

Semantic Similarity Clustering von Usability-Fragebögen für das automatische Analysieren von Online-Reviews

Thema:

Semantic Similarity Clustering von Usability-Fragebögen für das automatische Analysieren von Online-Reviews

Art:

[BA](#)

BetreuerIn:

[Manuel Burghardt](#)

BearbeiterIn:

Elias Haider

ErstgutachterIn:

[Christian Wolff](#)

ZweitgutachterIn:

[N.N.](#)

Status:

[abgeschlossen](#)

Stichworte:

[UX](#), [Text Mining](#), [Sentiment Analysis](#)

angelegt:

2017-11-23

Beginn:

2017-12-01

Antrittsvortrag:

2018-02-12

Textlizenz:

[Unbekannt](#)

Codelizenz:

[Unbekannt](#)

Hintergrund

User Experience (UX) ist ein wichtiger Faktor für die Akzeptanz und Anwendbarkeit von Produkten, sei es im Bereich Unterhaltungselektronik, Haushaltsgeräte oder auch Software. Dementsprechend hat sich Usability Engineering und UX-Testing zu einem wichtigen Teilbereich des Produktdesigns entwickelt. Üblicherweise wird die UX eines Produkts über Benutzerstudien ermittelt, bei denen Anwender bei der Benutzung des Produkts beobachtet und nach ihrem Nutzungserlebnis befragt werden. Diese Befragung ist häufig formalisiert durch den Einsatz standardisierter UX-Fragebögen, von denen der UEQ (User Experience Questionnaire) sicherlich einer der Bekanntesten ist.

Zielsetzung der Arbeit

Da die Durchführung solcher Nutzerstudien sehr aufwendig ist, soll im Rahmen der Bachelorarbeit erprobt werden, inwiefern mit Techniken des Natural Language Processing (NLP) und der Sentiment Analysis (SA) aus bestehenden, online verfügbaren Produkt-Reviews automatisiert Informationen zu deren UX abgeleitet werden können.

Konkrete Aufgaben

- Ausgehend von einer Wortliste mit 250 UX-Adjektiven aus mehreren UX-Fragebögen (zusammengestellt von Martin Schrepp) sollen zunächst semantische Cluster mithilfe von WordNet-Informationen gebildet werden, um so analog zum UEQ, größere UX-Dimensionen wie bspw. „Attraktivität“ oder „Effizienz“ abzuleiten.
- Als nächstes wird eine Pilotstudie mit einem Testkorpus durchgeführt, um ein Gefühl für die Abdeckung der 250 Wörter / Dimensionen zu bekommen; ggf. werden die Wörter der Wortliste noch mit NLP aufbereitet (Synonyme, Desambiguierung über Wortarten, etc.)
- Die überarbeitete Wortliste wird dann auf insgesamt 3 unterschiedliche Produkte angewandt
- Das Reviews-Korpus wird grundlegend mit NLP aufbereitet (Lemmatisierung, Stoppwörter entfernen, Wortarten-Tagging, etc.)
- Umsetzung einer UX-Mining-Komponenten, d.h. welche Wörter aus der Wortliste kommen wie oft in den Reviews vor, und wie kann man daraus einen Gesamtscore errechnen
- Evaluation 1: Die automatisch generierten UX-Ergebnisse werden mit bestehenden UX-Evaluationen zum Produkt abgeglichen und diskutiert
- Evaluation 2: Auf das Korpus wird außerdem ein Standard-Sentiment-Lexikon (z.B. NRC) angewandt – die Ergebnisse werden mit den Ergebnissen der UX-Liste verglichen und diskutiert (Sinn und Zweck: Bringt eine reduzierte UX-Adjektiv-Liste mehr als bestehende Sentiment-Lexika, in denen mehrere tausend Wörter enthalten sind)
- Evaluation 3: Wenn Amazon-Reviews verwendet werden, evtl. auch Abgleich mit den Sternebewertungen ...

Erwartete Vorkenntnisse

UX-Evaluation, Web-Programmierung, Python-Programmierung, empirische Grundlagen

Weiterführende Quellen

- Lechler & Burghardt (2017), UX-Mining-Prestudy:
http://dl.mensch-und-computer.de/bitstream/handle/123456789/5281/2017_MCI_228.pdf?sequence=1
- UEQ-Fragebogen: <http://www.ueq-online.org/>
- Semantische Ähnlichkeit berechnen: <http://swoogle.umbc.edu/SimService/api.html>
- Python NLTK für alle NLP-Aufgaben: <http://www.nltk.org/>
- Tutorial für Python BeautifulSoup, für das Crawling der Reviews:
<https://programminghistorian.org/lessons/intro-to-beautiful-soup>

- Wget Tutorial: <https://programminghistorian.org/lessons/automated-downloading-with-wget>;
<https://programminghistorian.org/lessons/applied-archival-downloading-with-wget>

From:

<https://wiki.mi.ur.de/> - **MI Wiki**

Permanent link:

https://wiki.mi.ur.de/arbeiten/ux-mining_fuer_online-produkt-reviews

Last update: **01.10.2019 12:26**

