Bioinformatik Textmining

Dominik Habermann (108019250645)

31. August 2022

Zusammenfassung

Das Praktikum im Bereich der Bioinformatik bestand aus dem Auswerten von großen Datenmengen, die in Form von Textdateien im JSON-Format vorlagen.
Während der Durchführung hat sich herausgestellt, dass die Möglichkeiten der Datenauswertung größer sind als ursprünglich geplant, sodass diese Ausarbeitung lediglich den aktuellen Stand widerspiegeln kann.
Der Inhalt des Praktikums besaß seinen Fokus hauptsächlich auf die Methoden der Datenauswertung und weniger auf den biologischen Hintergrund der Daten.

Die Umsetzung erfolgte mittels der Programmiersprache `C`. Die mathematischen Methoden basieren auf angepassten elementaren Mengenoperationen. Das Ziel der Auswertung war das Ermitteln des Vorhandenseins von kleinen Wortmengen in den Zusammenfassungen von unterschiedlichsten Ausarbeitungen. Hiermit soll die Relevanz von Ausarbeitungen bei gegebenen Wortmengen ermittelt werden.

1 Hintergrund

In diesem Kapitel wird beschrieben woher die Daten stammen und was das erhoffte Ziel der Auswertung ist.

1.1 Quelle der Daten

Die Daten stammen von der Doktorandin Anjani Dhrangadhariya, die aktuell in der Schweiz unter Hilfenahme dieser Daten ihre Doktorarbeit verfasst. Bei den Daten handelt es sich um reale Daten, die aus dem Klinikbetrieb stammen.

1.2 Ziel der Auswertung

Das Ziel ist die Sortierung bezüglich der Relevanz von Abschlussarbeiten mittels deren Zusammenfassungen bei gegebenen Wortgruppen. Diese Wortgruppen bestehen i.d.R. aus einem bis 10 Wörter, die entweder eine Thematik, Chemikalien oder Technologien beschreiben.
Durch diese Sortierung erhofft man sich einen schnellen Zugriff auf die möglichst am meisten relevanten Quellen. Da es sich bei den vorhandenen Wortgruppen um Mengen in der Größenordnung von vielen Hunderttausenden bis hin zu einigen Millionen handelt, ist eine Sortierung nur mit der Unterstützung von Computern möglich.

2 Umsetzung

Das folgende Kapitel nennt die verwendeten Technologien und eine Erklärung warum genau diese verwendet wurden. Dazu kommt der konzeptionelle Aufbau des Auswertungsverfahrens in Form eines Diagramms und eine Erläuterung der Hintergedanken, die dabei gemacht wurden.

2.1 Verwendete Technologien

- Programmiersprache: `C` (Standard: `C11`)
- Compiler: GCC (Version: 11.2.0)
- Betriebssystem: Linux Mint 21 Xfce-64 Bit
- Versionsverwaltungstool: git (Version: 2.34.1)
- IDE: Eclipse for `C`/`C++` (Version: 2022-06 (4.24.0))
- Weiteres: u.a. make, gitk, Perf, Valgrind, cJSON

Die Wahl der Programmiersprache wurde insbesondere durch die Größen der Eingabedateien beeinflusst. So war bereits zu Projektbeginn bekannt, dass einige der zu verarbeitenden Dateien viele hundert Megabyte groß sein werden. Um solch eine große Menge an Daten möglichst effizient auswerten zu können, bietet sich eine hardwarenahe Programmiersprache wie `C` an. Die weiteren Technologien sind gängige Werkzeuge für Projekte, die mittels `C` oder `C++` entwickelt werden.

2.2 Portierbarkeit

Während des gesamten Projektes wurde darauf geachtet, dass der Quellcode portierbar ist und ohne Änderungen auf andere Betriebssysteme und Prozessorarchitekturen übersetzt werden kann. Dies wurde nicht nur durch die Wahl der Programmiersprache erreicht, sondern auch durch die strikte Verwendung des `C11`-Standards sowie der Standardbibliothek. Betriebssystem-spezifische Codefragmente wurden mittels Präprozessoranweisungen zur Übersetzungszeit auswählbar gemacht.[1]
Die Portierbarkeit ist ein entschiedener Faktor, da die Zielgruppe nicht mit Linux sondern mit Windows und Mac-OS arbeitet.

2.3 Konzeptioneller Aufbau

Das einfachste Verfahren, um eine Menge an Informationen nach der Relevanz für bestimmte Eingaben zu sortieren, ist das Bilden der Schnittmenge von der Eingabe mit den zur Verfügung gestellten Informationen. Dabei wird sowohl die Eingabe als auch die bereitgestellten Informationen in Tokens zerlegt. Im vorliegenden Fall stellen die Wörter in der Eingabe die Tokens dar.
Der grundlegende Aufbau ist in der Grafik 1 dargestellt.

[1] Eine Übersetzung ist unter Linux Mint + 21 und unter Windows (8.1 + 10) getestet worden.

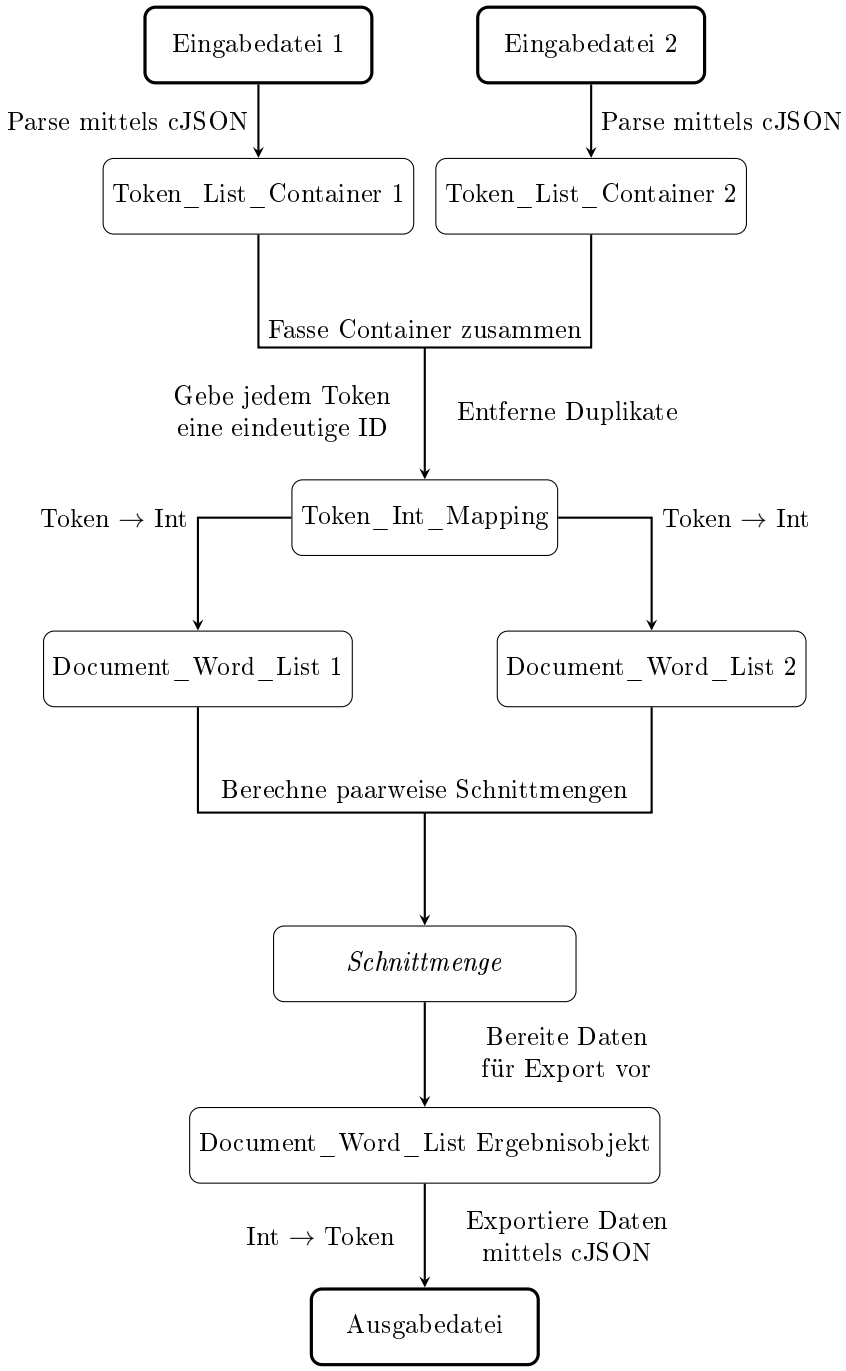

Abbildung 1: Aufbau des Auswertungsverfahrens

2.4 Token-Int Mapping

Das Token-Int Mapping ist eine bijektive Abbildung zwischen den Tokens aus den Eingabedateien und ganzzahligen Werten. Technisch gesehen ist solch eine

Abbildung nicht notwendig. Die Idee dahinter ist, dass Vergleiche von Zeichenketten relativ aufwendig sind, da jedes einzelne Zeichen miteinander verglichen werden muss. Wenn lediglich ein Wert verglichen werden muss, dann ist bei der vorgegeben Größe an Daten ein Geschwindigkeitsgewinn denkbar.[2]

2.5 Ergebnisdatei

Nach Rücksprache mit Anjani Dhrangadhariya wurde entschieden, dass die Ergebnisdatei – wie die Eingabedateien – auch dem JSON-Dateiformat entsprechen wird. Die Hauptmotivation dafür war die einfache Auswertbarkeit des Formats.[3] Die Bibliothek *cJSON* bietet auch Funktionen für den Export von Dateien im JSON-Format an, sodass keine weitere Bibliothek notwendig wurde.

3 Ergebnisse und Ausblick

Im letzten Kapitel werden die aktuellen Zwischenergebnisse genannt sowie ein denkbarer Projektablauf außerhalb vom Bioinformatik Modul.

3.1 Ergebnisse

Am Ende hat sich herausgestellt, dass die geplanten Auswertungen erfolgreich durchgeführt werden konnten. Zusätzlich haben sich bei der Umsetzung weitere Ideen ergeben, wie die Sortierung anhand der Relevanz noch verfeinert werden könnte.

3.2 Ausblick

Folgende Features können implementiert bzw. angepasst werden, um die Auswertung genauer zu machen und die Nutzerfreundlichkeit zu verbessern. Denn bisher lag der Fokus auf der Funktionalität und Korrektheit und nicht auf eine einfache Bedienbarkeit.

- Zusätzlich zur Schnittmenge kann die Position der Tokens in den Quelldateien ermittelt werden. Je geringer der Abstand zwischen den Tokens ist, desto wahrscheinlicher ist eine höhere Relevanz.
- Obwohl die Laufzeit trotz der großen Eingabedateien akzeptabel ist, besteht noch Potenzial für Verbesserungen.
- Bisher existiert lediglich ein rudimentäres CLI-Interface. Durch eine Verbesserung des CLI-Interfaces oder gar durch die Verwendung einer GUI, kann die Nutzerfreundlichkeit deutlich gesteigert werden.
- Das Programm erwartet aktuell Eingabedateien, die strikt dem erwarteten Format entsprechen. Bei falschen Datensätzen wird der Einfachheit halber die komplette Berechnung abgebrochen. Dieses Verhalten kann für die Zukunft angepasst werden, sodass das Programm flexibler auf fehlerhafte Datensätze reagiert. So können z.B. fehlerhafte Datensätze bei der Verarbeitung übersprungen werden.

[2] Eine Bestimmung des Geschwindigkeitsgewinns ist ohne weiteres nicht möglich, da nicht beide Verfahren (also mit und ohne eines Mappings) implementiert wurden.

[3] Dies war auch der Grund warum die Quellinformationen als JSON-Datei angelegt wurden.