LDA Topic Modeling über ein graphisches Interface

Simmler, Severin
severin.simmler@stud-mail.uni-wuerzburg.de
Universität Würzburg, Deutschland

Vitt, Thorsten
thorsten.vitt@uni-wuerzburg.de
Universität Würzburg, Deutschland

Pielström, Steffen
pielstroem@biozentrum.uni-wuerzburg.de
Universität Würzburg, Deutschland

LDA (Latent Dirichlet Allocation) Topic Modeling ist ein computergestütztes Verfahren zur semantischen Analyse digitaler Textsammlungen. Hierbei werden mit Hilfe eines probabilistischen Verfahrens aus Texten eine Reihe sogenannter “Topics” generiert: Gruppen semantisch ähnlicher Begriffe, die über mehrere Texte gemeinsam auftreten und im Modell als Wahrscheinlichkeitsverteilungen über die Gesamtheit des analysierten Vokabulars repräsentiert werden. Das heißt, daß zum Beispiel in einem Topic zum Thema Seefahrt nautische Begriffe besonders hohe Wahrscheinlichkeiten haben (Blei 2012, Steyvers und Griffiths 2006).

In den letzten Jahren ist das Interesse an LDA als Verfahren für die Analyse literarischer Textcorpora auf Seiten der digitalen Geisteswissenschaften stark gestiegen. Im Kontrast zu diesem gesteigerten Interesse ist die Anwendung der Methode allerdings nicht wesentlich leichter geworden. Gängige Implementierungen des LDA-Algorithmus werden entweder über ein kommandozeilenbasiertes Java-Programm (MALLET von McCallum 2002) oder über Skripte in der Programmiersprache Python (Gensim von Rehurek und Sojka 2010) angesprochen. Die Aufbereitung der Daten vor dem Topic Modeling, das sog. “Preprocessing” und die Analyse der Ergebnisse hinterher geschieht dann zumindest in Teilen häufig unter Verwendung weiterer Programme bzw. Arbeitsumgebungen. Alles in allem erfordert die Durchführung einer LDA-basierten Inhaltsanalyse damit zur Zeit relativ umfangreiche technische Kenntnisse.

Um den Zugang zu dieser Methode zu erleichtern entwickeln wir im Rahmen von DARIAH-DE (https://de.dariah.eu/) zur Zeit eine ausführlich dokumentierte Python-Programmbibliothek, die es ermöglichen soll, den gesamten Arbeitsprozess einer LDA-basierten Analyse in einer einzigen Umgebung durchzuführen (https://github.com/DARIAH-DE/Topics). Neben der Schaffung integrierter, flexibler Arbeitsabläufe, die vollständig in einer Programmiersprache und Umgebung stattfinden können, wollen wir auch Forscherinnen und Forschern ohne vorherige Programmierkenntnisse eine Möglichkeit zu bieten, Topic Modeling als Verfahren kennen zu lernen und an eigenen Daten auszuprobieren.

Um einen leichtgewichtigen Einstieg in diese Thematik zu bieten haben wir auf Basis unserer Programmbibliothek, der Python-nativen LDA-Implementierung von Allan Riddell (https://pypi.python.org/pypi/lda) und dem Python-Microframework “Flask” (http://flask.pocoo.org/) einen sogenannten GUI-Demonstrator entwickelt (Abb. 1). Dabei handelt es sich um eine browserbasierte graphische Benutzeroberfläche für die DARIAH-Topics Bibliothek, mit der sich ein basaler Topic-Modeling Analysevorgang lokal, mit eigenen Daten, aber eben ohne jegliche Programmierkenntnisse durchführen lässt.

Der GUI-Demonstrator übernimmt und erklärt hierbei exemplarisch alle Arbeitsschritte einer einfachen Analyse. Zunächst werden Textdateien über ein Auswahlmenü eingelesen und tokenisiert. Nutzerinnen und Nutzer können zur Reduktion des Vokabulars auf die Funktionswörter vorgeben, wie viele der häufigsten Wörter aus den Texten entfernt werden sollen, oder alternativ über ein weiteres Auswahlmenü eine externe Stopwortliste einbinden. Die Anzahl der zu berechnenden Topics und die Zahl der Iterationen, über die die Berechnung durchgeführt werden soll, ein Faktor, der die Qualität der Ergebnisse entscheidend beeinflusst, können ebenfalls über das Interface gesteuert werden. In der derzeitigen Form generiert das Programm als Output eine Tabelle mit den zehn am stärksten gewichteten Wörtern in jedem Topic, sowie ein Heatmap als Übersicht über die Verteilung der Topics über die Texte.

Im Fokus der gegenwärtigen Weiterentwicklung steht die Gestaltung interaktiver Outputs mit Hilfe von Bokeh (https://bokeh.pydata.org/), die einen flexibleren Zugriff auf eine größere Zahl von Aspekten der Modellierungsergebnisse ermöglichen sollen.

Das Ziel dieser Entwicklung bleibt aber in erster Linie ein didaktisches: Der GUI-Demonstrator führt die grundsätzlichen Möglichkeiten der Methode vor und informiert gleichzeitig über die Abläufe im Hintergrund, so dass der Schritt hin zur Verwendung der gleichen Funktionalitäten in einem vorbereiteten Notebook mit interaktiven Codeblöcken, das schnell an die spezifischen Bedürfnisse eine bestimmten Forschungsfrage angepasst werden kann, nur noch klein ist.

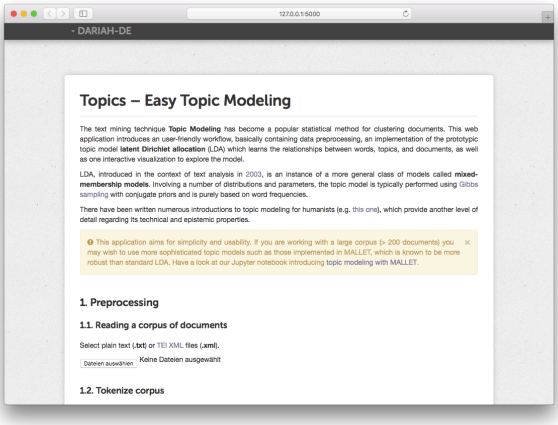

Abbildung 1.: Screenshot

Bibliographie

Blei, David M. (2012): „Probabilistic Topic Models“, in *Communication of the ACM* 55, Nr. 4 (2012): 77–84. doi:10.1145/2133806.2133826.

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

Rehurek, Radim/ Sojka, Petr (2010): "Software framework for topic modelling with large corpora." *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*.

Steyvers, Mark/ Griffiths, Tom (2006): „Probabilistic Topic Models“, in *Latent Semantic Analysis: A Road to Meaning*, herausgegeben von T. Landauer, D. McNamara, S. Dennis, und W. Kintsch. Laurence Erlbaum.