Visualisierungen für CTS Text Miner

Hans Dieter Pogrzeba
Abteilung Automatische Sprachverarbeitung
Institut für Informatik
Universität Leipzig
Augustusplatz 10
04109 Leipzig
pogrzeba@studserv.uni-leipzig.de

Jochen Tiepmar
ScaDS, Universität Leipzig
Ritterstrasse 9-13, 2.OG
04109 Leipzig
jtiepmar@informatik.uni-leipzig.de

Gerhard Heyer
Abteilung Automatische Sprachverarbeitung
Institut für Informatik
Universität Leipzig
Augustusplatz 10
04109 Leipzig
heyer@informatik.uni-leipzig.de

Zusammenfassung

In (Tiepmar, 2016) wurde *CTS Text Miner* (CTS-TM), ein Textmining Framework, welches *Canonical Text Service* (CTS) als Datenquelle verwendet, vorgestellt. In diesem Paper werden verschiedene Visualisierungsansätze beschrieben, die die Ergebnisse der Funktionen und Module von CTS-TM aufbereiten und es dem Benutzer erleichtern, die Daten zu analysieren und Erkenntnisse daraus zu gewinnen.

Keywords: Visualisierung, Text Mining, Canonical Text Service

1. Einführung und Zielstellung

CTS ist ein vom *Homer Multitext Project*[1] entwickeltes Protokoll zur Zitation. Dabei werden sogenannte Uniform Resource Names (URNs) benutzt, um Textstellen eindeutig zu referenzieren. Wie in (Tiepmar, 2015) beschrieben, steht hierbei zum Beispiel die URN *urn:cts:demo:goehte.faust.de:1.2-1.4* für den Textabschnitt in Goethes Faust von Akt 1 Szene 2 bis Akt 1 Szene 4. URNs sind hierarchisch aufgebaut, wobei die exakte Struktur und Tiefe vom jeweiligen Dokument abhängt (Tiepmar, 2015).

In (Tiepmar, 2016) wurde mit CTS-TM ein Textmining Framework vorgestellt, welches als Datenquelle CTS Instanzen nutzt. Ziel hierbei war es unter anderem, Textmining Arbeitsabläufe zu vereinheitlichen, einfach reproduzierbar und vergleichbarer zu machen (Tiepmar, 2016).

CTS-TM wird dazu auf einem Server installiert. Nach Durchlauf der Analyseschritte der jeweiligen Module sind die Ergebnisse über HTML Requests abrufbar und werden als Listen zurückgegeben. In dieser Arbeit werden interaktive Visualisierungen vorgestellt, die diese Ergebnisse graphisch aufbereiten und es dem Benutzer dadurch erleichtern, die Daten zu interpretieren.

2. Vorstellung der Visualisierungen

Die Visualisierungen wurden jeweils als unabhängige JavaScript Web-Apps auf Basis von *d3.js* (Bostock et al., 2011) umgesetzt, wobei Parameter und Einstellungen als URL-Parameter übergeben werden. Von den Web-Apps werden entsprechende Requests an CTS-TM gesendet, wobei angenommen wird, dass die Visualisierung neben einer CTS-TM Instanz platziert wurde. Der Ordnername, unter der diese verfügbar ist, kann über den Parameter *ctstm* angepasst werden.

Generell wurde darauf geachtet, die Visualisierungen möglichst flexibel zu gestalten, damit sie ggf. mehrere CTS-TM Funktionen mit ähnlichem Ergebnisformat abdecken, bzw. auch bei einer zukünftigen Erweiterung oder Änderung der CTS-TM Funktionen benutzbar bleiben. Daher ist auch die Anfrage an CTS-TM nicht fest programmiert, sondern wird meist als Parameter *function* dynamisch übergeben.

Damit die Parameter und Optionen für den Anwender leichter zu verstehen und anzupassen sind, bietet jede Web-App ein Formular (siehe Abbildung 1), über das eine Anfrage mit anderen Parametern gestellt werden kann. Somit bekommt der Benutzer schnell einen Überblick, welche Parameter für die entsprechende Visualisierung relevant sind. Eine Übersicht mit jeweils einer kurzen Erklärung der jeweiligen Parameter befindet sich auch als Textdatei in den Ordnern der Web-Apps.

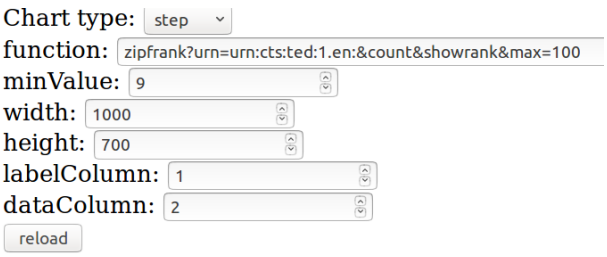

Abbildung 1: Formular für Charts Visualisierung.

Ein weiteres Ziel beim Erstellen der Visualisierungen war es, dem Betrachter die Möglichkeit zu geben die bereitgestellten Daten zu erkunden und durch Interaktion explorativ zugänglich zu machen. Hierzu wurden, wo dies

1 http://www.homermultitext.org/

möglich und sinnvoll war die dargestellten Einzelergebnisse verlinkt, sodass bei einem Klick eine Anfrage mit dem angeklickten Inhalt als Grundlage gestellt wird.

2.1. Adjazenzmatrix

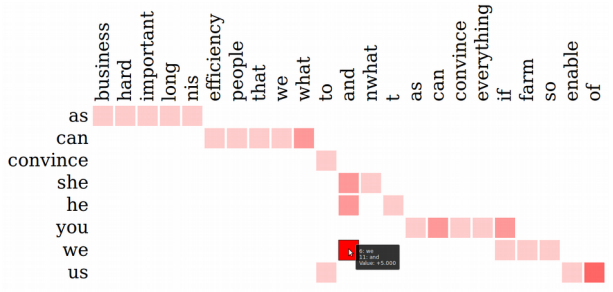

Abbildung 2: Adjazenzmatrix.

CTS-TM liefert zu einem in einem Dokument vorkommenden Token eine Liste von benachbarten Tokens. Dazu kann auch die Anzahl ausgegeben werden, wie oft diese beiden Tokens nebeneinander im Dokument vorgekommen sind. Um die Nachbarschaftsrelation verschiedener Tokens zu untersuchen, wurde eine Visualisierung in Form einer Adjazenzmatrix umgesetzt. Dabei werden für eine gegebene Liste von Tokens jeweils deren Nachbarn und die Häufigkeit abgefragt. In Abbildung 2 sieht man die vorgegebenen Tokens vertikal und die Nachbarn horizontal angeordnet. Je höher die jeweilige Häufigkeit, desto gesättigter ist der Farbton der entsprechenden Kachel. Der genaue Zahlwert erscheint beim Darüberfahren mit der Maus in einem Infofeld.
Durch Klicken auf ein Token kann nach bestimmten Zeilen oder Spalten sortiert werden.

2.2. WordGraph

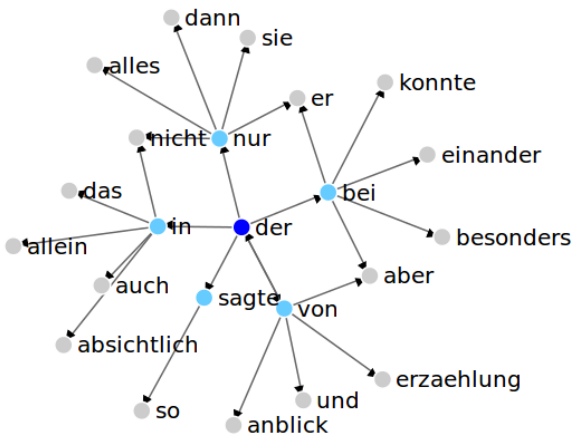

Abbildung 3: WordGraph.

Eine weitere Möglichkeit zweistellige Relationen, wie die Nachbarschaft von zwei Tokens darzustellen, ist die Anzeige als Graph. Abbildung 3 zeigt den *WordGraph*, eine Visualisierung, in der die Tokens als Knoten und die Nachbarschaften als Kanten zwischen diesen angezeigt werden. Als Vorgabe dient hier ein zentrales Token (dunkelblau), zu welchem die Nachbarn abgerufen werden (hellblau). Anschließend können rekursiv auch die Nachbarn der Nachbarn abgefragt werden. Als Parameter können hier angegeben werden, wie viele Rekursionsstufen durchlaufen werden sollen (*depth*) und wie viele Nachbarn in jeder Stufe maximal pro Vaterknoten hinzugefügt werden sollen (*filter*).
Diese Filterung ist notwendig, da der Graph stark vernetzt sein kann, was schnell zu Unübersichtlichkeit führt. Über den Parameter *allEdges* lässt sich einstellen, ob diese Filterung sich auf die hinzugefügten Kanten der Relation oder auf die Anzahl maximal hinzugefügter neuer Knoten beziehen soll. In ersterem Fall werden jeweils nur die ersten n am stärksten gewichteten Kanten der Relation hinzugefügt. Die zweite Option fügt ebenfalls diese ersten n Kanten hinzu, zeigt aber auch alle weiteren Kanten an, bei denen der Zielknoten vorher bereits im Graphen ist. Das heißt, die der dargestellte Graph bei der zweiten Filteroption ist eine Projektion der Gesamtrelation auf die Teilmenge der gefilterten Knoten.
Fährt der Benutzer mit der Maus über eine Knoten, so wird der zugehörige Text vergrößert dargestellt. Zur besseren Übersichtlichkeit kann der Anwender einzelne Knoten verschieben. Das dynamische, kräftebasierte Layout passt sich dann entsprechend an. Durch Klick auf einen Knoten wird eine analoge Visualisierung mit dem entsprechenden Knoten im Zentrum aufgerufen.

2.3. Charts

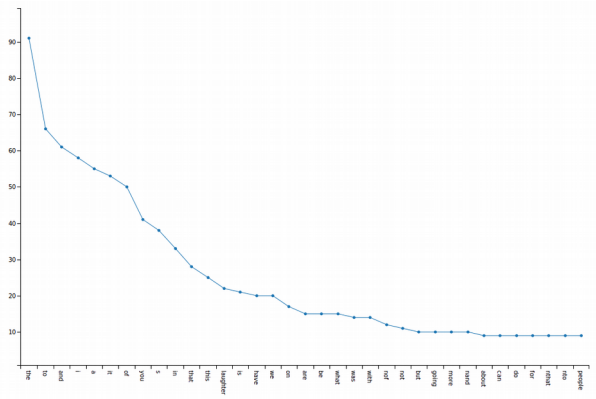

Abbildung 4: Charts (Liniendiagramm für Zipf-Verteilung).

Für gefundene Tokens kann CTS-TM zum Beispiel auch Häufigkeiten zählen und die Zipf-Verteilung (Tullo und Hurford, 2003) berechnen. Um dies graphisch darzustellen, können Diagramme verwendet werden. Hierzu wird die Bibliothek c3.js benutzt, die es ermöglicht verschiedene Diagrammtypen anzuzeigen. Hierzu gehören Linien-, Spline (geglättete Linie), Stufen-, Säulen- und Flächendiagramme, sowie Scatterplot-Darstellung.

Je nach Anwendungsfall kann der Diagrammtyp vom Nutzer ausgewählt werden. Auch die Spalten, in welchen Label und Werte stehen, sind vom Anwender auswählbar.
Auch bei dieser Visualisierung lassen sich die genauen Zahlwerte durch Darüberfahren mit der Maus anzeigen.

2.4. TopicCloud

Eine weitere Funktion von CTS-TM ist die Berechnung von Topic-Modellen mit Hilfe der von (McCallum, 2002) bereitgestellten Bibliothek Mallet. Dabei lassen sich nicht nur eine Übersicht der berechneten Topics mit ihren zugehörigen Tokens, sondern auch die zu einer Topic gehörenden Dokumente bzw. URNs und die zu einer URN gehörenden Topics mit ihren Tokens abrufen. Da zu den Tokens jeweils auch ein Gewicht berechnet wird, eignet sich hier eine Darstellung als *Schlagwortwolke* (*tag cloud*), in der jeweils die Tokens einer Topic nach ihrem Gewicht skaliert angezeigt werden.
Die *TopicCloud* Visualisierung basiert auf diesen Schlagwortwolken. Zunächst bietet sie eine Übersicht über die Topics mit ihren Tokens. Des Weiteren können per Klick auf eine Topic die zugehörigen URNs abgefragt und angezeigt werden. Diese sind dann wiederum auf eine Übersicht von Schlagwortwolken für die zu dieser URN gehörenden Topics verlinkt.

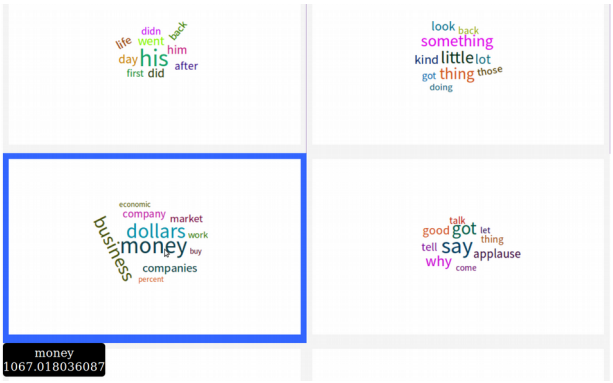

Abbildung 5: TopicCloud.

Wie in Abblidung 5 zu sehen, wird beim Darüberfahren mit der Maus die entsprechende *Schlagwortwolke* mit einem blauen Rahmen hervorgehoben. Außerdem wird in einem Infofeld das Token und das (relative) Gewicht angezeigt.

2.5. N-Gramm Tabelle

words			occurences	document
sagte	der	advokat	34	urn:cts:dta:kafka.prozess1925.de.norm:
sagte	der	kaufmann	31	urn:cts:dta:kafka.prozess1925.de.norm:
sagte	der	geistliche	29	urn:cts:dta:kafka.prozess1925.de.norm:
	der	direktor stellvertreter	24	urn:cts:dta:kafka.prozess1925.de.norm:
sagte	der	onkel	19	urn:cts:dta:kafka.prozess1925.de.norm:
mit	der	hand	18	urn:cts:dta:kafka.prozess1925.de.norm:
in	der	bank	18	urn:cts:dta:kafka.prozess1925.de.norm:
ja sagte	der		16	urn:cts:dta:kafka.prozess1925.de.norm:
nein sagte	der		14	urn:cts:dta:kafka.prozess1925.de.norm:

Abbildung 6: N-Gramm Tabelle.

CTS-TM kann N-Gramme, Sequenzen von jeweils N aufeinander folgenden Tokens, für verschiedene Längen N zählen und auswerten. In der in Abbildung 6 gezeigten Tabelle werden die Ergebnisse übersichtlich aufbereitet. Dabei wird ein Token bestimmt, das gelb hervorgehoben wird und an dem die N-Gramme ausgerichtet werden. Die Ausrichtung erfolgt dabei an der ersten Stelle, an der das Token im N-Gramm vorkommt. Daneben stehen die gezählte Anzahl und das Dokument.
Die einzelnen Tokens sind verlinkt auf eine Darstellung mit den gleichen sonstigen Parametern, basierend auf dem jeweiligen Token.

3. Flexibilität der Visualisierungen

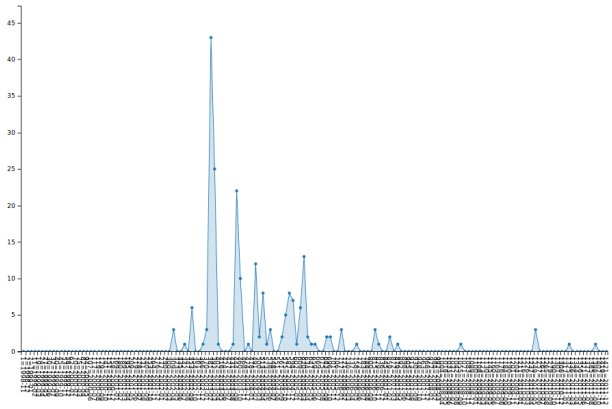

Abbildung 7: Charts (Flächendiagramm für Trendanalyse).

Wie bereits erwähnt, sind die Visualisierungen von den CTS-TM Funktionen getrennt und bekommen die zu verwendende Funktion meist als Funktionsparameter übergeben. Zusammen mit den weiteren Einstellungsmöglichkeiten lassen sich insbesondere *Charts* für viele unterschiedliche Funktionen verwenden. Abbildung 7 zeigt zum Beispiel eine Trendanalyse, d.h. eine Zählung der Häufigkeit für ein gegebenes Wort über einen Zeitraum.

4. Zusammenfassung und Ausblick

In Abschnitt 2 wurden verschiedene Visualisierungen vorgestellt, die die von CTS-TM bereitgestellten Ergebnisse graphisch aufbereiten. Dabei wurde darauf geachtet, Parameter inklusive der CTS-TM Funktionen möglichst dynamisch zu übergeben, sodass die Visualisierungen für verschiedene CTS-TM Funktionen verwendet werden können und auch nach möglichen Änderungen oder Erweiterungen von CTS-TM weiter funktionieren.
Zum Beispiel durch Interaktivität in *WordGraph* und der Adjazenzmatrix oder die Verlinkungen in *WordGraph*, *TopicCloud* und der N-Gramm Tabelle wird der Betrachter dazu befähigt, die Daten explorativ zu erkunden.
Alle vorgestellten Visualisierungen sind unabhängig voneinander lauffähig. In Zukunft wäre es möglich, die Darstellungen dahingehend zu überarbeiten, dass ein einheitliches Grunddesign erkennbar ist. Außerdem würde eine Übersichtsseite über die verschiedenen

Darstellungsmöglichkeiten die spätere Integration in CTS-TM erleichtern.

5. Acknowledgements

Diese Arbeit wurde im Rahmen des Seminars “Anwendungen Linguistische Informatik” der Universität Leipzig angefertigt.

6. Verwendete Programme und Bibliotheken

Alle vorgestellten Visualisierungen verwenden *d3.js*[2] (Bostock et al., 2011). Die Adjazenzmatix basiert auf *Correlation Explorer*[3] von Piotr Migdał (veröffentlicht unter CC-BY[4]). *WordGraph* basiert auf *Force-Directed Graph with Mouseover*[5], veröffentlicht unter GPL3[6]. Des Weiteren wurde für Charts die Bibliothek c3.js[7] und für TopicCloud wordcloud2.js[8] benutzt.

7. Literatur

Blackwell, C., Smith, N. (2014). The Canonical Text Services URN specification. http://folio.furman.edu /projects/citedocs/ctsurn/ Abgerufen am 01.09.2016

Bostock, M., Ogievetsky, V., Heer, J. (2011). D3: Data-Driven Documents. In *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis), 2011*.

McCallum, A. (2002) MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu. 2002.

Tiepmar, J. (2015). Release of the MySQL based implementation of the CTS protocol. In *Proceedings of the 3rd Workshop on Challanges in the Mangagement of Large Corpora (CMLC-3)*.

Tiepmar, J. (2016). CTS Text Miner, Text Mining Framework based on the Canonical Text Service Protocol. *In Proceedings of the 4th Workshop on the Challenges in the Management of Large Corpora p.1-7, LREC 2016.*

Tullo, C., Hurford, J. (2003). Modelling Zipfian Distribution in Language. In *Kirby, S. Language* Evolution and Computation, Proceedings of the *workshop at ESSLLI.*

2 https://d3js.org/
3 https://github.com/CompassInc/correlation-explorer
4 https://creativecommons.org/licenses/by/2.0/de/
5 http://bl.ocks.org/mbostock/2706022
6 https://opensource.org/licenses/GPL-3.0
7 http://c3js.org/
8 https://timdream.org/wordcloud2.js/