



## KONZEPTION UND PROTOTYPISCHE ENTWICKLUNG EINES CHATBOTSYSTEMS FÜR DEN ABGLEICH MIT SUPPORT TICKETS MITTELS NLP

**MIRIAM LANG**  
(BACHELORSTUDIUM WIRTSCHAFTSINFORMATIK)

Betreuer: Prof. Dr. Marcel Tilly, Prof. Dr. Reiner Hüttl

### Einleitung

Eine aufstrebende digitale Technologie, die Unternehmen zunehmend im Kundenservice einsetzen, sind Chatbots. Die Nutzer eines Chatbots können mit diesem in natürlicher Sprache eine Konversation bezüglich einem bestimmten Anliegen führen. Dies wird mittels Methoden des Natural Language Processing (NLP) erreicht, da diese es dem Computer ermöglichen, die natürliche Sprache zu verarbeiten (Stucki et al., 2019). Im Rahmen der Bachelorarbeit wird untersucht, inwiefern sich NLP bzw. der Ansatz des Vector Space Models (VSMs) eignet, um den technischen Support für Nutzer einer Software zu unterstützen.

### Analyse

Zu Beginn der Arbeit ging es zunächst darum, ein grundlegendes Verständnis für das Softwaremodul sowie die damit verbundenen Support-Prozesse zu schaffen. Anschließend wurden die Anforderungen an das Chatbotssystem erhoben, um einige entscheidende Punkte für die Konzeption und Umsetzung zu klären. Dazu zählen das aktuelle Problem im Support des Softwaremoduls sowie das Hauptziel, das durch den Einsatz des Chatbots erreicht werden soll. Weitere Klärungspunkte waren, von wem der Chatbot genutzt werden soll und ob es Vorgaben bezüglich der zum Einsatz kommenden Technologien gibt. Abschließend wurden alle zur Verfügung stehenden Daten im Rahmen des Supports des Softwaremoduls auf deren Verwendungsmöglichkeit im Rahmen der Konzeption und Implementierung untersucht.

### Konzeption

Nachdem die Ausgangssituation untersucht und die Anforderungen geklärt waren, wurde ein Anwendungsfall ausgearbeitet, mit dem der Chatbot den Support unterstützen soll. Die Erkenntnisse aus der Analyse und eine Auswertung des Support Chats, bzgl. der Häufigkeit auftretender Support Fälle, halfen, sich auf einen der drei aktuellen Support-Prozesse festzulegen. Final sollte es möglich sein, dem Chatbot einen Fehler im Zusammenhang mit dem Modul zu schildern, woraufhin dieser die Fehlerbeschreibung mit allen Bug-Tickets im Ticketsystem Jira abgleicht. Der Chatbot liefert dem Nutzer

anschließend das semantisch ähnlichste Bug-Ticket. Da der Chatbot einen geschlossenen Themenbereich abdeckt, wurden als nächstes alle möglichen Konversationsverläufe mit diesem als Conversational Flow definiert. Für das Vergleichen der Fehlerbeschreibung des Nutzers mit den Jira Bug-Tickets bzgl. der semantischen Ähnlichkeit wurde der Ansatz des VSMs verfolgt. Für das Erstellen eines VSM sollte ein BERT-Modell oder eines seiner Varianten verwendet werden. BERT-Modelle sind mit großen Datenmengen trainierte Modelle, die den semantischen Inhalt eines Wortes oder Satzes in Form eines Vektors im mehrdimensionalen Raum darstellen können (Ravichandiran, 2021). Abbildung 2 zeigt das Maß an Ähnlichkeit für vier verschiedene Wörter anhand deren Vektoren im dreidimensionalen Raum. Durch die Berechnung der Kosinus Similarität (Winkel zwischen zwei Vektoren) kann die semantische Ähnlichkeit der Vektoren und somit der Wörter/Sätze berechnet werden (Navrozidis & Jansson, 2020). Da trotz einer Vorauswahl vier BERT-Modelle/-Varianten in Frage kamen, wurde anschließend evaluiert, welches Modell am besten mit den sprachlichen Herausforderungen im Kontext des Softwaremoduls umgehen kann. Im nächsten Schritt ging es um die Auswahl eines Tools zur Entwicklung des Chatbots. Zusammen mit den Anforderungen des Unternehmens wurden relevante Merkmale und Funktionen, die das Tool aufweisen sollte, ausgearbeitet. Rasa Open Source erfüllte als einziges Tool alle Anforderungen und wurde somit verwendet. Im letzten Schritt der Konzeption wurde die Architektur des Gesamtsystems, wie in Abbildung 2 zu sehen, aus den zu implementierenden und bestehenden Systemen modelliert.

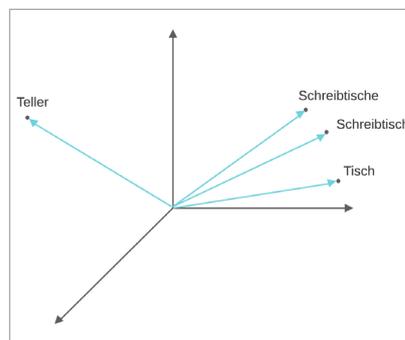


Abbildung 1: Ähnlichkeit der Bedeutung zwischen den Wörtern Schreibtische, Schreibtisch, Tisch und Teller vereinfacht dargestellt im dreidimensionalen Vektorraum (Eigene Darstellung nach Pilehvar und Camacho-Collados (2020))

### Entwicklung

Bevor die Implementierung des eigentlichen Chatbots begann, wurden zuerst die benötigten Ticketdaten aufbereitet und bereitgestellt. Hierfür wurde ein Python-Skript geschrieben, welches die Ticketinformationen über die Jira Schnittstelle abrufen, von dem Ticket Titel die Vektor-Darstellung berechnet und in einer Datenbank speichert. Die anschließende Entwicklung mit Rasa Open Source erfolgte hauptsächlich über YAML-Dateien. In diesen wurden zum einen beispielhafte Aussagen angegeben anhand derer der Chatbot lernen kann zu verstehen, welche Art

# ROSENHEIMER INFORMATIKPREIS WIF-BACHELOR

von Aussage der Nutzer tätig (Begrüßung, Fehlerbeschreibung, positive oder negative Rückmeldung). Des Weiteren wurden die meisten Antworten des Chatbots definiert sowie die möglichen Konversationsverläufe. Die dynamisch generierte Antwort mit den Ticketinformationen wurde innerhalb einer Python-Datei als Rasa Action umgesetzt. Hauptaufgabe der Action ist es, die Vektor-Darstellung der Fehlerbeschreibung zu berechnen, mit den in der Datenbank gespeicherten Vektoren zu vergleichen und somit das Bug-Ticket mit der größten Übereinstimmung zu ermitteln. Nachdem die Logik des Chatbots fertiggestellt war, ging es nun um die Durchführung von Tests, um Fehler und andere Mängel identifizieren und ausbessern zu können.

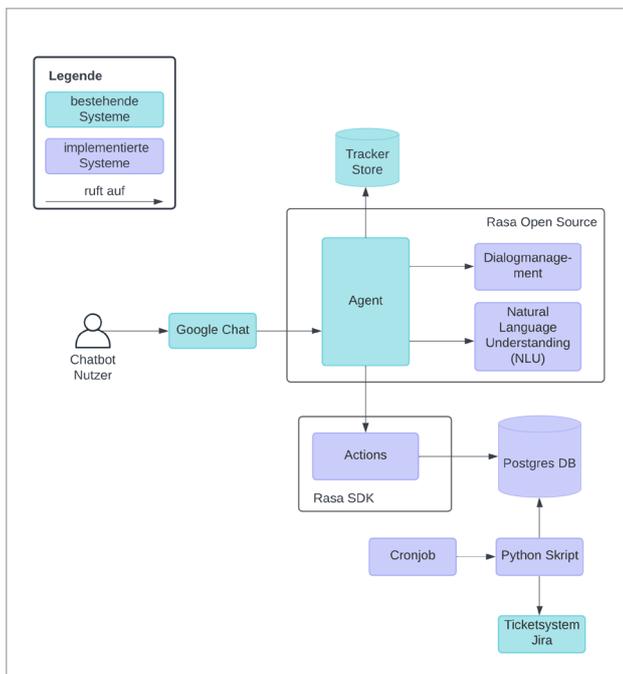


Abbildung 2: UML-Diagramm des Chatbot-Systems

Die automatisierten und manuellen Tests zeigten, dass es Optimierungsbedarf in Bezug auf die Dauer der dynamischen Antwort-Generierung gibt. Da die prototypische Entwicklung nun abgeschlossen war, wurde der Chatbot über einen Server der Hetzner Cloud in einer virtuellen Maschine gehostet. Danach wurde der Chatbot über ein Google Cloud Projekt in den Google Chat integriert. Dabei wurde festgestellt, dass sich die Dauer der Antwort-Generierung bei allen Antworten um zwei Sekunden erhöhte. Abschließend wurde das gesamte Konzept dahingehend bewertet, wie gut sich dieses dazu eignet, das passende Bug-Ticket zu ermitteln. Zusammenfassend lässt sich sagen, dass mit

diesem Konzept unter Einhaltung zweier Bedingungen zu 95 Prozent das korrekte semantisch ähnlichste Bug-Ticket ermittelt werden kann.

## Fazit und Ausblick

Das Ziel der Arbeit war es, herauszufinden, ob sich mithilfe von NLP ein Chatbot-System konzipieren und prototypisch umsetzen lässt, das die Softwareentwickler eines Moduls im täglichen internen Support für dessen Nutzer unterstützt. Dazu lässt sich abschließend festhalten, dass ein Prototyp entwickelt werden konnte, der Potenzial für die Unterstützung im Support aufweist. Vor seinem tatsächlichen Einsatz im Unternehmen sollte er allerdings weiter optimiert werden, sodass die Dauer der Antwort-Generierung reduziert und die einzuhaltenden Bedingungen zur Ermittlung des semantisch ähnlichsten Bug-Tickets aufgehoben werden können. Ein Ansatz zum Aufheben der Bedingungen wäre, auf das zur Ticket-Ermittlung im Einsatz befindliche pretrained BERT-Modell einen zusätzlichen Trainingsprozess anzuwenden, sodass dieses die Sprache im Rahmen des Moduls und seines Supports noch besser zu verstehen lernt. Konnten die Verbesserungen umgesetzt und das Chatbot-System dadurch zur Unterstützung im Support eingesetzt werden, könnte der Chatbot um weitere Anwendungsfälle erweitert werden.

## Abkürzungen:

- BERT** Bidirectional Encoder Representation from Transformer
- NLP** Natural Language Processing
- VSM** Vector Space Model
- YAML** Yet Another Markup Language

## Literaturverzeichnis:

- Navrozidis, J., & Jansson, H. (2020). Using Natural Language Processing to Identify Similar Patent Documents.
- Pilehvar, M. T., & Camacho-Collados, J. (2020). Embeddings in Natural Language Processing: Theory and Advances in Vector Representations of Meaning. Morgan & Claypool Publishers.
- Ravichandiran, S. (2021). Getting Started with Google BERT: Build and Train State-Of-the-art Natural Language Processing Models Using BERT. Birmingham: Packt Publishing.
- Stucki, T., D'Onofrio, S., & Portmann, E. (2019). Chatbots gestalten mit Praxisbeispielen der Schweizerischen Post: HMD Best Paper Award 2018. Wiesbaden: Springer Vieweg.