Exposé zur Bachelor Arbeit

Implementierung und Evaluierung einer Recommendation-Engine

Humboldt-Universität zu Berlin

by

Lucas Rebscher

(562827)

Berlin, May 08, 2017

1 Exposé

1.1 Kontext

Die Firma *XY* (anonymisiert) bietet ein Produkt an, das zum Beispiel Verlagen ermöglicht ihre Print-Magazine zu digitalisieren, interaktive Inhalte zu erstellen und über Apps anschließend automatisiert zu verbreiten.
Neben den Apps, die diese Verlage über das Produkt den Lesern anbieten können, betreiben die Verlage üblicherweise eine Webseite für jedes ihrer Magazine. Über diese können Besucher hauptsächlich kostenlosen Inhalt lesen, während in den Apps hingegen die kostenpflichtigen Ausgaben vertrieben werden. Die kostenlosen Inhalte sollen unter anderem den Leser davon überzeugen, ein Abonnement abzuschließen, um Zugriff auf den umfangreicheren und kostenpflichtigen Inhalt zu erhalten, der in den Apps angeboten wird.

1.2 Problemstellung

Bisher wird der Leser eines Magazins nur explizit auf die Möglichkeit hingewiesen ein Abonnement abzuschließen oder eine kostenpflichtige Ausgabe zu erwerben. Diese expliziten Hinweise nehmen jedoch keinen Bezug auf den Inhalt, den der Leser auf der Webseite gelesen hat. Am Ende eines Artikels auf der Webseite schlagen die Verlage üblicherweise dem Leser bereits weitere Web-Artikel vor. Die umfangreicheren und kostenpflichtigen Inhalte aus der App werden dabei jedoch nicht einbezogen. Es sollen aber mehr Leser, die einen kostenlosen Artikel auf der Webseite lesen, davon überzeugt werden, ein Abonnement abzuschließen oder eine kostenpflichtige Ausgabe zu kaufen und somit letztendlich auch die App herunterladen. Um dies zu erreichen, sollen am Ende eines Web-Artikels, ebenfalls Inhalte aus der App vorgeschlagen werden, die für den Leser von Interesse sein könnten.

1.3 Gegenstand der Arbeit

In dieser Arbeit soll eine Recommendation-Engine entwickelt werden, die für einen gegebenen Artikel Inhalte vorschlägt, die diesem Artikel ähnlich sind. Der Korpus wird aus den digitalisierten Artikeln der Magazine bestehen, sowie den Artikeln von den Webseiten der Verlage. Weiterhin wird der Korpus regelmäßig um neue Artikel aus den Magazinen und den Webseiten erweitert. Für die Berechnung der Vorschläge für einen gegebenen Artikel, sollen nur die Inhalte in Betracht gezogen werden, die Teil des gleichen Magazins sind, bzw. von der selben Webseite stammen.
Die Arbeit lässt sich dabei in 4 Abschnitte unterteilen:

(1) Zuerst wird die Recommendation-Engine implementiert. Die Engine soll auf dem weit verbreiteten Vector Space Model (Salton et al. (1975)) basieren und als Ähnlichkeitsmaß das TF-IDF-Maß verwenden. Eine Übersicht über verschiedene Variationen des TF-IDF-Maßes wird in (Salton and Buckley (1988)) verschafft. Die Open-Source Java-Library und Search-Engine Apache Lucene™ [1] bildet die Grundlage für die Recommendation-Engine, wobei in dieser Arbeit vor allem auf den Indexer und das integrierte Vector Space Model von Lucene zurückgegriffen werden wird. Diese Implementierung wird als Grundlage dienen, um darauf aufbauend weitere Konfigurationen auszuprobieren und zu evaluieren.

(2) Um die erhaltene Implementierung anschließend bewerten zu können, werden auf Basis eines reduzierten Korpus einige Artikel ausgewählt. Für diese Artikel bewerten Mitarbeiter der Firma, welche Inhalte sie in dem Korpus als relevant und welche sie nicht als relevant für den gegebenen Artikel sehen. Mit Hilfe dieser Auswertung wird die Recommendation-Engine schließlich bewertet.

(3) Nachdem durch die Evaluation der Implementierung eine Vergleichsbasis geschaffen wurde, werden nun verschiedene Anpassungen und Konfigurationen an der Implementierung vorgenommen und mit dieser verglichen.

(4) Diversifikation der vorgeschlagenen Inhalte ist zudem eine wichtige Anforderung an das System. So sollen dem Leser Inhalte vorgeschlagen werden, die sich untereinander nur zu einem gewissen Maße ähneln, um Redundanz zu minimieren und die Interessen-Breite der

[1] https://lucene.apache.org/core/

Leser abzudecken. In dieser Arbeit wird deshalb ein Optimierungsproblem formuliert und ein Diversifikations-Algorithmus implementiert. Wie in (Drosou and Pitoura (2010)) beschrieben wird, kann man Diversifikations-Algorithmen in zwei Kategorien unterteilen. Algorithmen, die mit expliziten Informationen arbeiten, wie eine vorliegende Themen-Kategorisierung der Artikel, und Algorithmen, die mit impliziten Information arbeiten, wie den berechneten Ähnlichkeits-Werten. Aufgrund fehlender expliziter Informationen über Inhalte in dem vorliegenden Korpus, wird ein impliziter Algorithmus gewählt. Dieser wird für ein Ranking von Recommendations, welches von der Basis-Implementierung aus (1) ermittelt wurde, die Ähnlichkeits-Werte der Recommendations verwenden, um ein neues, diversifiziertes Ranking zu berechnen. Im besonderen wird der *Mean-Variance-Analysis*-Algorithmus (kurz MVA) implementiert, der auf der Portfolio-Theory basiert und in (Wang and Zhu (2009)) vorgestellt wurde. Die Implementierung wird abschließend bewertet. Einen Überblick über Diversifikation, mit Verweise auf relevanten Veröffentlichungen, kann sich ebenfalls in (Drosou and Pitoura (2010)) verschafft werden.

References

Drosou, M. and E. Pitoura (2010): "Search result diversification," *ACM SIGMOD Record*, 39, 41–47.

Salton, G. and C. Buckley (1988): "Term-weighting approaches in automatic text retrieval," *Information processing & management*, 24, 513–523.

Salton, G., A. Wong, and C.-S. Yang (1975): "A vector space model for automatic indexing," *Communications of the ACM*, 18, 613–620.

Wang, J. and J. Zhu (2009): "Portfolio theory of information retrieval," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, ACM, 115–122.