Inhalt

Vorbemerkung zur Textanalyse in R

Diese Dokumentation beschreibt, wie man von einem Ausgangstext zu den Ergebnissen einer Emotionskategorisierung oder einer Valenz-Arousal-Analyse kommt. Dazu sind verschiedene Zwischenschritte notwendig, die von den Gegebenheiten der Emotionsdiktionäre abhängen. Daher ist dieser Text modular aufgebaut; je nach Emotionsdiktionär muss der Nutzer die Abfolge der Arbeitsschritte selbst festlegen.

Grundsätzlich kann man zwei Typen von Emotionsdiktionären unterscheiden: Die eine Sorte kategorisiert die Emotionen, mit denen die Wörter verbunden sind, die anderen Emotionsdiktionäre verknüpfen Worte mit sog. „Ratings" (Bewertungen). Dies sind Valenz (the pleasantness of the stimulus) und Arousal (the intensity of emotion provoked by the stimulus), aber auch dominance (the degree of control exerted by the stimulus) oder imageability (Vorstellbarkeit / Visualisierbarkeitgrad des Wortes) kommen vor.

Die meisten Emotionsdiktionäre bestehen aus Tabellen, in denen Worte in alphabetischer Reihenfolge in einer Spalte angeordnet und in ihrer Grundform (Lemma) präsentiert werden. Ist das der Fall, dann müssen auch die Worte des zu untersuchenden Textes zum einen in Listenform gebracht werden (alle Wörter finden sich in einer Spalte wieder), zum anderen müssen die Wörter in ihre Grundform gebracht (lemmatisiert) werden.

Emotionsdiktionäre wie das Affektive Diktionär Ulm (ADU) hingegen liegen zwar in Tabellenform vor, die Worte sind aber in sämtlichen Flexionen aufgeführt; hier muss also nicht lemmatisiert werden. Die revised Berlin Affective Word List (BAWL-R) präsentiert die Worte in der Grundform mitsamt Valenz- und Arousalwerten.

In den Emotionsdiktionären finden sich in den Spalten hinter jener Spalte, die die Worte enthält, weitere Spalten, die entweder die Emotionskategorie (ADU), 0 oder 1 für nein oder ja von Emotionstypen (NRC Word-Emotion Association Lexicon EmoLex) oder aber Zahlenwerte für Valenz, Arousal usf. enthalten.

Der Abgleich zwischen Ausgangstext und Emotionsdiktionär kann vorgenommen werden, wenn der Ausgangstext mindestens als Wortliste aufbereitet wurde (alle Worte finden sich in einer Spalte, dies genügt für das ADU); und der ggf. zwei Spalten hinzugefügt wurde, die das Ergebnis der Lemmatisierung enthalten: Eine Spalte für die Wortklassen und eine für die Lemmata, d.h. die Worte in ihrer Grundform (dies gilt für die BAWL-R, das EmoLex und die von Warriner et al publizierten „Norms of valence, arousal, and dominance for 13,195 English Lemmas".

Technisch gesehen funktioniert sowohl die Emotionskategorisierung als auch die Valenz-Arousal-Analyse als Abgleich beider Listen, nämlich der Wortliste des Ausgangstextes mit dem Emotionsdiktionär. Bei allen Worten des Ausgangstextes, die im Emotionsdiktionär erkannt werden, werden alle weiteren im Emotionsdiktionär enthaltenen Informationen (Emotionstyp, Valenzwert o.ä.) in die Tabelle des Ausgangstextes kopiert. Die nichterkannten Worte werden aussortiert und die Ergebnisse können dann in ihren verschiedenen Dimensionen ausgewertet werden.

Installation von R

Die aktuelle Version von R (derzeit 3.4.0, Juni 2017) kann von der CRAN-Website heruntergeladen werden; sie steht für jede Plattform zur Verfügung:

https://cran.r-project.org/

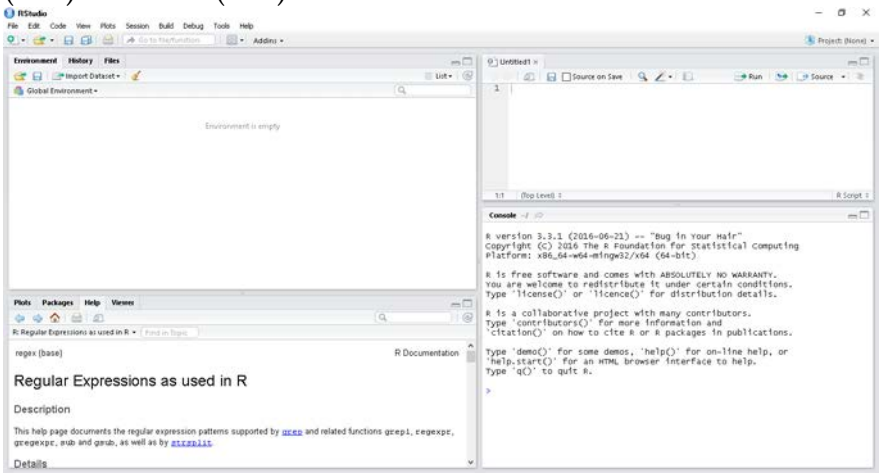
Windowsnutzer klicken auf „base“ und laden sich dann die aktuelle R-Version als .exe-Datei herunter
Mac-Nutzer wählen den link zur neuesten Version aus, die mit dem vorhandenen Betriebssystem kompatibel ist.
Auch für eine ganze Reihe von Linux-Distributionen gibt es installer-files.

Nachdem R auf dem eigenen Rechner installiert wurde, ist es absolut empfehlenswert, auch RStudio zu installieren. RStudio ist eine IDE, d.h. ein „Integrated Development Environment“, verfügt über eine grafische Benutzeroberfläche und läuft auf Windows, Mac und Linux. Hier sollte die „Desktop“ (und nicht die „Server“) Version von der Seite

https://www.rstudio.com/

heruntergeladen werden. Wie bei jedem anderen Programm gibt es Installationsanweisungen, danach kann RStudio z.B. über das Menü „Programme“ oder ein Desktop-Icon gestartet werden.

RStudio zeigt nach dem Start vier Fenster an, die man sich nach Belieben durch einen Klick auf das Icon mit dem Fensterkreuz (neben demSuchfeld) einrichten kann. Es gibt ein script editing Fenster und eine Konsole sowie zwei Fenster für Environment (usf.) und Plot (usf.).



Sehr hilfreich ist die Installation wichtiger Packages in R über das „Tools“-Menü in RStudio: Auf das Menü „Tools“ und dann „Install Packages“. In dem Fenster, das sich nun öffnet, kann nach zu installierenden Pakete gesucht werden. In das Suchfeld z.B. XML (Achtung, Großschreibung beachten!) eingeben, ein Häkchen bei „install dependencies“ setzen und dann auf „Install“ klicken. Das Häkchen bei „Install

dependencies“ sorgt dafür, dass alle weiteren Pakete installiert werden, die benötigt werden, damit die gewünschten Pakete auch funktionieren.

Für die Verwendung der R-Funktionen benötigte Pakete:

koRpus
dplyr
ggplot2
RColorBrewer

Download und Installation benötigen in der Regel nur einige Sekunden; hier tut sich dann in der Konsole von RStudio einiges.

Vorbereitende Schritte

Zunächst muss man in RStudio das Working Directory auswählen. Dies geschieht, indem man in RStudio auf Session>Set Working Directory>Choose Directory klickt. Dann öffnet sich der übliche Filebrowser und man kann ein beliebiges Verzeichnis auswählen; es muss dasjenige sein, in das man die eingesetzte R-Funktion und die zu analysierenden Texte legt.

Um einen Text für die Analyse vorzubereiten, muss man ihn entweder lemmatisieren (für die Verwendung der BAWL-R, des EmoLex und der Warriner-Liste) oder in Listenform bringen (ADU).

Der Grund für diesen vorbereitenden Schritt liegt darin, dass die BAWL-R, das EmoLex und die Warriner-Liste die mit Emotionskategorien bzw. Valenz und Arousal assoziierten Worte nur in ihrer Grundform auflisten. Werden die Worte des Ausgangstexts nicht vorher lemmatisiert, dann kann das gravierende Folgen für die statistische Auswertung des Textes haben – gibt es beispielsweise im Text die Wortformen „traf“, „treffe“, „trifft“ und „treffen“ jeweils einmal, so würde die Grundform „treffen“ bei einem Abgleich mit dem Emotionsdiktionär auch nur einmal erkannt – statt mit einer Frequenz von vier. Liegen im Emotionsdiktionär nur die Grundformen vor, sollte der Anwender daher den von ihm zu untersuchenden Text dann so aufbereiten, dass dessen Worte in der Grundform zur Verfügung stehen. Die lexikographische Reduktion der Flexionsformen eines Wortes auf die Grundform nennt man Lemmatisierung. Bei diesem Vorgang wird auch die Wortart bestimmt; das ist generell nützlich, wenn man analysieren und erläutern muss, welche Worte nicht erkannt und daher aus der Analyse ausgeschlossen wurden.

Es gibt eine ganze Reihe von Werkzeugen zur Lemmatisierung, oft für jede Sprache einen. Ein einfaches und sehr gutes Tool zur Lemmatisierung für Deutsch, Englisch, Französisch und eine Reihe weiterer Sprachen ist der TreeTagger. Er kann hier für jede Plattform nebst den für jede Sprache unterschiedlichen Parameterdateien von der Webseite des Computerlinguisten Helmut Schmid heruntergeladen werden. Eine Installationsanweisung liegt dem Paket bei. Selbstverständlich kann die Lemmatisierung auch mit den klassischen (aber komplexeren) Tools zur linguistischen Annotation – wie etwa dem Stanford Log-linear Part-of-Speech-Tagger oder dem Python-basierten Natural Language Tool Kit NLTK – vorgenommen werden.

Um den TreeTagger effektiv einzusetzen, können sich Windows-Nutzer noch eine grafische Benutzeroberfläche für den TreeTagger herunterladen, die von Ciarán Ó Duibhín entwickelt wurde. Download und Bedienungsanleitung finden sich hier. Dieses GUI legt man in den /bin-Ordner des TreeTaggers und ruft es dann auf, wenn man einen Text lemmatisieren möchte. Es funktioniert sehr simpel. Es gibt zwei Eingabefelder, eines für die Ausgangsdatei (der zu lemmatisierende Text im .csv- oder .txt-Format) und eines für die Ausgabedatei. Hier kann man einen beliebigen Ort auf seinem Rechner angeben und einen Namen für die Ausgabedatei eintippen (mit .txt enden lassen!).

Im Ergebnis erhält man eine Liste, die zuerst das Wort, dann das Kürzel für die Wortart (den Part-of-Speech POS) und dann das Lemma enthält. Das sieht beispielsweise so aus:

During	IN	during
The	DT	the
Last	JJ	last
Hundred	CD	hundred
Years	NNS	year

Eine Auflösung der Wortklassenkürzel für das Englische findet sich beispielsweise hier. IN steht für subordinierende Konjunktion, DT für Determiner, JJ steht für ein Adjektiv, CD für cardinal number, NNS für Nomen in der Pluralform. Und so weiter …

… und natürlich gibt es immer wieder nicht erkannte und problematische Worte. Wer es ganz gründlich machen möchte, sollte daher die vom TreeTagger erzeugte txt-Datei durchsehen und manuell nachkorrigieren.

Nutzer, die die R-Funktion für das Affektive Diktionär Ulm verwenden möchten, gehen ebenfalls wie oben beschrieben vor und lemmatisieren den Text. Da im ADU aber die Worte in flektierter Form vorliegen, werden die Lemmata nicht benötigt, wohl aber die Wortarten (POS). Der Einfachheit halber öffnet man daher die lemmatisierte txt-Datei, kopiert die erste Spalte und fügt sie als dritte Spalte wieder ein (die Lemmata können dabei überschrieben werden, sie werden ja nicht benötigt).

Abgleich eines Texts mit dem Affektiven Diktionär Ulm (ADU)

Um diese Analyse durchführen zu können, muss eine ADU.R-Datei im Arbeitsverzeichnis liegen, ebenso wie die Datei „ADU-unicode-R.csv“. Das Affektive Diktionär Ulm (ADU) ist eine mit Wortliste, die rund 27.000 flektierte Worte umfasst und diese entlang 12 Emotionskategorien kategorisiert. Mehr zu den Eigenheiten dieses Emotionslexikons siehe unten. Schließlich muss im Arbeitsverzeichnis noch eine mit dem TreeTagger getaggte Textdatei liegen; im Ausgangstext muss die Groß- und Kleinschreibung erhalten bleiben. In unserem Beispiel heißt sie xyz.txt.

Als erster Schritt muss das Arbeitsverzeichnis ausgewählt werden. Dies geschieht, indem man in RStudio auf Session>Set Working Directory>Choose Directory klickt. Dann öffnet sich der übliche Filebrowser und man wählt das Arbeitsverzeichnis aus.

Danach gibt man den unten aufgeführten Code in das Source-Fenster von R ein. Anschließend markiert man den gesamten Code (beide Zeilen) und klickt Strg+Return.

```
source("ADU.R")
ADU(chunks=10)
```

Nun öffnet sich ein Pop-Up-Fenster, das das Arbeitsverzeichnis anzeigt. Hier wählt man die Datei xyz.txt aus und klickt „Öffnen“. Der Rest funktioniert automatisch. Im Console-Fenster kann man ein STOP-Hinweisschild sehen, solange der Rechner arbeitet. Wenn er fertig ist, zeigt das Console-Fenster wieder das Zeichen für die Befehlseingabe >.

Diese Befehlskette gleicht den Text mit dem Emotionsdiktionär ADU ab. Alle Ergebnisse sowie zwei Grafiken im .png-Format werden in das Arbeitsverzeichnis geschrieben. Die Ergebnisse bestehen aus

- Einer Grafik, die den Verlauf der erkannten Worte pro chunk darstellt, differenziert nach den 12 Emotionskategorien: xyz-Ratings-per-chunk-Plot.png (Achtung: Hier werden nur die gerateten Worte dargestellt)
- Einer Grafik, die den relativen Anteil erkannten Worte an den 12 Emotionskategorien darstellt: xyz-Global-ratings-Plot.png (Achtung: Hier werden nur die gerateten Worte dargestellt)
- Einer Datei, die globale Angaben über die Gesamtanzahl der Worte im Text, über den relativen Anteil der erkannten Worte nach Emotionskategorie, über den Anteil der erkannten Worte mit oder ohne rating sowie die Häufigkeit der verschiedenen Wortklassen im Text gibt: xyz-ADU_global-summary.csv
- Einer Datei, die sämtliche erkannten Worte sowie die ihnen zugeordnete Emotionskategorie enthält: xyz-ADU_raw-words.csv
- Einer Datei, die die absoluten Zahlen der erkannten Worte je Emotionskategorie pro Textabschnitt (chunk) zusammenfasst: xyz-ADU_summary-statistics.csv
- Einer Datei, die angibt, wie häufig jedes erkannte Wort („token“) im Text auftaucht: xyz-ADU_word-frequencies.csv
- Einer Datei, die beschreibt, welche Worte zwar abgeglichen, aber nicht erkannt wurden und daher keiner Emotionskategorie zugeordnet wurden („unrecognized words“ in der Global summary); diese Datei benötigt man, um beschreiben zu können, welche Worte und insbesondere Wortarten nicht berücksichtigt wurden: xyz-ADU_outtakes.csv

- Eine Datei, die beschreibt, welche Worte aufgrund der Wortart von der Analyse ausgeschlossen wurden (Konjunktionen, Bestimmungsworte (der, die, das), Zahlen, Präpositionen, Pronomen etc.): xyz-ADU_excluded_words.csv

Natürlich gibt es verschiedene Parameter, die man über die Befehlskette einstellen kann. Wenn man z.B. eine andere Größe der Textabschnitte (chunks) haben will, gibt man den Wert entsprechend an; hier 100 Wörter; diese Zahl sollte generell aber gering gehalten werden, denn es werden nur selten mehr als 3% aller Worte erkannt:

```
source("ADU.R")
ADU(chunks=100)
```

Oder man möchte die Grafiken statt im png- im pdf-Format erhalten. Dann lautet die Befehlskette

```
source("ADU.R")
ADU(plot.format="pdf", chunks=100)
```

Wenn man eine Textdatei benutzen möchte, die in einem anderen Verzeichnis als dem Arbeitsverzeichnis liegt, kann man den Pfad so setzen:

```
ADU(text.file = "/the/path/to/your/new/working-directory/INPUTFILE.txt")
```

Wenn man ein anderes Verzeichnis benutzt als das Arbeitsverzeichnis, kann man den Pfad so setzen:

```
ADU(path.wd="/the/path/to/your/new/working-directory")
```

Wenn man den TreeTagger auf seinem Rechner laufen hat und ihn für den gesamten Prozess benutzen möchte, kann man einen Fließtext (xyz.txt) einlesen und dieser wird dann vom TreeTagger lemmatisiert. Die Funktion geht dann mit den Standardeinstellungen davon aus, dass sich der Ordner TreeTagger ebenfalls im Arbeitsverzeichnis befindet und der Text in einer txt-Datei bereitliegt. Der Befehl dafür lautet:

```
ADU(use.TreeTagger=TRUE)
```

Wenn man den TreeTagger in einem anderen Verzeichnis installiert hat (z.B. C:\) und ihn nicht eigens in den Arbeitsverzeichnis kopieren möchte, kann man diesen Pfad auch so setzen:

```
ADU(path.TT="/the/path/to/your/own/treetagger")
```

Wenn man den Pfad zum ADU-Emotionsdiktionär anders setzen möchte, gibt man folgendes ein:

```
ADU(path.lex="/the/path/to/your/own/folder/ADU-unicode-R.csv")
```

Wie immer wird jeder Nutzer dazu aufgefordert, mit den Parametern herumzuspielen und auch verschiedene Auswertungsmöglichkeiten auszuprobieren – beispielsweise einen Text nach Kapiteln analysieren. Dazu muss die xyz.txt-Datei entsprechend aufbereitet werden.

ACHTUNG! Bei jedem neuen Durchlauf in R werden die Ergebnisse neu in das Arbeitsverzeichnis geschrieben; eventuell dort bereits vorhandene Ergebnisse werden

überschrieben! Wer sich also für diese älteren Ergebnisse interessiert, muss sie vor dem nächsten Lauf an einem anderen Ort sichern.

NACH der Analyse in R sind folgende Eigenheiten zu berücksichtigen:
Es gibt im ADU 12 Emotionskategorien, die aber auf 8 reduziert werden können.[1] Das ADU unterscheidet nämlich zwischen me- und it-Emotionen, d.h. worauf sich die Emotion richtet (auf mich, den ‚feeler', oder auf ‚es', das z.B., wovor ich Angst habe). Für wen diese Unterscheidung nicht interessant / relevant ist, der kann die erkannten Emotionswörter auch auf 8 Kategorien reduzieren. Das Kategorienschema des ADU sieht folgendermaßen aus:

POSITIVE: 1-4

POSITIVE IT: 1-2

1 Liebe (1725 Worte)

2 Begeisterung (2191 Worte)

POSITIVE ME: 3-4

3 Zufriedenheit: (1407 Worte)
3a Zufriedenheit (1187 Worte)
3b Erleichterung (220 Worte)

4 Freude: (3600 Worte)
4a Freude (2741 Worte)
4b Stolz (859 Worte)

NEGATIVE: 5-8

NEGATIVE IT: 5-6

5 Zorn (3269 Worte)

6 Furcht (3530 Worte)

NEGATIVE ME: 7-8

7 Depressivität: (5081 Worte)
7a Depressivität (4912 Worte)
7b Schuld (169 Worte)

8 Ängstlichkeit: (4419 Worte)
8a Ängstlichkeit (2866 Worte)
8b Scham (1553 Worte)

Das zweite, was nach der Analyse berücksichtigt werden muss, ist eine mögliche Verfälschung durch die Groß- und Kleinschreibung. Steht nämlich ein Wort am Satzanfang, dann wird es ja groß geschrieben, aber vom Diktionär falsch erkannt: „Würde er ihr einen Blumenstrauß geben?" – In einem solchen Fall wird das Wort „Würde" falsch als Substantiv erkannt und in die Emotionskategorie „Stolz" einsortiert. Hilfreich bei dem Ausmerzen von

[1] Ausführliche Erläuterungen hierzu finden sich bei: Michael Hölzer, Nicola Scheytt, Horst Kächele, "Das 'Affektive Diktionär Ulm' als eine Methode der quantitativen Vokabularbestimmung." In: Textanalyse. Anwendungen der computerunterstützten Inhaltsanalyse, Wiesbaden: VS Verlag für Sozialwissenschaften 1992, S. 131–154.

falsch erkannten Worten ist die Datei „…ADU_raw-words.csv“, denn sie enthält sämtliche erkannten Worte sowie den vom TreeTagger vergebenen „tag“ – und der wäre im gegebenen Beispiel nicht der Tag für Substantive (NN).

Weiterführende Literatur und Hilfe

Cookbook.for R
http://www.cookbook-r.com/

Fridolin, Wild. CRAN Task View: Natural Language Processing.
http://cran.r-project.org/web/views/NaturalLanguageProcessing.html

Newest 'r' Questions - Stack Overflow.
http://stackoverflow.com/questions/tagged/r

Quick-R: Home Page.
http://www.statmethods.net/

R 1.1 - Initial Setup and Navigation - YouTube.
http://www.youtube.com/watch?v=iffR3fWv4xw

R: Mailing Lists
http://www.r-project.org/mail.html

R Programming - Wikibooks, open books for an open world.
http://en.wikibooks.org/wiki/R_Programming

Revolutions.
http://blog.revolutionanalytics.com/atom.xml

RSeek.org R-project Search Engine.
http://www.rseek.org/

Rydberg-Cox, Jeff. Statistical Methods for Studying Literature Using R.
http://www.chlt.org/StatisticalMethods/index.html

Try R.
http://tryr.codeschool.com

Videos from Coursera's four week course in R.
http://blog.revolutionanalytics.com/2012/12/coursera-videos.html

Bücher, echt gedruckte Bücher:
Baayen, R. Harald (2008). Analyzing linguistic data: a practical introduction to statistics using R. English. Cambridge, UK; New York: Cambridge University Press.

Dalgaard, Peter (2008). Introductory statistics with R. English. New York: Springer.

Gries, Stefan Thomas (2009). Quantitative corpus linguistics with R: a practical introduction. English. New York, NY: Routledge.

Teetor, Paul (2011). R cookbook. English. Beijing: O'Reilly.

Wickham, Hadley (2009). ggplot2 elegant graphics for data analysis. English. Dordrecht; New York: Springer.