Topic Modeling für Historiker (TopModHis)

Die parallel zur Arbeit entstandene Anwendung TopModHis vereint zwei Funktionen. Zum einen stellt die Anwendung alle notwendigen Operationen bereit, um die von LDA-C und HDP-faster erstellten Modelldateien bearbeiten und einsehen zu können. Zum anderen bietet es Funktionen zur Verarbeitung der Testergebnisse an. Das Programm funktioniert für LDA-C wie auch HDP-faster gleichermaßen. Es handelt sich um eine menügesteuerte Konsolenanwendung, die aber auch automatisierte Aufrufe mittels Parameterübergabe zulässt.

Allgemeines

Das Programm wurde in der Sprache C++ entwickelt und kompiliert auf allen bekannten Betriebssystemen. Bei der Erstellung wurde darauf geachtet, dass nur betriebssystemunabhängige Bibliotheken eingebunden wurden. Die auf GitHub[1] zu findende ausführbare Datei wurde unter Windows kompiliert. Die Anwendung besteht aus einer Header und einer Cpp Datei, wobei in der Cpp Datei nur der Programmstart (main), das Menü und die verschiedenen Abkürzungen zum automatisierten Programmstart zu finden sind. Um TopModHis für beide Typen von Modellen nutzbar zu machen, wurde mit einem Template gearbeitet, wodurch sämtliche Funktionen, die TopModHis direkt betreffen, in der Header Datei definiert und initialisiert werden müssen.

```
No filepath given
Please type in the filepath to the .beta or .topics file
Path: final.beta
Reading Model at final.beta
read it
The model includes 6 topics
The model includes 762588 terms
The minimum got set to -758.265.
The maximum got set to 0.
Please choose:
1 = delete topic
2 = add topic
3 = append another model
4 = save topic to edit in external file
5 = show a range of values of a topic
6 = edit single value in a topic
7 = Print Topics
8 = Print assignment document to topics
9 = Print Dates
10 = exit and save
0 = exit
Your choice:
```

Abbildung 1 Startansicht TopModHis

Anwendungsstart

TopModHis kann über drei verschiedene Arten aufgerufen werden. Die einfachste Form des Starts ist der Doppelklick auf die ausführbare Datei oder der Aufruf über die Konsole. TopModHis erbittet dann einen Pfad zur .beta oder .topics Datei und lädt diese dann zur weiteren Verarbeitung. Es ist zu beachten, dass die Anwendung auch unter Windows akzeptiert, dass einzelne Pfadelemente mit einem Slash (/) abgetrennt werden und nicht, wie bei Windows eigentlich üblich, mit einem Backslash (\). Die Pfadangabe kann relativ erfolgen, wobei nur der Doppelpunkt (../) und nicht der einfache Punkt (./) für die Pfadangabe akzeptiert werden.

Möchte man direkt den Pfad zur zu verarbeitenden Modelldatei angeben, so hängt man diesen bei dem Aufruf über die Konsole einfach an. Möchte man auf die Navigation in der Konsole verzichten, so kann man auch eine Verknüpfung zur Anwendung erstellen und in dieser die gewünschten Informationen unter „Eigenschaften“ -> „Ziel“ an den Pfad der Anwendung anhängen.

[1] Müller, Florian, GitRepository TopModHis, online verfügbar unter: https://github.com/ImdWf/TopModHis. Zuletzt geprüft am: 14.08.2019.

Geht man beispielsweise davon aus, dass die „final.beta“ Datei im Ordner „Ergebnisse“ liegt und der Aufruf einen Ordner darüber erfolgt, so wäre ein direkter Aufruf über

tmh.exe Ergebnisse/final.beta

möglich. Unter Linux und MacOS ist ein führender Punkt notwendig. Der Aufruf lautet hier

./tmh.exe Ergebnisse/final.beta

Möchte man keine Änderungen an einem Modell vornehmen, sondern nur die Begriffe pro Thema oder die Hierarchie ermitteln, so können diese Operationen ohne Menü direkt aufgerufen werden, indem man ähnlich wie zuvor die notwendigen Informationen dem Programmaufruf anhängt. Alle modellrelevanten Informationen befinden sich bei den folgenden beispielhaften Aufrufen wieder im Unterordner „Ergebnisse“ und alle korpusrelevanten Informationen auf der gleichen Ebene wie der Funktionsaufruf. *vocab.txt* steht dabei für das Vokabular, in dem pro Zeile ein Begriff aufgeführt wird. Die Zeilennummer entspricht hierbei später der ID im Modell. *doc_id.txt* enthält eine geordnete Liste der Dokumentennamen, wobei jede Zeile einem Dokumentennamen entspricht. In dieser Struktur lauten die direkten Aufrufe:

- Für das direkte Ausgeben der Topbegriffe pro Thema:

  HDP: tmh.exe Ergebnisse/final-test.topics 25 vocab.txt

  LDA: tmh.exe Ergebnisse/final.beta 25 vocab.txt

Die „25“ im Aufruf zur Ausgabe der Topbegriffe steht für die Menge an Begriffen, die gedruckt werden sollen.

- Für das Ermitteln der Hierarchie und der Themen geordnet nach Dokumenten:

  HDP: TopModHis.exe Ergebnisse/final-test.topics Ergebnisse/final-test.doc.states

  LDA: TopModHis.exe Ergebnisse/final.beta Ergebnisse/final-gamma.dat

- Für das Ermitteln der Zeit:

  HDP: tmh.exe Ergebnisse/final.topics Ergebnisse/final.doc.states Doc_id.txt

  LDA: tmh.exe Ergebnisse/final.beta Ergebnisse/final-gamma.dat Doc_id.txt

- Für das Erstellen eines Modells aller vereinten Hierarchieelemente, die ein bestimmtes Thema enthalten:

  HDP: tmh.exe Ergebnisse/final.topics Ergebnisse/final.doc.states Doc_id.txt 6

Es gilt zu beachten, dass das Ermitteln einer Hierarchie nur für HDP Modelle sinnvoll ist (.topics Datei), während alle anderen Funktionen auch für LDA Modelle zur Verfügung stehen. Die Zahl 6 bei der Erstellung der vereinten Hierarchieelemente steht für das Thema, das in allen zu berücksichtigenden Hierarchieelementen enthalten sein muss. TopModHis extrahiert diese Themen, addiert alle Themen für das entsprechende Hierarchieelement auf und gibt ein neues Modell aus.

Für die Weiterentwicklung der Anwendung wurde ein Debug-Modus integriert. Dieser ermöglicht es, langwierige Ladezeiten zu vermeiden und direkt auf bestimmte Funktionen zuzugreifen. Die verschiedenen Testszenarien müssen in der Funktion DebugMode() am Ende der Header Datei konfiguriert werden. Die einzelnen Bausteine, die bereits für die Erstellung eines Szenarios bereitstehen, beginnen mit dem Präfix *Debug.* Eine Auflistung über die vorhandenen Bausteine findet man in der

Klassendefinition der Klasse Modeleditor unter dem Punkt Debug. Der Debug Modus kann auf zwei Arten gestartet werden. Wenn man dem Programm beim Start statt des Pfades für die Modelldatei „42“ oder „42,42“ übergibt, startet TopModHis den Debug Modus entweder im HDP Format (42) oder im LDA Format (42,42). Zudem kann der Debug Modus im Hauptmenü über die nicht aufgeführte Option 42 gestartet werden.

```
Reading Model at 42
You are in the Debug Mode of TopModHis

Please select the routine you want to check:
1: Doc_Names_to_Date
2: Print_Topic_in_dates
3: PrintHDP
4: Print Topics
0: Exit

Your choice: 1
We need a document with filenames. Please tell the path to a testfile:
Filepath: ../Daten/Dokumente_sortiert_all_Unix.txt
Using Date Format 7.
```

Abbildung 2 Debug Modus

Abbildung 2 zeigt einen Aufruf des Debug Modus im HDP Format (ersichtlich an der Zeile: „Reading Model at 42“). Interessant ist hierbei die Zeile „Using Date Format 7“, die auf eine weitere Besonderheit von TopModHis hinweist. Um dem Anwender bei der Zuweisung von Datumszahlen zu seinen Dokumenten größtmögliche Freiheit zu lassen, wurde eine breite Zahl an Datumsformaten im Programm berücksichtigt. Im konkreten Fall weist TopModHis darauf hin, dass es Jahreszahlen und Kalenderwochen als Datumsformat gefunden hat. Die Ermittlung und Zuordnung der Themen zu Datumsangaben erfolgt hierbei über die Dokumentenbezeichnungen selber. Es wurde darauf geachtet, dass die Vorgaben für die Formatierung sehr gering sind. Das Programm unterscheidet zwischen der amerikanischen und der europäischen Datumsformatierung, akzeptiert Jahrestage und Kalenderwochen und schreibt die Position des Datums im Dokumentennamen im Grunde nicht vor. Dennoch sind, um die unterschiedlichsten Formate unterscheiden zu können, einige Regeln bei der Benennung der Dokumente zu beachten. Um diese Regeln zu verstehen, lohnt es sich, den Mechanismus hinter der Datumerkennung zu erläutern.

TopModHis ermittelt das Datum aus einer Reihe von Dokumentennamen, indem nach den ersten maximal drei aufeinanderfolgenden, durch **ein** beliebiges nichtnumerisches Trennzeichen getrennten Zahlen in jedem Namen gesucht wird. Dies erlaubt Dokumentennamen wie *Name01-01-2000* oder *35#2000#Name#Laufziffer*, aber nicht *Laufziffer-Name-01-01-2000*, da im letzten Fall mit der Laufziffer bereits eine Zahl im Dokumentennamen vor dem Datum steht, die TopModHis fälschlicherweise als Datum interpretiert.

Bei der Analyse der verschiedenen gefunden Datumsangaben wird nach dem Maximalwert von jedem der drei Werte gesucht und diese Maximalwerte danach nach ihrer Wertigkeit geordnet. Wird nur eine Zahl gefunden, so muss es sich für TopModHis um eine Jahreszahl handeln. Bei zwei gefundenen Zahlen wird die größte der beiden gefundenen Zahlen als Jahreszahl angenommen. Für die kleinere Zahl versucht TopModHis zu unterscheiden, ob es sich um Kalenderwochen, Monate oder Tage im Jahr handeln kann. Die einfache Logik ist hierbei, dass ein Jahr maximal 12 Monate, 53 Wochen besitzt oder 366 Tage besitzt. Wenn der Maximalwert der zweiten gefundenen Zahl also kleiner gleich 12 ist, nimmt TopModHis an, dass es sich um Monate handeln muss. Für Zahlen, die größer als zwölf aber kleiner als 54 sind, geht TopModHis von Kalenderwochen aus. Für Zahlen im Bereich von 54 bis 366 wiederum geht TopModHis von Tagen im Jahr aus. Sollte die zweite Zahl keinen der Kriterien entsprechen, so wird der Wert nicht weiter berücksichtigt und stattdessen die

Jahreszahl als einziger Bestandteil der Datumsangabe angenommen. Wie der Schilderung entnommen werden kann, ist vom Nutzer darauf zu achten, dass TopModHis das gewünschte Datumsformat auch erkennen kann. Besonders bei den zweiteiligen Datumsangaben ist es empfehlenswert, Werte zu haben, die klar im Bereich des jeweiligen Formates liegen.

Für die Fälle, dass TopModHis das Format Monat und Jahr bzw. nur Jahr erkennt, werden die nicht erkannten Bestandteile mit einer 1 aufgefüllt. Wird also nur das Jahr 1969 erkannt, hinterlegt TopModHis den 01.01.1969 als vollständiges Datum. Wird der Dezember 1969 als Datum erkannt, wird der 01.12.1969 im System als Datum abgelegt.

Ergebnisse

TopModHis erstellt unterschiedliche Dokumente in Abhängigkeit von den gewählten Optionen. Um die Funktion der unterschiedlichen Dokumente und ihren Inhalt besser verstehen zu können, folgt an dieser Stelle eine kurze Übersicht über die einzelnen Dokumente. Dabei wird sowohl auf den Inhalt als auch auf die Möglichkeiten eingegangen, wie die Datei erzeugt werden können.

Modell Datei [.beta/.topics]

Bei den Modelldateien bestimmt der Nutzer selber, wie er die Datei benennen möchte, jedoch wird automatisch in Abhängigkeit des Modus von TopModHis die Endung .topics im HDP Modus oder .beta im LDA Modus angehängt. Abgelegt werden die neuen Modelldateien immer in dem Ordner, in dem auch das Modell liegt, mit dem die Anwendung initialisiert wurde. Der Aufbau folgt dem jeweiligen Vorgaben der beiden Topic Modeling Anwendungen. Um eine solche Datei erstellen zu können, muss im Hauptmenü die Option 10 ausgewählt werden und das Anlegen einer neuen Modelldatei gewünscht werden (Überschreiben? -> Nein).

Datei für einzelne Topics

Bei den Dateien für die exportierten Themen kann der Nutzer Name und Dateiendung selber bestimmen. Der einzige Unterschied zwischen der Datei für einzelne Topics und einer Modell Datei besteht darin, dass die Zeilen transponiert wurden. Da die meisten Texteditoren Schwierigkeiten haben, unendlich lange Zeilen darzustellen, mit unendlich vielen Zeilen hingegen gut zurechtkommen, hat sich diese Umwandlung für eine bessere Bearbeitung angeboten. Die Erzeugung solcher Dateien geschieht über die Option 4 des Hauptmenüs.

Final_Cluster.txt

Die Final_Cluster.txt Datei beinhaltet im Grunde die Informationen, die die meisten Topic Modeling Anwendungen anzeigen. Nach den jeweiligen Themen sortiert stehen untereinander die wichtigsten Begriffe jedes Themas. Die Anzahl der Begriffe pro Thema wurden dabei vorher vom Nutzer festgelegt. Es wurde bei der Gestaltung dieser Datei darauf geachtet, dass der Import in anderen Anwendungen wie z.B. R oder Excel besonders einfach ist. Dazu wurde eine feste Spaltengröße von 50 Einheiten gewählt, die Themen nebeneinander angeordnet und als Trennzeichen das Leerzeichen gewählt. So können die Dateien wie CSV Dateien importiert werden. Um final_cluster.txt erstellen zu können, kann man entweder den weiter oben beschriebenen parametergestützten Start nutzen oder im Hauptmenü die Option 7 auswählen.

TopicsinDocuments.txt

Jede Zeile dieser Datei entspricht einem Dokument und in jeder Zeile reihen sich der Auftretens-häufigkeiten nach absteigend sortiert die Themen, die dem Dokument zugeordnet werden konnten. Die Ausgabe der Datei erfolgt parallel zur Ausgabe der TopicLevelIndex.txt und hilft, die Dokumente zu finden, die einem zu untersuchenden Thema zugeordnet wurden, da die Zeilennummern der Zeilen, in denen das gewünschte Topic enthalten ist, den Ids der Dokumente entspricht, die das Topic enthalten.

TopicLevelIndex.txt

Aufbauend auf den Informationen, die der Nutzer in TopicinDocuments.txt einsehen kann, entwickelt TopModHis eine Hierarchie der Themen. Die Hierarchie bildet sich dabei wie folgt:

Die erste Ebene der Hierarchie wird aus dem jeweils ersten Topic der Topics gebildet, da diese das jeweils größte Topic darstellen. Dies folgt der Logik aus der theoretischen Vorbetrachtung, dass es ein Restaurant / Topic geben muss, dass von nahezu allen Dokumenten angesteuert wird. In den weiter unten betrachteten Szenarien gab es zwar in der Regel ein Topic, das diesem Restaurant nahekam, aber es gab auch verschiedene andere Topics, die als Einstiegsrestaurant dienten. Im nächsten Schritt müssen alle Themen ermittelt werden, die in den Dokumenten das nächst kleinere Restaurant darstellen, d.h. dem Startrestaurant folgen. Dieser Vorgang wird im Anschluss solange wiederholt, bis alle Topics in allen Dokumenten erfasst wurden. Ein Knoten auf einer Ebene stellt nun die Kombination der verschiedenen Themen dar. Sei Thema 0 also der Startpunkt, dann ist Thema 0 auch Element der ersten Ebene. Seien nun die Themen 10 und 11 Themen, die auf das Thema 0 folgen, dann sind mögliche Elemente der zweiten Ebene „0 10“ und „0 11“. Folgt auf Thema 10 nun in der nächsten Iteration Thema 20 und auf Thema 11 Thema 22, dann sind mögliche Elemente der dritten Ebene „0 10 20“ und „0 11 22“.

Betrachten wir für ein konkretes Beispiel:

14 35 55 31 34 5
14 55 50 31 34 17 0 54 53 91 58
14 35 55 50 31 34 17 5 16 22 19 11
14 35 55 50 31 34 17 5 0 7 16 22 46 130 80
14 35 55 50 31 34 0 13 2 19
14 50 31 34 5 0 13

Element der ersten Ebene ist eindeutig das Thema 14. Hierbei handelt es sich meist um Begriffe, die von fast allen Dokumenten geteilt werden. Ohne Vorverarbeitung handelt es sich bei diesen Begriffen meist um Funktionswörter, die bei den meisten Analysen über Stopwort Listen entfernt werden.[2] Auf dieses Thema folgen die Themen 35,50 und 55. Dies bedeutet, dass sich die Elemente der zweiten Ebene aus den Kombinationen „14 35“, „14 50“ und „14 55“ ergeben. Die Elemente der dritten Ebene bilden sich nun aus allen eindeutigen Kombinationen, die auf den Elementen der zweiten Ebene aufbauen. Im konkreten Fall bedeutet dies „14 35 55“, „14 55 50“, „14 50 31“. Es gilt zu betonen, dass diese Rangfolgen eindeutig sind. Folgt also in einem Hierarchieelement auf Thema 14 Thema 50 als nächstgrößeres Thema, so wird Thema 35 nicht zu einem späteren Zeitpunkt in der Kette als Element erscheinen! Eine Kombination 14 50 35 ist in dem betrachteten Beispiel nicht möglich.

Die eigentlichen Themen bilden sich bei HDP-faster nun aus der Verkettung der einzelnen Unterthemen. Es genügt also nicht wie bei LDA-C, die Topwörter der ermittelten einzelnen Themen zu betrachten, sondern erst die Verknüpfung der einzelnen Unterthemen miteinander lässt ein eindeutiges Thema entstehen. Würde man die Themen nur getrennt betrachten, so erscheinen zwar einige Themen bereits ohne die anderen Themen klar, die meisten Themen aber erscheinen als unlogisches Gemisch aus Topwörtern verschiedener Themen. Die Datei wird im Zuge der Hierarchieerstellung (Option 8) erstellt oder die Erstellung wird direkt angefordert (s. *Ermitteln einer Hierarchie*).

---

[2] Vgl. David M. Blei, Thomas L. Griffiths und Michael I. Jordan, The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies, in: Journal of the ACM 57, 2010, Nr: 2, S. 1–30, hier S. 21. online verfügbar unter: doi.org/10.1145/1667053.1667056.

[Zahl]_[Dateiname].topics

Auf Grundlage der Themenhierarchie, wie sie in der Datei TopicLevelIndex.txt eingesehen werden kann, ist es möglich, sich die vereinten Hierarchieelemente als eigenständiges Modell ausgeben zu lassen. Da die Ausgabe aller Elemente eine lange Bearbeitungsdauer bedeutet und das Ergebnis aufgrund der Dateigröße kaum zu nutzen ist, wurde die Ausgabe auf Elemente beschränkt, die ein gewünschtes Thema enthalten. Damit der Nutzer im Anschluss weiß, welches Thema bei welcher Modelldatei im Fokus steht, wird die Zahl dem Dateinamen vorangestellt. Die Ausgabe dieser Datei erfolgt entweder im Zuge der Hierarchieerstellung (Option 8), kann aber auch direkt aufgerufen werden (s. *Erstellen vereinter Hierarchieelemente*).

DateTopic.txt

In dieser Datei werden die Vorkommen jedes Topics zu einem Zeitpunkt abgelegt. Das jeweilige Datum steht hierbei am Anfang einer Zeile und entspricht dem europäischen Standard. Im Anschluss folgt die Topic ID und die Häufigkeit des Auftretens des jeweiligen Themas. Die Erstellung der Datei kann über die Option 9 im Hauptmenü angefordert werden, direkt gestartet werden (s. *Ermitteln der Zeit*) oder im Zuge der Hierarchiestellung mit angefordert werden.

DateDocument.txt

In dieser Datei wird abgelegt, welches Dokument welchem Datum zugeordnet wurde. Die Erstellung der Datei erfolgt parallel zur DateTopic.txt. Beide Dateien werden entweder über einen parameter-gesteuerten Start (s. *Ermitteln der Zeit*), über die Option 9 im Hauptmenü oder zusammen mit der Hierarchie erstellt.

Laden eines leeren Modells

In einigen Situationen bietet es sich an, ein neues, leeres Topic zu erstellen. Hierfür übergibt man TopModHis einen gültigen Dateinamen (.topics oder .beta am Ende), für den es im entsprechenden Ordner keine Datei gibt. TopModHis lädt dann ein leeres Topic. In dieses können nun exportierte Topics oder ganze Modelle eingefügt werden und am Ende kann das Modell bei Bedarf über die Option 10 gespeichert werden. Es ist lediglich darauf hinzuweisen, dass an verschiedenen Stellen nicht alle Funktionen zur Verfügung stehen, da dafür eine korrekte Initialisierung des Modellmaximums und –minimums notwendig ist. In diesen Fällen bietet es sich an, zunächst ein Modell mit einem einzigen Topic zu speichern, zu laden und erst dann alle gewünschten Operationen durchzuführen.

```
Reading Model at empty.topics
The model includes 0 topics
The minimum got set to 500.
The maximum got set to 0.
Please choose:
1 = delete topic
2 = add topic
3 = append another model
4 = save topic to edit in external file
5 = show a range of values of a topic
6 = edit single value in a topic
7 = Print Topics
8 = Print assignment document to topics
9 = Print Dates
10 = exit and save
0 = exit
Your choice:
```

Abbildung 3 Ein leeres Topic, man beachte die Werte für Maximum und Minimum