that is a Clarin member and do not want to join a Working Group). The username and password for that account can also be used to access Clarin services like Adelheid.

Scenario I: Processing a manuscript

In this first scenario, we show the most frequent use of Adelheid: annotating a manuscript and inspecting the results. Please go through the following steps:

1. Take the Tagger-Lemmatizer Manual. Skip the section on file formats (since you will be using our examples as input) and proceed to Activating the System in The Adelheid Tagger/Lemmatizer.
2. Go through the instructions there until you get access to “your” project list.
3. Create a new project. Please incorporate your name in the project name.
4. Now follow the instructions on providing input to the system. You should choose XML Input from the menu and then lead the system to the file P065p34101.xml on your machine.
5. Press the Start button and wait for results. This may take a while. You do not need to keep your browser open on this page, but can close it and return later.
6. After some time, a number of output files will be listed on your screen.
7. First open P065p34101.tag to see whether it contains any sensible output (it should). You should see four columns with tokens, lemmas and tags. The format is explained in more detail in the section on file formats which we advised you to skip.
8. Go back to the output file list, using the Back-button of your browser.
9. Now open P065p34101_atl.xml. This shows the full output of Adelheid. At the top, you will find the DTD, then the header and even further down the tokens with their annotation. This output is not really meant for human consumption.
10. Go back to the output file list, using the Back-button of your browser.
11. Now download P065p34101_atl.xml to your work folder on your own machine.
12. Leave the Adelheid system.
13. Take the Annotation Tool Manual. Skip the section on file formats (since you will be using our examples as input) and proceed to the chapter “The Adelheid Annotation Tool”.
14. Go through the instructions there until you get access to “your” workspace.
15. Proceed following instructions and upload the file P065p34101_atl.xml which you have downloaded from Adelheid above.
16. Work through the rest of the chapter, trying out the various ways of inspecting the annotation, but stop when entering the section “Adjusting annotations”. Wait with adjusting the annotation until the next step. Once you have explored the viewing options, proceed to…
17. Now read the section “Adjusting annotations” and then make the following changes (note that these are by no means all errors the system makes):
    - The sixth word “letteren” is tagged as plural instead of singular.
    - A few lines down, “comsente” should have lemma “consent”.
    - Towards the end of the manuscript, there is another “letteren”. It is again tagged as plural, but as you will see this time with only a slightly higher probability than singular (0.51 versus 0.49). Here too it should be singular.
18. Check that “linen” and “laken” do not provide the tag-lemma combinations that are desired (linnen Adj() and laken N(sing)). This means that these interpretatiosn are absent in the Adelheid lexicon. In the next scenario, you will see how you can add this to the lexicon.

19. Now leave the annotation tool by clicking Log out (top right) and closing the browser.

This concludes the first scenario.

Scenario II: Adding your own lexicon

In the second scenario, we show how you can provide your own lexicon in order to fill gaps in Adelheid's own lexicon. Please go through the following steps:

1. Read the section "Customizer lexicon" in the Tagger-Lemmatizer Manual and inspect the example lexicon file demolex.xml.
2. Activate the Adelheid system and go to the project you created in the first scenario.
3. Click on the button Discard output and restart. You are led back to the page where you can provide input. The file P065p34101.xml is still present as XML input.
4. Now follow the instructions on providing a lexicon to the system. You should choose Additional lexicon from the menu and then lead the system to the file demolex.xml on your machine.
5. Activate the tagger, move the output to the annotation tool and inspect the results as in Scenario I. Check that "linen" and "laken" are now also annotated correctly (or at least have the desired tag-lemma combination among the potential choices).
6. Now leave the annotation tool by clicking Log out (top right) and closing the browser.

This concludes the second scenario.

Scenario III: Processing several manuscripts at once

In the third scenario, we show how you can process several manuscript at once. We also show the annotation tool's search functionality. Please go through the following steps:

1. It is not possible to provide more than one input document to Adelheid. You can, however, place more than one manuscript in a single document. Open the example lexicon file demoallxml.xml and have a look to see how. Start at the bottom and notice how the text of the various manuscripts is distinguished with <manuscript> markers. Also notice that each manuscript is identified with a manid attribute. Then scroll up and notice that the teiHeader has a general part, describing what is common to all manuscripts, and within the <sampleStmt> in the <profileDesc> a sequence of individual manuscript headers, describing information belonging to each of the manuscripts. At this point you could also read the section XML format in the Input files part of chapter Input and output file formats of the tagger manual.
2. Activate the Adelheid system and create a new project.
3. Go to the new project and provide demoallxml.xml as XML input. You can choose between adding demolex.xml or just using Adelheid's own lexicon.
4. Activate the tagger, move the output to the annotation tool and inspect the results as in Scenarios I and II. As you will see, you can now choose which manuscript you want to inspect in the annotation tool. It is only possible to inspect one at a time.
5. In the Text view, select View options and activate Precede search matches with the "@" sign.
6. Go back to the document level by clicking on the document name (top left).
7. Read the section "Search" in the annotation tool manual.
8. Let us start with a simple search: type letteren in the search window and do not change the choice in the criterion menu (leaving it as Form). Then click on the button Search in document.
9. As you see, there are ten matches, two in each manuscript. We have seen above that Adelheid tends to tag these as plural while in these manuscripts it should generally be singular. Through the search mechanism we can easily visit all occurrences and correct this. Enter the first manuscript, O178p36601, by clicking on the appropriate Edit this manuscript button.
10. Notice that the matches are highlighted and, if you switched this on in the View options, that a @ is appended to them.
11. Now activate your browser's Search on this page function, e.g. ctrl-f in Chrome, and serach for @. This will lead you to the next search match. For each match, correct the tag if necessary.
12. Another example for systematic checking and correcting is the interpretation of specific verb forms. Some verb forms are ambiguous between participle and finite or even infinitive. Since the correct choice is often dependent on a longer distance context, it is wise to inspect and where necessary correct Adelheid's choices. A search for such forms is slightly more complicated. The two main criteria are on the alternative tags (atag in the menu). One demands a particle: V\(particle . Notice the backslash in front of the bracket; brackets have a special function in regular expressions and literal brackets have to be escaped. The second demands a finite tense (fin) or infinitive (infin). Both can be specified together: V\(.*fin . Somewhere in the tag the sequence fin has to be present. In order to limit our search to those cases where Adelheid has at least decided that a verb form should be selected, we add a third criterion on the tag (not the

atag): V\(. Type the regular expressions in the search windows, after twice clicking add more criteria, and choose atag and tag in the appropriate menus.

13. Now activate the search. You will notice that this takes longer than the first search since many more fields have to be inspected and the .* also slows things down. When the search ends, explore the matches.

14. Then leave the annotation tool by clicking Log out (top right) and closing the browser.

This concludes the third scenario.

Scenario IV: Processing heritage material

Although we strongly advise to use XML files as input, we fully understand that many people still have manuscripts in raw text format. In the fourth scenario, therefore, we show how to process such files after only minimal editing. Please go through the following steps:

1. Inspect the file demoallraw.txt to see what a raw text format should look like. At this point you could also read the section Text format in the Input files part of chapter Input and output file formats of the tagger manual.
2. Go to the new project and provide demoallraw.txt as Text input. You can choose between adding demolex.xml or just using Adelheid's own lexicon.
3. Activate the tagger, move the output to the annotation tool and inspect the results as in the previous Scenarios. The annotation should be no different than that of demoallxml.xml.

This concludes the fourth scenario.

Scenario V: Replacing system components

In this fifth and last scenario, we show how you can replace system components such as the tokenizer by alternative web services. Please go through the following steps:

1. Read the section “External tokenization” in the Tagger-Lemmatizer Manual.
2. Activate the Adelheid system and go to the project you created in the first scenario.
3. Click on the button Discard output and restart. You are led back to the page where you can provide input. The file P065p34101.xml is still present as XML input.
4. In the text window next to Tokenisation URL near the bottom of the screen, type http://lux17.mpi.nl/adelheidws/nontokenizer/
5. Activate the tagger, move the output to the annotation tool and inspect the results as in Scenario I. The normal tokenizer erroneously merged the tokens *de* and *feeste* across a newline, very creatively assigning the lemma *diefsteeg* to the result. The nontokenizer does not attempt any adjustments from the tokenization provided by the transcriber. You can see that this is true by checking that *de* and *feeste* now stay unmerged.
6. Now leave the annotation tool by clicking Log out (top right) and closing the browser.

This concludes the fifth scenario.