



Discourse Network Analyzer (DNA)

Manual

Philip Leifeld*

March 18, 2011

(valid for version 1.25)

*Max Planck Institute for Research on Collective Goods, Kurt-Schumacher-Str. 10, 53113 Bonn, Germany — Leifeld@coll.mpg.de — <http://www.philipleifeld.de>

Contents

1	Introduction	4
2	Installation	5
3	Tutorial for beginners	6
	Step 1: Opening the sample file	6
	Step 2: Navigating through articles	6
	Step 3: Editing statements	6
	Step 4: Navigating through statements	8
	Step 5: Creating statements	8
	Step 6: Exporting co-occurrence networks	9
	Step 7: Exporting affiliation networks	10
	Step 8: Exporting statements to a CSV list	10
4	In-depth description	12
4.1	File format	12
4.2	Dealing with articles	13
4.3	Recoding statements	14
4.4	Regular expressions highlighter	15
4.5	Within-actor contradictions	15
4.6	The bottom bar	16
	4.6.1 Full-text search	16
	4.6.2 Organizations	17
	4.6.3 Persons	17
4.7	Exporting time series statistics	18
	4.7.1 Inclusion of actors and categories	18
	4.7.2 Time period	18
	4.7.3 Time steps	18
	4.7.4 Duplicate filter	18
	4.7.5 Type of actor	19
	4.7.6 Aggregation	19
4.8	Exporting network data	19
	4.8.1 Algorithm	19
	4.8.2 Export format	23
	4.8.3 Time period	24
	4.8.4 Normalization	24

4.8.5	Agreement	25
4.8.6	Exclude lists	25
5	rDNA. A Package to Control Discourse Network Analyzer from R	26
5.1	Motivation	26
5.2	Functionality	26
5.3	Examples	27
6	FAQ: Frequently Asked Questions	28
6.1	Reporting bugs	28
6.2	UTF-8 character encoding	28
6.3	Importing large amounts of articles	28
6.4	Java Heap Space Error	29
6.5	Unsupported Class Version Error	29
6.6	Strange symbols in exported CSV files	29
6.7	Can I use DNA on a Mac?	30
6.8	“Could not find the main class.”	30
6.9	DNA is slow and requires lots of computing power	30
6.10	Other QDA software	31
6.11	Cannot open exported CSV files	31
6.12	Exported CSV files are messed up	31
6.13	Which network format do you recommend?	31
7	Version history	33

1 Introduction

DISCOURSE NETWORK ANALYZER (DNA) is a Java-based software for qualitative, category-based content analysis (QDA) with a special emphasis on the extraction of quantitative network data. Its main purpose is the actor-based analysis of political discourses. Yet it may be used for other applications or paradigms.

DNA involves two steps: coding statements of actors into categories, and converting these structured data into networks that are readable by UCINET, *Visone*, NETMINER, GEPHI and other network-analytic software packages. The software can extract either bipartite (affiliation) networks or adjacency networks.

It is complementary to semantic network analysis applications because it neither relies on algorithms for automatic text processing, nor does it focus on the internal meaning or mental representation of a single text or document. Instead, tags are applied to the text data by manual inspection, thus rendering it more flexible, yet at the same time more work-intensive. DNA can be used to code a large body of text documents and then convert them into graphs.

More information about the software can be found in [Leifeld \(2009, 2010\)](#) and [Leifeld and Haunss \(2010\)](#). The underlying conceptual model is elaborated in the book chapter of [Janning et al. \(2009\)](#). The beginner's tutorial in chapter 3 is based on empirical data collected by [Fisher \(2009\)](#).

If you would like to be informed about updates, please subscribe to the [DNA-help] mailing list at <http://www.philipleifeld.de>.

2 Installation

First, download the most recent version of the DISCOURSE NETWORK ANALYZER from <http://www.philipleifeld.de>. Then make sure you have a recent version of the Java Runtime Engine (JRE) installed. Recent means version 1.6 or newer. If this is not the case, please download the newest Java version from <http://www.java.com>. Please do not try to run DNA with an older Java version because some important features will not work properly. DISCOURSE NETWORK ANALYZER has been tested on Windows XP (SP 1), Windows Vista, Windows 7, Ubuntu Linux 10.4, MacOS Leopard and MacOS Snow Leopard. However, it should work on any other operating system for which Java is available. Please check the FAQ in section 6 if you encounter problems.

Some Java installations offer to start the software via double-clicking the file or right-clicking and selecting an entry from the context menu. However, the preferred way of starting DNA is from the command line by executing the command `java -jar dna-1.xx.jar` (where `xx` should be replaced by the current version number) from the folder where you have saved the file.

On Windows XP, you can do this as follows: Save the current version of DNA on your desktop. Go to the start menu and click on **Run**. A small window will pop up. Enter `cmd` into the text field of the window and press enter. A black window with the command line will be opened. Enter `cd Desktop` and press enter. Enter `java -jar dna-1.xx.jar` (replace `xx` by the current version number!) and press enter. DNA will now be loaded.


On Unix-based machines, you should be able to use the same command on the shell to start DNA.

Do not close the command line window because it will display important messages and possibly warnings or error messages. If you encounter an exception or error message and don't know what you did wrong, please submit the error log from the command line by e-mail! Thank you very much!

3 Tutorial for beginners

This section covers many of the basic coding concepts of DNA and the most important network export options. More detailed descriptions of the options will be given in chapter 4 right after this hands-on tutorial. To learn how to use the DISCOURSE NETWORK ANALYZER, please follow these eight steps precisely:

Step 1: Opening the sample file

Figure 3.1 shows the main window of DNA where newspaper articles, interviews, position papers, parliamentary protocols or other qualitative material can be inserted. Download the file `sample.dna` from the DNA documentation section at <http://www.philipleifeld.de>. This is a sample file with a couple of articles and statements. The articles are actually congressional hearings about the issue of climate change, taken from Fisher (2009).¹ First, open the file by clicking on the  Open DNA file... menu item.

Step 2: Navigating through articles

Once you have opened the file, you will see seven entries in the article table, ranging from January 26 to February 16, 2005. Each entry contains the speech of a business, NGO or government representative. The items in the article table are called “articles” (even though they were not taken from the news media in this case). If you click on an article, its corresponding text (i.e. the speech) will be displayed in the text area. Click on the entry “109-1: Callahan, Kateri-NGO-Y” now to see the speech of *Kateri Callahan*, a representative of the *Alliance to Save Energy*.

Step 3: Editing statements

Scroll down the text until you see a yellow text portion. This yellow text portion is called a “statement”. A statement is a text portion where an actor reveals his/her opinion/belief/etc. about an issue. It depends on your theory what kind of information should be coded as statements. In the case of the sample file of Fisher (2009), different

¹The author wishes to thank Dana R. Fisher for contributing the sample data set. More information about the full data set and the political discourse on climate change in the U.S. is provided by Fisher (2009). A full-text archive of U.S. congressional hearings can be found at <http://www.gpoaccess.gov/hearings/>.

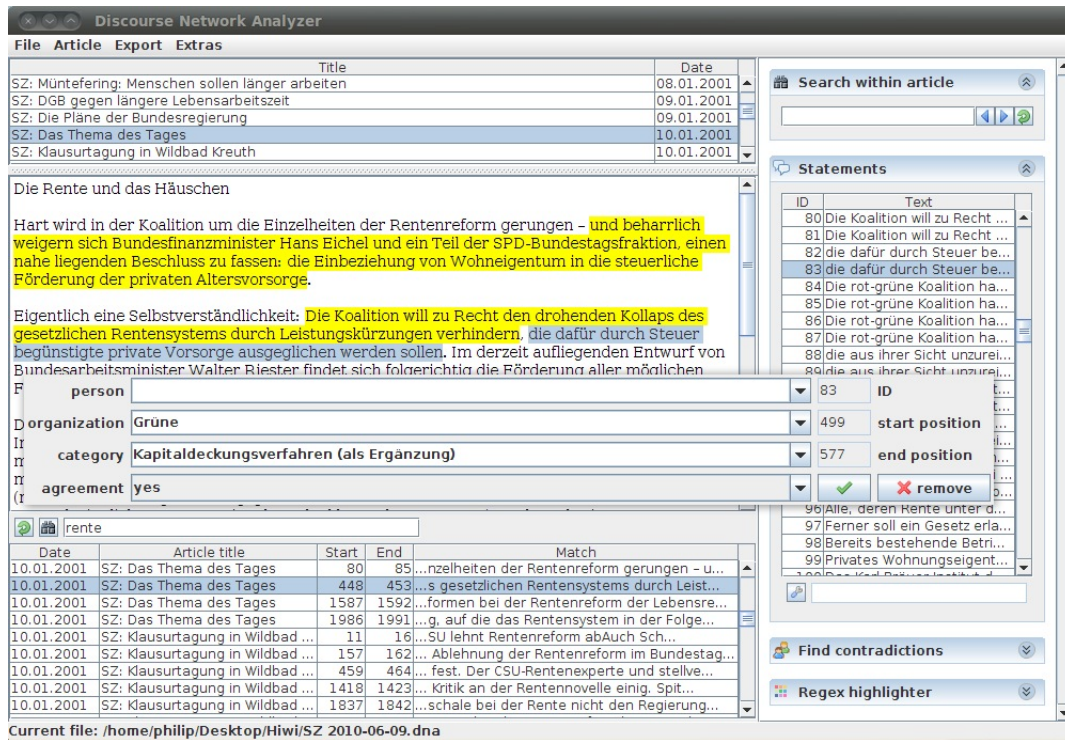


Figure 3.1: The main window

convictions about what should be done about climate change and how were coded as statements. Click on the yellow text starting with “Energy Star...” now. A small window will pop up and reveal the details of the statement. The screenshot in figure 3.1 shows such a popup window roughly at the center of the screen. You will find four editable pieces of information:

- the *person* who makes the statement
- the *organization* the speaker is affiliated with
- the *category* which is raised by the actor (in this case the conviction that “emission legislation should regulate CO₂”)
- a dummy variable indicating whether the actor agrees with the category or not.

It is possible to enter new information or select an actor/category from the combo boxes. The latter show all actor/category entries present in the current DNA file. DNA maintains lists of previous actors or categories on-the-fly, so it is not necessary to re-enter previously coded information. The advantage is an increase in reliability: It would lead to spurious results if typing errors prevented edges between actors or categories to be inferred from the text. Statements may be left incomplete if, for example, the person is not known. The ID field shows a unique identification number that every

statement possesses, and which is not necessarily chronological. It cannot be changed by the user, but it can be used as a primary key if you want to record additional information about your statements in a separate database. If you click on the ✓ button or somewhere outside of the popup window, all changes to the statement will be saved, and the statement popup window will be closed.

Step 4: Navigating through statements

The statement table in the sidebar on the right hand side shows all statements present in the current DNA file, together with their IDs. Click on an arbitrary statement. The corresponding article will be selected from the article table, the statement will be selected in the text window, and the statement popup window will be opened. Close the statement details again. If you click on the small wrench icon ⚙ underneath the statement table and then on **show statements in current article**, the list of statements will be filtered, and only those statements that occur within the currently selected article are shown. If you click on another article, this selection will change. It will always display the statements contained in the current article. The other filter options are based on regular expressions. Click the wrench icon ⚙ again, then **filter by category**, and enter the word “economy” into the text field right beside the button. Only statements with the category “CO₂ legislation will not hurt the economy” will be displayed. The statement filter works analogously for persons, organizations, statement IDs, and the text that is actually coded as a statement. The filters are based on regular expressions (see <http://www.regular-expressions.info> for details and instructions). For instance, if you type “1[2-5]” into the **statement ID** field, DNA will display all statements between 12 and 15.

Step 5: Creating statements

Click on the article of *Christine Whitman* and select some words or lines of text with your mouse. Use your right mouse button to open the context menu. Now click on 🗨 **Format as statement**. The highlighted text will become yellow. Congratulations, you have just created a statement. It should appear in the statement list on the right. If you have not added or removed a statement before, the ID of the statement should be “37”. Click on the statement either in the text or in the statement list to open the statement popup window. Now you can enter the details of the statement or select them from the combo boxes. Since we have created this statement merely for the purpose of illustration, you can safely click on ✖ **remove** and confirm that you want to delete the statement from the file.

Figure 3.2: Network export window

Step 6: Exporting co-occurrence networks

Now you know the concept of a statement and how to deal with it in DNA. Next, we will export our structured data to a network file. This is the main purpose of the software. Click on the **Network export...** entry in the **Export** menu. The network export window shown in figure 3.2 will be opened. If you check the **display help** option, moving the mouse cursor over the various options of the network export window will show small tooltips with instructions. The default options will generate a network where two actors are connected by an edge if they share the same category. The more categories they share, the higher the edge weight will be. This is called a “co-occurrence network” or “congruence network”.

For our example, select the category “There should be legislation to regulate emissions” in the category exclude list. The latter category is a very general one, with most actors agreeing on it. Hence it would not help us to examine the structure of the discourse network. By selecting it, statements with this category are ignored during the export procedure. Now click on **Export...**, enter a file name, and click on **save**. This will produce a DL file, which can be read by UCINET, NETDRAW² and . If you open the DL file in NETDRAW, the network should look like the graph in figure 3.3. You can see two discourse coalitions of actors and one organization taking intermediate positions (i. e. agreeing on categories exhibited by both camps).

²NETDRAW is delivered with UCINET.

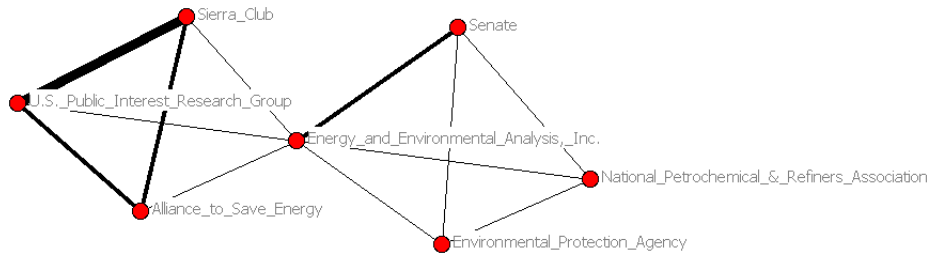


Figure 3.3: Co-occurrence network of organizations

Step 7: Exporting affiliation networks

If you want to understand why this structure looks the way it does, it is advisable to take a look at a (bipartite) affiliation network. While the co-occurrence network shows the relations between actors as a function of their agreement or disagreement with the same categories, an affiliation network shows directly how actors are connected to categories by positive (or negative) relations. To export an affiliation network, unselect the previously excluded category (press **Ctrl** on your keyboard while clicking on the entry), select **Affiliation network** in the **Algorithm** section, and select **yes** in the **Agreement** section. In this network, only positive relations (where agreement is coded as “yes”) are included.³ The resulting network DL file should look like figure 3.4. You can now see that actors from both previously identified camps (the *Environmental Protection Agency*, the *Senate*, *Sierra Club*, and the *Alliance to Save Energy*) agree on having legislation to regulate emissions (this is why we excluded the category in the first place). Furthermore, you can see that *Sierra Club* and the *U.S. Public Interest Research Group* agree on four different categories. This explains the large edge weight in the co-occurrence graph, represented as the wide black line between the two organizations in figure 3.3. To get the full picture, you may want to look at the negative affiliation network (select **no** in the **Agreement** section) in order to be able to reconstruct the co-occurrence actor \times actor network.

Step 8: Exporting statements to a CSV list

Now imagine you would like to have a list of statements in Excel or a similar spreadsheet software. This can be useful to take stock of the overall picture or to combine the statements with other data for statistical purposes. Close the **Network export** window, go to the **Export** menu, select **List of statements**, and enter a file name. You can open the resulting CSV file in any spreadsheet software. It should look similar to the table depicted in figure 3.5.

³Combining positive and negative relations in the affiliation graph would return a multiplex network.

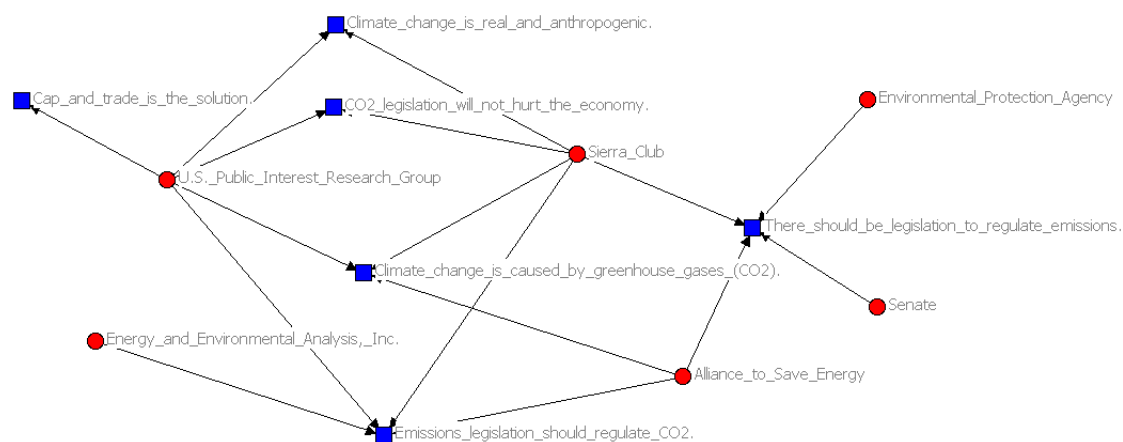



Figure 3.4: Affiliation graph of positive agreement

id	article title	start pos	end pos	date	person	organization	category	agree	text
19	109-867: Whitman, Ch	30629	31330	02/02/05	Christine Todd Whitman	Environmental Protection Agency	There should be legislation to regulate emissions.	yes	Because of the lessc
20	109-867: Whitman, Ch	32376	33538	02/02/05	Christine Todd Whitman	Environmental Protection Agency	There should be legislation to regulate emissions.	yes	If legislation passes
21	109-1: Callahan, Kateri	14870	16451	10/02/05	Kateri Callahan	Alliance to Save Energy	Emissions legislation should regulate CO2.	yes	Energy Star The Ene
22	109-1: Callahan, Kateri	25468	26565	10/02/05	Kateri Callahan	Alliance to Save Energy	There should be legislation to regulate emissions.	yes	Public Benefits Fund
23	109-1: Callahan, Kateri	27916	28443	10/02/05	Kateri Callahan	Alliance to Save Energy	Climate change is caused by greenhouse gases (CO2).	yes	Today, more than tw
24	109-1: Callahan, Kateri	36296	37200	10/02/05	Kateri Callahan	Alliance to Save Energy	Emissions legislation should regulate CO2.	yes	The conference repo
25	109-1: Hamilton, David	1339	1950	16/02/05	David Hamilton	Sierra Club	Climate change is real and anthropogenic.	yes	H.R. 6 does not addr
26	109-1: Hamilton, David	3487	3759	16/02/05	David Hamilton	Sierra Club	Emissions legislation should regulate CO2.	yes	We were very please
27	109-1: Hamilton, David	3929	4552	16/02/05	David Hamilton	Sierra Club	CO2 legislation will not hurt the economy.	yes	We need to be much
28	109-1: Hamilton, David	11444	13260	16/02/05	David Hamilton	Sierra Club	Climate change is caused by greenhouse gases (CO2).	yes	Global Warming--It s
29	109-1: Hamilton, David	19731	21893	16/02/05	David Hamilton	Sierra Club	There should be legislation to regulate emissions.	yes	Prioritize policies tha
30	109-1: Hamilton, David	29190	30034	16/02/05	David Hamilton	Sierra Club	CO2 legislation will not hurt the economy.	yes	In October, 2004, Mr
31	109-1: Nayak, Navin-N	2931	3361	16/02/05	Navin Nayak	U.S. Public Interest Res	Climate change is real and anthropogenic.	yes	We have released a
32	109-1: Nayak, Navin-N	11403	12011	16/02/05	Navin Nayak	U.S. Public Interest Res	Climate change is real and anthropogenic.	yes	Renewable Energy S
33	109-1: Nayak, Navin-N	12174	13102	16/02/05	Navin Nayak	U.S. Public Interest Res	CO2 legislation will not hurt the economy.	yes	Moreover, numerous
34	109-1: Nayak, Navin-N	13775	15155	16/02/05	Navin Nayak	U.S. Public Interest Res	Climate change is caused by greenhouse gases (CO2).	yes	Global Warming Tod
35	109-1: Nayak, Navin-N	20903	21111	16/02/05	Navin Nayak	U.S. Public Interest Res	Emissions legislation should regulate CO2.	yes	The report concludes
36	109-1: Nayak, Navin-N	22109	22237	16/02/05	Navin Nayak	U.S. Public Interest Res	Cap and trade is the solution.	yes	We urge Congress to
44	109-1: Slaughter, Bob	24191	25448	16/02/05	Bob Slaughter	National Petrochemical	CO2 legislation will not hurt the economy.	no	Congress should req
45	109-1: Slaughter, Bob	27682	28566	16/02/05	Bob Slaughter	National Petrochemical	There should be legislation to regulate emissions.	no	Emergency plans, in
46	109-1: Slaughter, Bob	34557	35545	16/02/05	Bob Slaughter	National Petrochemical	There should be legislation to regulate emissions.	no	New Source Review
47	109-1: Slaughter, Bob	35586	35941	16/02/05	Bob Slaughter	National Petrochemical	CO2 legislation will not hurt the economy.	no	Requirements for ad

Figure 3.5: CSV list of statements

4 In-depth description

Figure 3.1 on page 7 shows the main window where newspaper articles, interviews, position papers, parliamentary protocols or other qualitative material can be encoded. You first have to create an article entry by clicking on the  **Add new article...** item in the **Article** menu and entering the relevant information (see section 4.2 for details). The menu also contains functions for opening, closing and saving DNA files (section 4.1), deleting articles (section 4.2), editing the title or date of an article (section 4.2), recoding statements (section 4.3), exporting time series CSV files (section 4.7) or network files (section 4.8), and conducting a full-text search (section 4.6.1).

4.1 File format

DISCOURSE NETWORK ANALYZER is based on an XML file format, which can be read and written by any text editor. Listing 4.1 shows the basic structure of a .dna file:

Listing 4.1: The DNA file format

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <discourse>
3   <version>1.21</version>
4   <metadata />
5   <articles>
6     <article>
7       <text>put the article text here</text>
8       <date>dd.mm.yyyy</date>
9       <title>put the article title here</title>
10      <statements>
11        <statement>
12          <id>numeric statement ID</id>
13          <start>starting caret coordinate</start>
14          <end>ending caret coordinate</end>
15          <person>put the name of the person here</person>
16          <organization>put organization here</organization>
17          <category>put the category here</category>
18          <agreement>yes</agreement>
19        </statement>
20        <!-- you can put more statements here -->
21      </statements>
22    </article>
23    <!-- you can put more articles here -->
24  </articles>
25 </discourse>
```

There may be a situation where you want to convert existing text data with actor and/or category tags into the DNA file format. Given the above XML file format specification, this is not particularly easy because statements are saved separately from the actual text, and caret coordinates are used to locate the beginning and end of a

statement inside each article. In order to make life easier in this kind of situation, there is a second file format which can be read by DNA, but not written. This import file format stores statements directly inside the text. Listing 4.2 shows the structure of the import file format.

Listing 4.2: An alternative format specification for importing converted data

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <discourse>
3   <article day="dd" month="mm" year="yy" title="title">
4     put article text here <statement person="person" organization="organization"
       category="category" agreement="yes"> put statement text here </statement> put
       more article text here
5     <!-- you can put more statements in the text -->
6   </article>
7   <!-- you can put more articles here -->
8 </discourse>



```


Please make sure your imported text is compatible with XML (e. g., it must not contain angle brackets and some other special characters; please look for the XML specification on the internet to obtain more information about incompatible characters and how to replace them by XML-compatible character sequences). Although the file format shown in listing 4.2 is better for conversion, it has some disadvantages (e. g. non-overlapping statements). Therefore, the format shown in listing 4.1 is used to save files by default, and the format in listing 4.2 is only used as an alternative input format.

The functions in the file menu can be used to open, save, save as, and close a .dna file, respectively.

4.2 Dealing with articles

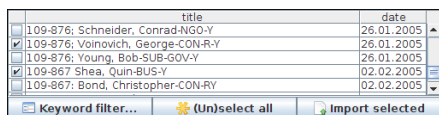
There are four things you can do with articles: import articles from another DNA file, add a new article to the current project file, edit the title or date of an article, and remove an article from the current project file.

The  **Add new article...** menu item opens a new window upon mouse click, where you can enter the title, date and text of a new article. Figure 4.1(a) shows a screenshot of this window. Please make sure you enter the date correctly because otherwise the algorithms for longitudinal data (described in section 4.8.1) will not work properly. The title may include any kind of information, for instance a unique ID if you plan to collect additional information about the articles in a separate database. Duplicate article titles are not allowed. You can copy your article from a website or any other text source and paste it in the text field. Single line breaks are automatically removed, while double line breaks (paragraph breaks) are preserved. Some escape sequences and special characters are automatically removed when text is inserted. The  **Delete selected article** function will completely remove an article from the DNA file, including its statements.

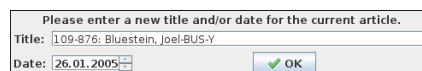
If there is more than one person working on the same project at the same time, it is usually difficult to merge the files manually later on. To avoid trouble with duplicate statement IDs and article names, the function **import articles** () can be used. A



(a) Adding a new article



(b) Importing articles



(c) Edit title/date

Figure 4.1: Operations on articles

screenshot is shown in figure 4.1(b). You can select the articles from a .dna file that you would like to insert into your current document, and DNA will take care of duplicates, etc. You can also automatically select or unselect all articles by pressing the button in the middle. The left button allows to select articles by keyword filter, i. e., you enter a regular expression (such as `^NYT:`), and DNA will select all articles starting with `NYT:`. Section 4.4 provides more details on regular expressions.

There is also a function in the article menu to **edit** the title or date of the currently selected article (**Rename selected article...**). This will open the window shown in figure 4.1(c). Again, duplicate titles are not permitted.

4.3 Recoding statements

The item **Recode statements...** in the **Extras** menu opens a recoding function for statements. A screenshot is presented in figure 4.2. In the upper part of the window, you can select the statements you want to recode based on the **person**, **organization** and **category** fields. The recode operations specified in the lower part of the window will be applied to all statements matching the selection. If several criteria are selected, the intersection (e. g. all statements with a certain actor *and* a certain category) is selected, not the union (an *or* operator). There are two possible operations: removing all matching statements completely, and recoding them. The recoding operation makes use of wildcards (`%o` stands for the organization, `%p` for the person and `%c` for the category). It is possible to enter any kind of text in any of the three fields; for example entering “Environmental Protection Agency” into the **organization** field will simply rename the organization of all matching statements. In addition to this simple renaming, wildcards can be combined with normal text (e. g. `%o (NGO)` will add a suffix to the

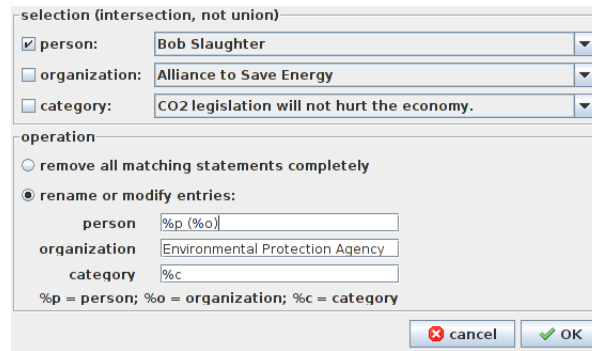


Figure 4.2: Recoding window

organization field). This is particularly useful if some organizational actors should be coded as individuals (e. g. %p (%o)).

4.4 Regular expressions highlighter

The sidebar on the right contains a regular expressions manager. In this panel, you can specify keywords which are highlighted in the text, along with a text color. These keywords are in fact regular expressions, an extremely powerful tool to match actors, categories or any kind of text.¹ The regex function implemented in DNA relies on case-insensitive expressions. The keywords are automatically saved in the `.dna` file. The purpose of the regex highlight function is to assist the user in the encoding of statements by rendering the coding procedure semi-automatic. Coding is much faster if a proper list of keywords is specified in advance, so actors and categories can easily be found when eyeballing the document. Here are some examples:

^EPAS\$ will find the word “EPA” only if it is not part of another word, i. e. “Nepal” is not found.

(G.W.)|(G. W.)|(George W.)|(George)|(George W) Bush will find several name variations.

CO[0-9] will find “CO2”, but also “CO3”, “CO4”, etc.

4.5 Within-actor contradictions

The sidebar on the the right contains a function which searches for self-contradictions of actors (👤 Find contradictions). The results are displayed in a tree view. The tree contains direct contradictions within persons. For example, if a politician uses a category in a positive way in one article and uses the same category in a negative way somewhere

¹The websites <http://www.regular-expressions.info> and <http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html> provide details on how to use regular expressions.

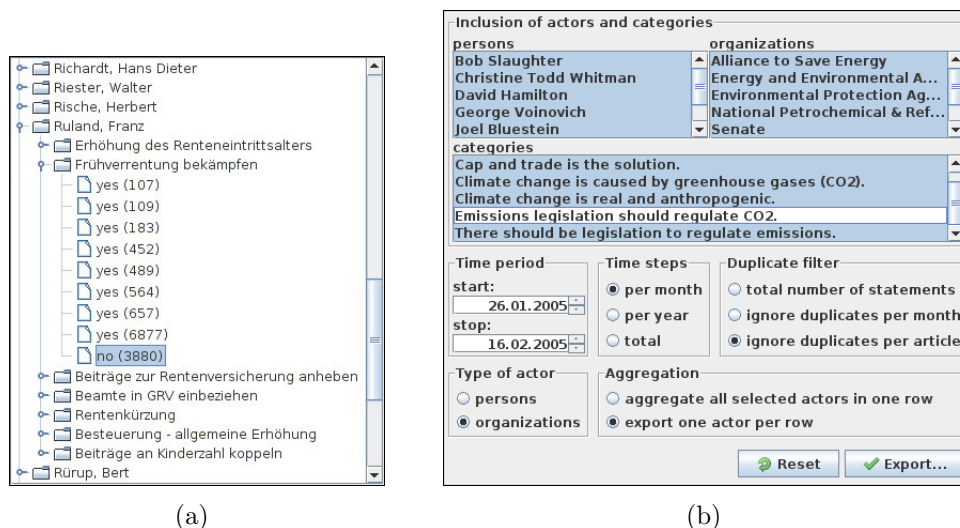


Figure 4.3: Within-actor contradictions and the statistics export component

else, this is counted as a within-person contradiction. The tree lists all contradictions in the file in a hierarchical way. Figure 4.3(a) shows a screenshot.

If you click on a statement in the tree, DNA will highlight the statement in the main window. You can thus use the contradiction report as a navigation instrument. It basically serves to identify either coding inconsistencies or learning behavior of actors over time. You can choose if you want to examine self-contradictions of persons or of organizations.

4.6 The bottom bar

The item **Toggle bottom bar (show/hide)** in the **Extras** menu shows or hides a panel with additional functions: a full-text search function and an attribute manager for actors. The items in the **Extras** menu can be used to specify which function should be shown in the bottom bar.

4.6.1 Full-text search

This is a search panel as shown at the bottom of figure 3.1 on page 7. You can enter a regular expression and search for it in the whole file across articles. A table will display the results. If you click on a result, the corresponding article will be opened, and the matched text will be highlighted.

Actor	Type	Alias/description	Note	
Magix				Liberal MEP
Malcolm Harbour (EPP-ED)	Conservative MEP			Social-democrat MEP
Marek Lazewski	Individual person			Conservative MEP
Maria Berger (PSE)	Social-democrat MEP			Green MEP
Mercedes Echerer (Greens/EF...)	Green MEP			Leftist MEP
Michael Kofler	Individual person			Individual person
Michael Lang	Individual person			
Michel Rocard (PSE)	Social-democrat MEP			
Microsoft				
Monica Frassoni (Greens/EFA)	Green MEP			

add remove selected clean up... add type change... remove type

Figure 4.4: Actor attributes

4.6.2 Organizations

If you activate **Show organizations in bottom bar**, there will be a panel where you can enter additional information about the organizations in the data set (see figure 4.4). There is a field for the **type** of organization, where you can select the actor group this organization belongs to. You can specify the types using the list on the right. A color is associated with each type, and you can change the name or the color of the type at any time in the whole data set. You may, for example, specify the tag “socialist interest group” with a red color and assign this tag to all organizations which have socialist goals. The second field, **alias/description** is reserved for an alternative name of the organization. If you use abbreviations as organization names, it may be helpful to enter the full names of the organizations in the alias field. If you don’t use abbreviations, it may be helpful to add a description. The note field can contain arbitrary kinds of text. Currently, the attribute data are used in three ways:

1. You can export the table of attributes to a CSV file from the **Export** menu.
2. When you open a statement popup window, the organizations in the combo box will be colored according to the selected organization type. You can switch off this behavior in the **Extras** menu by deactivating the option **Use colors for actor selection**.
3. The attribute data are included in the network export when you use graphML files (e. g., for **Vison_E**). This makes it easier to visualize the network. Additionally, the vertex color uses the color of the type tag.

However, other network formats currently do not make use of the attribute data. Support for attribute data in other network export formats may be added in future versions.

You can also add new organizations to the list. As long as an organization is not contained in any statement, it will be highlighted with a red background color. You can delete all of these organizations from the list if you click on **clean up...** or only the selected red organizations by clicking on **remove selected**. Organizations which are currently used somewhere in a statement cannot be removed.

4.6.3 Persons

This is the same as the organizations panel, but for persons.

4.7 Exporting time series statistics

The menu item `Time series statistics...` in the `Export` menu starts the statistics export component shown in figure 4.3(b). The purpose is the extraction of time-series information about statement frequencies. The output will be a table with actors as the rows and time periods as the columns. The cells contain the number of statements of each actor per time period. The resulting CSV file can be read by EXCEL, OPENOFFICE.org CALC or any other spreadsheet software. There are six option panes in the statistics component:

4.7.1 Inclusion of actors and categories

The upper three lists show all persons, organizations and categories in the .dna file. By default, all entries are selected. You can select or deselect entries with your mouse while pressing the `Ctrl` key on your keyboard. If an entry is not selected, all statements containing this entry will be ignored by the statistics export procedure.

4.7.2 Time period

Statements are only included in the statistics export if they are within the range of dates. By default, the first date and the last date present in the .dna file will be used.

4.7.3 Time steps

This option governs whether the statement frequencies will be aggregated into one number per actor (`total`), whether they will be aggregated by `month` (i.e., the number of statements of each actor will be exported for each month separately), or whether they will be aggregated by `year` (counts how many statements each actor has made in each year).

4.7.4 Duplicate filter

The option `total number of statements` simply counts how many statements an actor has made overall in each time period. This implies that repeated statements are counted several times. The option `ignore duplicates per month`, in contrast, will count repeated statements only once per month. As soon as a new calendar month has started, previously excluded statements are counted again, but not more than once per month and per actor. `Ignore duplicates per year` is the same as before, but this time on an annual basis.

4.7.5 Type of actor

If you select **organizations**, this will aggregate the frequencies organization-wise, while the **person** option will aggregate frequencies per person, no matter what organization the person is affiliated with.

4.7.6 Aggregation

By default, frequencies are aggregated for each actor separately. If you want to obtain a simple time series that does not distinguish between actors (i.e., an overall frequency assessment), you should select the option **aggregate all selected actors in one row** for convenience.

4.8 Exporting network data

Clicking on the network export button will convert your qualitative codes into networks. Figure 3.2 reveals the currently implemented export options. They work as follows:

4.8.1 Algorithm

There are currently six different algorithms for network generation.

Affiliation network

This creates a bipartite network, e.g. **organizations**×**categories** or **persons**×**organizations**. An edge is created between a class 1 vertex and a class 2 vertex if there is a statement that contains both of them.

The affiliation algorithm has two options: The **include isolates** option will also include all actors (or categories) that do not have any links to categories (or actors). This may be useful, for instance, if you export several time slices or several relations separately and want to combine the different matrices later on.

The other option is to **ignore duplicates**. If **yes** or **no** is set in the **agreement** option panel (described in section 4.8.5) and if **ignore duplicates** is activated, the resulting network will be binary, i.e. an actor either refers to a category at least once, or (s)he does not refer to the category. This is reflected by the presence or absence of a tie. If **yes** or **no** is set and the **ignore duplicates** option is unchecked, the resulting network will contain weighted edges, and the edge weight reflects the number of times the actor cites the category in a positive or negative way (depending on the **agreement** option).

If **combined** is set in the **agreement** option panel, both positive and negative citations of categories will be counted. However, the behavior deviates from the other algorithm in the following respect: If the **ignore duplicates** options is enabled, the resulting network will be multiplex, i.e. an edge value of 0 represents no connection, a value of 1 indicates that the actor only agrees with the category, a value of 2 reflects only

disagreement with the category, and a value of 3 indicates that the actor both agrees and disagrees with the category. These information can be used in `visone` or `NETDRAW` to apply different colors or line types to the edges. If the `ignore duplicates` option is disabled, positive citations will be added to the edge weight and negative ones will be subtracted. For example, if an actor agrees five times with a category and disagrees seven times with the same category, the edge weight will be -2. Values above 0 indicate overall agreement while negative values indicate overall disagreement.

Unchecking the `ignore duplicates` or the `include isolates` option has no effect when exporting to a CSV list file. This is a simple nodelist format which merely lists the affiliations of a vertex per line. The format can be used to get an overview of different organizational affiliations of individual persons, or vice-versa.

Affiliation networks can also be exported to SoNIA for dynamic network visualization. The result will be a two-mode .son file. Please refer to the subsection on dynamic visualization for more details.

Number of co-occurrences

This creates an adjacency network, e. g. `organizations×organizations` or `categories×categories`. An edge is created e. g. between two organizations if they share a category, or between two categories if they share an actor. The edge weight reflects the number of common items. If, for example, organization 1 refers to three concepts which organization 2 also refers to, an edge with a weight of 3 is created between organization 1 and organization 2. The same is possible for persons or for categories.

There are two options for this algorithm: If you check `include isolates`, actors without any overlap with other actors are included in the network file. This is helpful if you export several time slices and want to combine the files later on. If isolates are not included, it may become difficult to match the matrices because they differ in their lengths. This may also be the case if you export single relations and plan to join them in a multiplex network. The other option, `ignore duplicate statements`, guarantees that edge weights are not overestimated simply because an actor repeats the same statement several times. Edges are essentially *binary per relation*, i. e. if two actors share five different categories three times each, the edge weight will be five, no matter how often the statements are repeated. But if you uncheck the option `ignore duplicate statements`, frequencies are counted, and the edge weight in the above example will be 45 because $3 \cdot 3 \cdot 5 = 45$.

Although the above examples are based on actors, it is as well possible to compute a co-occurrence network of categories, where two categories have an edge weight of w if they are both used by w different actors. Similarly, you can generate an interlocking-directorates network of `persons×persons` via common organizational memberships.

Time window

Statements are context-sensitive. If an actor refers to a category in 1995 and again in 2005, the meaning of the category in the discourse may have changed in the meantime.

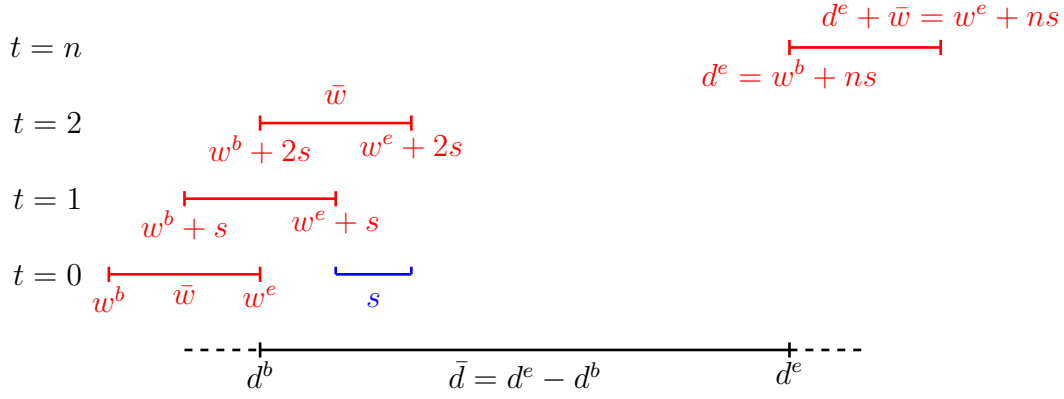


Figure 4.5: The time window algorithm

This leads to an overestimation of edges between actors and eventually to a higher density of the measured network. The time window algorithm offers a solution: It runs a time window of a specified size (the number of days is called the chaining parameter) chronologically through the discourse from the beginning to the end. Whenever two actors make a statement within the time window, an edge is created. If the edge has already been present, the edge weight is increased. If at least one of the statements is made outside of the time window, the link is not established. This guarantees context sensitivity of the resulting network. The drawback is that edge weights may become quite large, so you may consider to normalize or dichotomize the data. A second parameter, the shift parameter, governs the number of days by which the time window is moved at every step. If you prefer discrete, non-overlapping time units, you may set the shift parameter to the size of the chaining parameter (i. e. its maximum value). If you prefer a more precise continuous-time result, you may set the parameter to 1. Figure 4.5 illustrates how the time window is moved through the discourse.

Attenuation

It may be interesting to add another assumption to the model: If actor 1 makes a statement, and a couple of days later, actor 2 makes the same statement, actor 2 is likely to refer to actor 1 by making this statement. It is thus possible to create a directed adjacency graph of referrals. The edge weight between actor 2 and actor 1 is proportional to the probability that actor 2 is really referring to actor 1. Now the trick is to infer this probability from the actual empirical data by using the time that has passed since the statement of actor 1. If, for example, actor 2 makes the same statement immediately on the next day, (s)he is very likely to refer to actor 1, so the edge weight should be large. If, in contrast, actor 2 makes the same statement two years later, (s)he is not very likely to refer to actor 1, so the edge weight should be small. The algorithm adds up the inverse duration since the last statement (e.g. a value of $\frac{1}{7}$ if it was seven days ago) to the existing edge weight, for all pairs of actors over all statements. This produces a very detailed network of referrals. However, you should

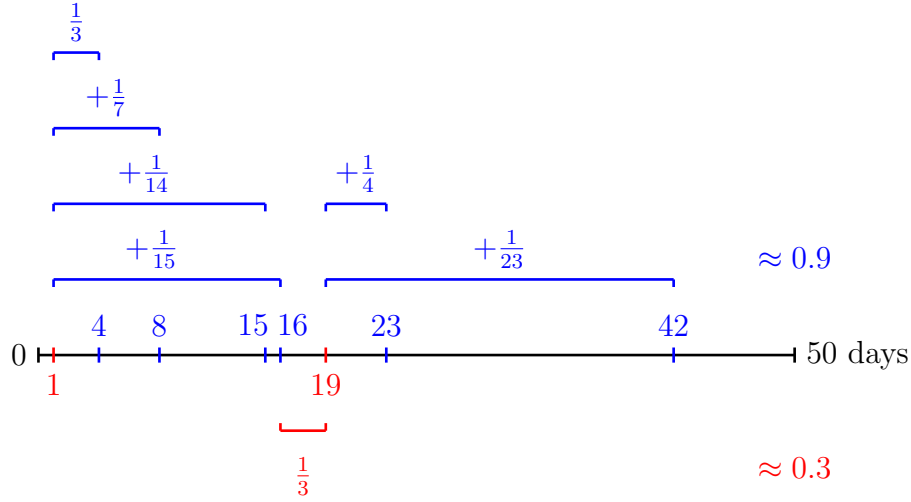


Figure 4.6: An illustration of the attenuation algorithm

be aware of the additional assumptions this kind of network is based upon. Figure 4.6 provides an illustration for the two-player case.

Dynamic visualization

This algorithm generates a list of edges, where each edge is affiliated with a date. If, say, organization 1 makes a statement on the 1st of July, and organization 2 makes the same statement on the 23rd of August, an edge between 1 and 2 is created, and the creation date is the 23rd of August. Once having produced such an edgelist, the network can be visualized in a dynamic, continuous-time way by employing the tool COMMETRIX (<http://www.commetrix.de>). The advantage is that you can now directly observe how the network evolves over time and on what dates interesting changes happen. COMMETRIX also offers a time window visualization where older edges are faded out. Please look at the description of the COMMETRIX file format below in order to understand how you can use it in your project.

For COMMETRIX, there are two parameters: A **backward window** determines, from the perspective of the second statement, how many days to go back in time in order to create edges. In the above example, a backward window of 20 days would not establish an edge between the actor on the 1st of July and the other actor on the 23rd of August, but a backward window of 60 days would. The other parameter is the (arbitrary) **network name** to be used in the COMMETRIX file.

An alternative dynamic visualization software is SoNIA (<http://www.stanford.edu/group/sonia/>). SoNIA is free to use, but COMMETRIX seems to offer much better visualization capabilities for the type of data produced by the DISCOURSE NETWORK ANALYZER. If you use the SoNIA file format, there is an additional parameter, the **forward window**. It determines how long an edge is valid before fading out in the dynamic visualization.

List of statements

This exports a plain list of statements without any network information as a CSV file. Columns include the date of a statement, the person, organization, category, agreement and the contents (i.e. the text that was encoded).

4.8.2 Export format

CSV matrix

A comma-separated value text file with semicolons as column separators. The file will contain a socio-matrix with the complete network and the labels. CSV files can be opened in any text editor or spreadsheet software like EXCEL or OPENOFFICE.org CALC.

DL fullmatrix

A text format for networks. The fullmatrix specification means that the file will contain a socio-matrix with the complete network and the labels. DL files can be imported by the network-analytic software UCINET (<http://www.analytictech.com>) and then exported into a variety of formats from UCINET. DL fullmatrix files can also be opened by the visualization software Visone (<http://www.visone.info>). In some cases, however, Visone does not display the labels properly. In these cases, importing and re-exporting the DL file in UCINET will do the trick.

graphML

The graphML file format is based on XML and is the native format of Visone (<http://www.visone.info>). The specification of the format can be found at <http://graphml.graphdrawing.org>. Please note that coordinates of the vertices will not be exported, i.e. all nodes will at first appear at the same location. You should use layout algorithms to assign X-Y coordinates to the vertices.

Commetrix SQL

This is a text file format that contains data generated by the `edgelist with time stamp` algorithm (see above). Here is how to use it: The result of the export will be a .sql file. You should send this file to the developers of COMMETRIX (<http://www.commetrix.de>). They will – for a small fee – convert the .sql file to a .CMX file and send it back to you. You can use this .CMX file in the CMX Analyzer to visualize your data. Alternatively, you can purchase a license for a tool called CMX Producer from the COMMETRIX developers and create your own CMX files out of the .sql files. This is a little bit more expensive but you may create an unlimited number of CMX files. Please obtain CMX Analyzer and information about the rates and licenses from the COMMETRIX team. Academic or private rates are not very expensive.

SoNIA

This format is similar to the COMMETRIX format and can be used with the dynamic visualization software SoNIA. A disadvantage compared to COMMETRIX is that relative durations are saved instead of absolute time stamps.

CSV list

A comma-separated value text file which can be used for the statement list algorithm. Semicolons are used as column separators. CSV files can be opened in any text editor or spreadsheet software like EXCEL or OPENOFFICE.org CALC.

4.8.3 Time period

Only statements between the start date and the stop date will be considered for export. The default values are the first and the last date found in the .dna file.

4.8.4 Normalization

Edge weights between actors are usually affected by the propensity of an actor to make a statement. For example, a government agency concerned with the labor market is more likely to make a statement about the pension system than an interest group concerned with gender equality. This difference is partly due to institutional roles. As a consequence, some actors will occupy central positions in the discourse network just because they are more mediagenic or because of their institutional role. If you want edge weights to be independent of this, you should use edge weight normalization. There are three different edge weight transformations, depending on the algorithm that you use:

Co-occurrence normalization: Divide the edge weight by the average number of empirically observed statements of the source actor and the target actor, who are involved in an edge.

Time window normalization: This works like the co-occurrence normalization, but on top of that, it divides the resulting value by the number of time steps taken by the time window, and multiply it by 100 in order to get an appropriate scale.

Attenuation normalization: This is the most complicated normalization algorithm. Normalization is already carried out during edge weight calculation. Imagine a situation where actor 1 makes a statement on day 0, and actor 2 repeats the same statement on days 4, 10 and 15. According to the attenuation algorithm, the edge from actor 2 to actor 1 should have a weight of $\frac{1}{4} + \frac{1}{10} + \frac{1}{15} = 0.42$. The normalization option now divides this value by the number of equal statements that actor 2 makes, before adding up the values across statements and actors. We would thus add a value of $\frac{0.42}{3} = 0.14$, which corresponds to the *average* inverse duration between actor 2 and actor 1 for this specific statement. The resulting

values are added up for each statement that is shared by the two actors. This algorithm is far from being perfect, so please contact me if you have ideas on how to improve it.

Please be aware that the normalization algorithms are in an experimental stage, so they may produce biased results.

4.8.5 Agreement

If you select **yes**, only statements where agreement was encoded as “yes” are considered in the export process. If you select **no**, only statements where agreement was encoded as “no” are considered in the export process. **Combined** will produce adjacency networks where actor 1 and actor 2 both agree or both disagree. In the case of the affiliation algorithm, the bipartite graph will contain agreement as well as disagreement relations without making a distinction. The **conflict** option will yield a network where actor 1 agrees and actor 2 disagrees or vice-versa. This is useful for analyzing direct contradictions between actors.

4.8.6 Exclude lists

There are three lists showing all the persons, organizations and categories from the .dna file. You can select items that you want to exclude from the export process. If you do this, all statements that contain excluded items will be skipped during export. To (un)select multiple entries from a list, you can press **Ctrl** while (un)selecting additional items.

5 rDNA. A Package to Control Discourse Network Analyzer from R

rDNA is an R package that allows to integrate the DISCOURSE NETWORK ANALYZER into the statistical programming environment R.

5.1 Motivation

R is becoming increasingly popular as a platform for social network analysis (see, for example, the [statnet project](#)). The focus of DNA is the extraction of network data from manually encoded textual discourses. Bringing together the data-generating software and the methods for processing these data therefore seems quite natural.

Before the development of rDNA, the easiest way to get DNA data into R was to export a network to a CSV file and then import this CSV file in R using the `read.table()` command. This can quickly become tedious because every time even minor parts of the analysis are changed, the data have to be re-exported from DNA and re-imported into R. With rDNA, in contrast, the analyst can directly pull the desired network data from a .dna file without going through the whole user interface and export options of DNA. Or, if desired, rDNA allows to start the coding window of DNA from within R in order to provide an interactive content analysis solution for R (similar to the [RQDA project](#), but with an additional focus on network analysis).

Installation instructions are provided online.

5.2 Functionality

As of version 1.2, rDNA provides seven R functions:

dna.init initializes the connection between R and DNA.

dna.gui starts the user interface of DNA directly from R.

dna.network can import several kinds of networks directly from a .dna file into R (equivalent to the network export function in DNA).

dna.attributes can import the attributes of persons or organizations from a .dna file into R (equivalent to the actor attribute export in DNA).

dna.timeseries computes time series statistics with the options described in section [4.7](#).

dna.density computes the weighted or binary within- and between-block density of sociomatrices. Blocks are given as an optional argument.

dna.categories returns a list of all categories in the .dna file as a character vector.

5.3 Examples

The following R code is an introductory session which briefly demonstrates how rDNA works:

Listing 5.1: Example R code

```
1 # download files and initialize DNA:
2 > library(rDNA) #load the rDNA package
3 > download.file("upload/Downloads/dna-1.25.jar", destfile="dna-1.25.jar", mode="wb")
4 > download.file("upload/Downloads/sample.dna", destfile="sample.dna", mode="wb")
5 > dna.init("dna-1.25.jar") #connect R to DNA
6
7 # plot a congruence network using the statnet package:
8 > congruence <- dna.network("sample.dna", exclude.categories="There should be
9   legislation to regulate emissions.") #create a congruence discourse network
10 > library(statnet) #load the statnet package for network analysis
11 > congruence.nw <- network(congruence) #create network object
12 > plot(congruence.nw, displaylabels=TRUE, label.cex=0.6, pad=0.8) #visualize the
   network
13
14 # do a hierarchical cluster analysis with an affiliation network:
15 > affiliation.yes <- dna.network("sample.dna", algorithm="affiliation",
16   agreement="yes", include.isolates=TRUE) #export positive statements first;
   include isolates because all matrices shall have the same dimensions
17 > affiliation.no <- dna.network("sample.dna", algorithm="affiliation", agreement="no",
18   include.isolates=TRUE) #then export negative statements
19 > affiliation <- cbind(affiliation.yes, affiliation.no) #merge the two datasets
20 > affiliation <- affiliation[rowSums(affiliation) > 0,] #remove isolates
21 > distances <- dist(affiliation, method="binary") #create a dissimilarity matrix
22 > clustering <- hclust(distances) #hierarchical clustering
23 > plot(clustering) #show a dendrogram of the cluster structure
24
25 # open the GUI of DNA in order to manually work on the data
26 > dna.gui()
```

6 FAQ: Frequently Asked Questions

6.1 Reporting bugs

Question: The program behaves in a strange way, and I don't know what is going on. Please help me.

Answer: Please start DNA as proposed in section 2. The terminal window should tell you what is going on. If it reports an exception or error message, please copy the whole output into a text file and submit the problem via e-mail. Please only ask questions via e-mail if this manual does not answer your question or if you get an exception in the terminal window. Please also make sure that you can reproduce the error, otherwise it is unlikely to be fixed.

6.2 UTF-8 character encoding

Question: I am using a non-Western character set, and the characters get messed up after exporting to a DL file. What can I do about it?

Answer: You are probably using UCINET, NETDRAW, *Visone* NETMINER or another network-analytic software package that is not capable of displaying non-Western languages correctly. DISCOURSE NETWORK ANALYZER supports UTF-8 Unicode character encoding by default, and the exported DL files contain the correct characters. The authors of UCINET announced in the summer of 2009 that they would implement UTF-8 support. *Visone* already supports UTF-8, but not on Microsoft Windows systems (it works on Linux though). NETMINER also supports UTF-8, but this can sometimes be tricky. Please complain to the authors of your favorite social network analysis software if you encounter problems with character encoding.

6.3 Importing large amounts of articles

Question: I would like to import many articles at once. Why does DNA not support any *sophisticated* database format like Microsoft Excel?

Answer: DISCOURSE NETWORK ANALYZER supports the eXtensible Markup Language (XML), which is a very versatile database format. Section 4.1 contains a description of the import format. Here is a quotation from Wikipedia for clarification:

“XML’s design goals emphasize simplicity, generality, and usability over the Internet. It is a textual data format, with strong support via Unicode for the languages of the world. Although XML’s design focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services. There are a variety of programming interfaces which software developers may use to access XML data. [...] XML-based formats have become the default for most office-productivity tools, including Microsoft Office (Office Open XML), OpenOffice.org (OpenDocument), and Apple’s iWork.” (source: <http://en.wikipedia.org/wiki/XML>, 03 Dec 2009)

6.4 Java Heap Space Error

Question: I am encountering a Java Heap Space Error. The program now refuses to create statements or display them properly. Is the maximum size of a .dna file limited?

Answer: If your .dna file grows rather big (about 6 MB or > 4,000 statements), you may encounter a “heap space error”, which causes DNA to show some strange behavior. The reason is that your operating system does not provide enough memory to the program, and DNA was not designed to be very efficient regarding its memory usage. You can solve this problem if you start DNA by calling `java -Xmx1024m -jar dna-1.xx.jar` instead of `java -jar dna-1.xx.jar`. This will allocate 1 GB of memory. You can replace the amount by any amount you like (e.g. 512 MB if you don’t have enough RAM).

6.5 Unsupported Class Version Error

Question: Upon start, the error message `java.lang.UnsupportedClassVersionError: Bad version number in .class file` is displayed. What can I do?

Answer: Please make sure you are using a recent Java version. If you try the command `java -version` instead of `java -jar dna-1.xx.jar`, you will see the version number. If it is below 1.6.0_0, please upgrade your Java installation. You can obtain the most recent version from <http://www.java.com>. If you are actually running an up-to-date Java version and you still encounter the error message, please report this bug via e-mail and include the name and version of your operating system in the e-mail.

6.6 Strange symbols in exported CSV files

Question: I have exported my data to a CSV file, but some strange symbols appear instead of ä, ö, ü, ß and some non-Western characters. How can I avoid this?

Answer: The exported CSV files are encoded in UTF-8, which is the default character encoding of Linux. The problem occurs if your operating system or your editor (for example Microsoft Excel) does not know that you are trying to open a UTF-8 encoded file. In Windows, you can convert the CSV file to an ISO-8859-15 or CP-1252 (=ANSI) file by opening it in Notepad (or another editor), saving it as another file, and selecting “ANSI” as the character encoding before saving it. However, ANSI only supports Western characters. It is recommended to change the character set of Excel (or the software you are using) to UTF-8 instead.

6.7 Can I use DNA on a Mac?

Question: When trying to open DNA on a Mac, I get an error message. What can I do to use DNA on a Mac?

Answer: DISCOURSE NETWORK ANALYZER does in fact work on a Mac. It has been tested successfully under MacOS 10.5 “Leopard” and MacOS 10.6 “Snow Leopard”. However, Leopard comes with Java version 1.5 pre-installed, so you have to update Java before it is possible to run DNA. You can do this by executing the built-in software update of MacOS. One case was reported where this update caused trouble because two different Java versions, 1.5 and 1.6, were installed at the same time, and the former was used to start DNA, which of course failed. Generally, updating MacOS “Leopard” to the latest Java version should be hassle-free though. In any case, MacOS 10.6 “Snow Leopard” comes with Java 1.6, so there should be no problem. Please also take a look at FAQ entries [6.5](#) and [6.8](#).

Please note, however, that rDNA is not working on MacOS right now (except for the `dna.network()` method. This may be fixed in a future version.

6.8 “Could not find the main class.”

Question: I try to open DNA by double-clicking on the .jar file. But I get an error message saying “Could not find the main class. Program will exit.” What is happening?

Answer: Your Java version is probably outdated. If you start DNA on the command line, you should see an “UnsupportedClassVersionError” message instead. Please refer to FAQ entry [6.5](#) to solve this problem.

6.9 DNA is slow and requires lots of computing power

Question: Why does DNA get very slow and use lots of memory and CPU when my articles are very long?

Answer: When you select an article, DNA tries to highlight statements and regular

expressions in the text. If the text is very long, this will take considerable time. This can, for example, occur if you encode parliamentary protocols like Congress Hearings, or when you work with interview transcripts. You may want to subdivide your documents into smaller units to solve this problem. DISCOURSE NETWORK ANALYZER was originally designed for the analysis of newspaper articles, which are usually not very long.

6.10 Other QDA software

Question: Other software packages for qualitative analysis also support network analysis. Why should I use DNA though?

Answer: There are one or two other QDA packages available which support network analysis. However, their network functionality is very limited. They only focus on categories, not on actors. They do not distinguish between positive and negative agreement patterns. Moreover, they do not support export into network formats or dynamic algorithms. Usually, they use SNA merely as a tool to organize your thoughts, rather than to analyze discourse.

6.11 Cannot open exported CSV files

Question: I am not able to open my exported CSV file in a spreadsheet software like Excel. Everything is in one table cell.

Answer: You have not properly specified the field separator in your software (e.g., Excel). Each new cell is separated by a semicolon (;). Try to find the CSV import options in your application.

6.12 Exported CSV files are messed up

Question: My exported CSV file is messed up. Data are separated in the middle of a single field.

Answer: Please make sure that your persons, organizations, categories and the encoded text do not contain a semicolon (;). The semicolon is used as a field separator.

6.13 Which network format do you recommend?

Question: There are CSV, DL and graphML files plus several dynamic formats. Which format should I use?

Answer: The best implementation is provided for .graphML files and the software *Vison*. The .graphML export format includes actor attributes, vertex shapes, vertex colors

and edge colors. You can download `visone` from <http://www.visone.info>. For dynamic analyses (i.e., network animations), COMMETRIX works best. Please refer to section [4.8.2](#) for details.

7 Version history

DNA 1.25 – 18 March 2011

- New feature: `dna.categories()` method in rDNA.
- New feature: Added `invert.persons`, `invert.organizations`, `invert.categories` and `verbose` arguments to `dna.network()` and `dna.timeseries()` in rDNA.
- Bugfix: “Include lists” in the time series statistics component had no effect in DNA.
- Bugfix: Normalization did not work when exporting networks (neither in DNA nor in rDNA).
- Bugfix: Fixed `separate.actors` argument in the `dna.timeseries()` method in rDNA.

DNA 1.24 – 23 November 2010

- Bugfix: Recode function caused an exception when either the person, organization or category list in the statement popup window was empty.
- Improved: The time series statistics component was rewritten to make it compatible with rDNA.

DNA 1.23 – 17 November 2010

- New feature: R bindings. DNA can be integrated into the statistical programming environment R via the R package rDNA.
- Improved: Actors and concepts are now put in quotation marks when being exported to a CSV file.
- Improved: The search field within articles listens to enter key presses.
- Improved: Menu item **Exit** was added to the file menu.
- Bugfix: Exporting actor attributes caused an exception when there were actors without an associated actor type.
- Bugfix: The New Article window was not scrollable.

DNA 1.22 – 17 September 2010

- Bugfix: Under some circumstances, statements could not be removed anymore.
- Bugfix: Removing an unused actor from the organization or person attribute manager and subsequently clicking on another actor's entry sometimes caused an exception.
- Bugfix: When closing a file, the regular expressions were not cleared in the sidebar.
- New feature: Document statistics are shown in the sidebar.

DNA 1.21 – 23 August 2010

- New feature: Attributes can be entered for persons and organizations. These attributes include the type of actor (including a color), an alias name or description, and additional notes.
- New feature: CSV export facility for attributes of actors.
- Improved: GraphML export (for `Visone`) has been rewritten. It now includes colors, shapes and actor attributes.
- Improved: The combo box for actor or category selection in a statement popup has been improved. Overwriting entries is more intuitive, and colors can be used to show the type of actor.
- Improved: Exporting a list of all statements is now possible from the **Export** menu (used to be in the **Network export** window).

DNA 1.20 – 17 July 2010

- Bugfix: prohibit two concurrent search threads at the same time
- Bugfix: disable search button while searching
- Bugfix: saving upon close sometimes killed the current file
- Bugfix: regular expressions are now directly highlighted after adding them
- Improved: statement filter takes less space
- Improved: statement list has been replaced by a sidebar where many functions now reside
- Improved: toolbar replaced by a menu
- Improved: collapsible full-text search panel at the bottom of the main window

DNA 1.19 – 15 June 2010

- New feature: Full-text search function across articles.
- New feature: SoNIA algorithm for the dynamic visualization of affiliation networks.
- Improved: DNA can try to repair corrupt .dna files automatically.
- Improved: Resizable layout. DNA can now be run in full-screen mode.
- Bugfix: after clicking on **show all statements**, one can now open and import files without exceptions being displayed.

DNA 1.18 – 5 June 2010

- New feature: Within-actor contradiction report.
- New feature: Keyword filter in the “import articles” function.
- New feature: Affiliation networks can be exported as CSV lists, a nodelist format.
- Improved: Added a progress bar for the “import articles” function.
- Bugfix: The “import articles” function imported the wrong articles.

DNA 1.17 – 28 February 2010

- New feature: Search function inside articles (in the upper right corner of the DNA window).
- Improved: Added an improved progress bar to the network export window.
- Improved: Error messages on the terminal are now shown in red color.
- Improved: The progress log in the network export window was replaced by more detailed progress messages on the terminal.
- Improved: The text window does not jump to the bottom anymore after inserting or removing statements.
- Bugfix: Spaces are not replaced by underscores anymore during DL file export.
- Bugfix: The “import articles” function was broken in version 1.16.
- Bugfix: The **out of memory** error message now works properly.
- Bugfix: Now displaying a more detailed error message when DNA can’t open a file.
- Bugfix: The “network export” window is now loaded faster than before.

DNA 1.16 – 23 February 2010

- Improved: More flexible `recode` window with wildcards and more flexible statement selection.
- New feature: Regular expressions highlighter (can highlight keywords in the text).
- New feature: Progress bars are displayed when saving or loading big files.
- Bugfix: A message is now displayed when Java runs out of memory (i.e. Java Heap Space Error).
- Bugfix: A graphical error during the network export has been fixed. There is now a new status message log.
- The DL `edgelist` file format has been removed.

DNA 1.15 – 31 December 2009

- The `affiliation` algorithm has been modified: If the `combined` option is selected in the `agreement` panel, multiplex networks are now created.
- `Swap actor fields` option added to the `recode` function.

DNA 1.14 – 30 November 2009

- The combo boxes for actors and categories in the statement tooltips now have an autocomplete function. The corresponding list items are automatically selected if you enter some characters.

DNA 1.13 – 30 October 2009

- `ignore duplicates` option for the co-occurrence and affiliation algorithm
- `include isolates` option for the affiliation algorithm

DNA 1.12 – 25 October 2009

- SoNIA export filter
- Article import function
- Statement ID regex filter

DNA 1.11 – 22 September 2009

- Export component for time series/frequency statistics

DNA 1.10 – 17 September 2009

- Statement recoding component
- Export option for isolates in the co-occurrence algorithm

DNA 1.09 – 05 September 2009

- Rewritten coding module
- New data structure with statements saved separately from the text
- Statement list in a sidebar
- Statement filters

DNA 1.08 – 02 August 2009

- CSV list export was corrupt (bugfix)

DNA 1.07 – 15 July 2009

- Filtering hidden control sequences in article (bugfix)
- Improved edgelist export

DNA 1.06 – 04 July 2009

- Redesigned export window
- Commetrix export filter
- Exclude lists in the network export window

DNA 1.05 – 21 June 2009

- Many small improvements in the coding window
- GraphML labels embedded

DNA 1.04 – 18 June 2009

- Preventing user from inserting <, >, and " into text window (bugfix)

DNA 1.03 – 03 June 2009

- Export filters for CSV matrix and DL fullmatrix

DNA 1.02 – 28 May 2009

- Support for Eastern languages was corrupt on non-Unix systems (bugfix)

DNA 1.01 – 17 April 2009

- Export window was not opened when range of dates was < 20 (bugfix)

DNA 1.00 – 16 April 2009

- First public release of the DISCOURSE NETWORK ANALYZER

Bibliography

Fisher, D. R. (2009). Understanding Political Discourse on Climate Change in US Congressional Hearings. Presentation to the US Global Change Research Program in Washington, DC: 28 October.

For more information on this project, see <http://www.columbia.edu/~drf2004>.

Janning, F., Leifeld, P., Malang, T., and Schneider, V. (2009). Diskursnetzwerkanalyse. Überlegungen zur Theoriebildung und Methodik. In Schneider, V., Janning, F., Leifeld, P., and Malang, T., editors, *Politiknetzwerke. Modelle, Anwendungen und Visualisierungen*, pages 59–92. VS Verlag, Wiesbaden.

Leifeld, P. (2009). Die Untersuchung von Diskursnetzwerken mit dem Discourse Network Analyzer (DNA). In Schneider, V., Janning, F., Leifeld, P., and Malang, T., editors, *Politiknetzwerke. Modelle, Anwendungen und Visualisierungen*, pages 391–404. VS Verlag, Wiesbaden.

Leifeld, P. (2010). Political Discourse Networks. The Missing Link in the Study of Policy-Oriented Discourse. Paper Presented at the Workshop “Ideas, Policy Design and Policy Instruments. Casting Light on the Missing Link”, ECPR Joint Sessions of Workshops, Münster, March 22–27. Download: http://www.philipleifeld.de/cms/upload/Downloads/leifeld_ecpr_paper.pdf.

Leifeld, P. and Haunss, S. (2010). A Comparison between Political Claims Analysis and Discourse Network Analysis: The Case of Software Patents in the European Union. Preprint 2010/21, Max Planck Institute for Research on Collective Goods, Bonn. http://www.coll.mpg.de/pdf_dat/2010_21online.pdf.