

Seminararbeit

Stefan Kert



Information Engineering and Management
University of Applied Sciences Upper Austria

7th July, 2017

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ziel und Methodik	2
2	Natural Language Processing	4
2.1	Überblick	4
2.2	Verarbeitung und Interpretation von natürlicher Sprache	4
2.2.1	Eingabe von Sprache und Vorverarbeitung	4
2.2.2	Vorverarbeitung und Information Extraction	5
2.2.3	Interpretation	6
2.2.4	Ausgabe und Ergebnis	6
2.3	Anwendungen von Natural Language Processing	6
2.3.1	Chatbots	6
2.3.2	Spam-Filter	7
2.3.3	Sprachassistenten	7
3	Anwendung von NLP in IT-Prozessen	9
3.1	Allgemeines	9
3.2	Das Beispielunternehmen	9
3.3	Supportprozess	10
3.3.1	Ablauf	10
3.3.2	Verbesserung 1 - Kategorisierung der Anfrage	12
3.3.3	Verbesserung 2 - Hinzufügen von Informationen aus der Wissensbasis und Verknüpfen mit Anfragen aus der Vergangenheit	14
3.3.4	Verbesserung 3 - Erkennen von duplizierten Tickets	18
4	Fazit	20
	Quellenverzeichnis	21
	Literatur	21

Kapitel 1

Einleitung

1.1 Motivation

Natural Language Processing ist die Möglichkeit der maschinellen Verarbeitung und Interpretation von natürlicher Sprache. Dabei beschränken sich die Anwendungen von *NLP* nicht ausschließlich auf geschriebene Sprache in Form von Wörtern und Texten, sondern auch auf Lautsprache (gesprochene Sprache). Bereits seit mehreren Jahren wird im privaten Bereich auf sogenannte Sprachassistenten wie Cortana (Microsoft), Siri (Apple) und Amazon Echo (Amazon) gesetzt. Diese ermöglichen es, verschiedene Aufgaben mittels Lautsprache zu erledigen. Dabei sind Anwendungen wie z.B. das Ein und Ausschalten von Licht, das Abspielen von Playlists oder auch das Starten eines Anrufs nur einige der vielfältigen Anwendungsgebiete der Spracherkennung. Im Zusammenhang mit geschriebenen Wörtern und Texten wird *NLP* vor allem bei der Interpretation von Texten eingesetzt. Ein Beispiel für die Anwendung von *NLP* ist die Klassifizierung von Texten. Dies findet vor allem bei wissenschaftlichen Arbeiten Anwendung [DHH17]. Ein weiteres Anwendungsgebiet im Zusammenhang mit Texten ist das Zusammenfassen oder die semantische Überprüfung der Inhalte. Vor allem in den letzten Jahren hat das Thema Natural Language Processing im Zusammenhang mit sozialen Netzwerken an Bedeutung gewonnen. Eine Anwendung, welche in letzter Zeit sehr stark im Fokus von Behörden steht, ist die Erkennung von sogenannten *Hasspostings* [Nob+16]. Durch die Masse an Postings die täglich in den sozialen Medien getätigt werden ist es nahezu unmöglich alle strafrechtlich relevanten Beiträge herauszufinden und diese gegebenenfalls zu löschen oder auch weitere rechtliche Schritte einzuleiten. Ein weiteres Anwendungsgebiet im Zusammenhang mit Postings in sozialen Medien ist das Analysieren dieser Postings zu Werbe- oder Marketingzwecken (User A postet, dass er gerne wieder auf Urlaub fahren würde, bekommt gezielt Werbung zu Urlaubsangeboten). Vor allem auf Grund des sehr werbelastigen Geschäftsmodells ist diese Analyse ein Thema von hoher Relevanz für die Betreiber von Werbeplattformen bzw. sozialen Medien wie *Facebook* oder *Twitter*.

Ein Gebiet, in dem die Möglichkeiten der Texterkennung sowie der Spracherkennung sehr stark zu einer Verbesserung beitragen könnten, sind IT-Prozesse mit einem starken Fokus auf zwischenmenschliche Kommunikation. In diesen kommu-

nikationslastigen IT-Prozessen werden viele Daten in Form von menschlicher Sprache erzeugt. Diese kann entweder in Form von Texten (E-Mail, Dokumentation, Defect-Beschreibung), oder in Form von Sprache (Telefonanruf, Mailboxnachricht) vorliegen. Ein klassisches Beispiel für einen IT-Prozess bei dem die Daten in Form von menschlicher Sprache vorliegen sind Support-Prozesse.

Ein konkretes Beispiel für die Vorselektierung bzw. Vorsortierung von Daten mit der Unterstützung von Natural Language Processing ist die Erkennung von Spam-Mails [RDP14]. Beim Erkennen von Spam-Mails gibt es triviale Vorgehensweisen wie z.B. Blacklists mit Hilfe derer an Hand des Absenders festgestellt wird ob es sich um eine Spam-Mail handelt oder nicht. Komplexere Verfahren wären zum Beispiel die Analyse des Textes auf spezielle Schlagwörter oder auch den semantischen Inhalt des Textes mit Hilfe von Natural Language Processing.

Allgemein lässt sich sagen, dass der richtige und gezielte Einsatz von Natural Language Processing entscheidend zum Unternehmenserfolg beitragen kann. Vor allem in Prozessen, in denen die direkte Kommunikation zwischen Menschen eine wichtige Rolle spielt, kann durch die Verarbeitung und Auswertung der entstehenden Daten (textuell oder Sprache) eine Verbesserung erzielt werden.

Einige IT-Prozesse werden bereits rudimentär durch Natural Language Processing unterstützt, andere wiederum könnten von der Anwendung sehr profitieren. Ziel dieser Arbeit ist es, an Hand ausgewählter IT Prozesse (z.B. Support-Prozess, Anfrage eines Neukunden) beispielhaft zu erläutern, wie diese Prozesse durch die Anwendung von Natural Language Processing verbessert werden könnten. Der Fokus dieser Arbeit liegt dabei darauf, dass die Personen, die Teil dieses Prozesses sind, entlastet bzw. unterstützt werden. Dies kann die Mitarbeiter des Unternehmens betreffen, welche die Serviceleistung in Form eines Prozesses anbietet, aber auch die Kunden dieses Unternehmens durch eine möglicherweise schnellere Abwicklung dieses Prozesses (schneller bearbeitete Support-Anfragen, Chat-Bot statt FAQ) unterstützen und schlussendlich die Zufriedenheit auf beiden Seiten zu erhöhen.

1.2 Ziel und Methodik

Der Inhalt dieser Arbeit besteht in erster Linie aus der Analyse ausgewählter IT-Prozesse und dem Erläutern der Möglichkeiten der Anwendung von Natural Language Processing zur Unterstützung dieser.

Für die Analyse werden die IT-Prozesse eines klassischen Softwareentwicklungsbetriebes verwendet. Diese Prozesse werden kurz analysiert und stellen danach den IST Zustand dar. Es wird ein kurzer Überblick über den aktuellen Prozess gegeben und anschließend erläutert, welche Daten in diesem Prozess zwischen den einzelnen Schritten erzeugt bzw. verarbeitet werden. Danach wird gezeigt, wie mit Hilfe von Natural Language Processing eine Vorverarbeitung, Optimierung bzw. Erweiterung dieser Daten erfolgen kann. Am Ende sollte ein neuer Prozess skizziert werden, der durch die Unterstützung von Natural Language Processing verbessert wurde.

Ein wichtiger Punkt, der bei der Analyse miteinbezogen wird, ist die Machbarkeit der einzelnen Verbesserungen, sowie der Limitierungen die sich mit der Anwendung ergeben könnten. Diese Limitierungen bzw. die Machbarkeit sollte in

eine abschließende Bemerkung mit einfließen und Aufschluss darüber geben, unter welchen Voraussetzung NLP eingesetzt werden könnte.

Kapitel 2

Natural Language Processing

2.1 Überblick

In diesem Kapitel wird ein grober Überblick über den Ablauf beim *NLP* gegeben und außerdem einige Beispiele gezeigt, die bereits im IT-Umfeld eingesetzt werden.

2.2 Verarbeitung und Interpretation von natürlicher Sprache

Grundsätzlich ist Basis jeder *NLP-Engine* die Verarbeitung und Interpretierung von geschriebener Sprache. Dabei ist zu unterscheiden, aus welcher Quelle der Input für die Verarbeitung kommt. Meist wird hier zwischen zwei Teilgebieten unterschieden:

- Maschinenlesbare Texte
- Spracheingabe

Diese beiden Teilgebiete unterscheiden sich darin, dass bei der Spracheingabe noch eine Vorverarbeitung stattfinden muss, sodass der gesprochene Text interpretiert werden kann. Sobald der gesprochene Text in Form eines maschinenlesbaren Textes vorliegt, kann mit der Weiterverarbeitung begonnen werden. Dies bedeutet im Umkehrschluss, dass beide Teilgebiete auf der gleichen Engine basieren können, sich jedoch in ihrer Komplexität und der Möglichkeit der Eingabe unterscheiden.

Im Folgenden wird kurz erläutert, wie der Ablauf einer Analyse im *NLP* vom Input bis zum Output abläuft.

2.2.1 Eingabe von Sprache und Vorverarbeitung

Die Eingabe von Sprache für das *NLP* kann auf mehrere verschiedene Arten erfolgen. Es gibt die Möglichkeit, über ein Mikrofon Lautsprache aufzunehmen, oder verschiedene Texte als Input zu verwenden. Dabei könnten zum Beispiel Texte automatisch von Internetseiten ausgelesen, oder auch Dokumente am Rechner durchsucht werden. Eine weitere Möglichkeit für textuellen Input lässt sich in Form eines Chats finden. Hier findet eine textuelle Eingabe statt, die später von einer Engine verarbeitet werden kann.

Prinzipiell unterscheiden sich diese beiden Eingabemöglichkeiten sehr stark in ihrer Komplexität. Bei der Eingabe über Lautsprache muss beim Einlesen nicht nur auf das Erkennen der Wörter, sondern auf eine etwaige Fehlerkorrektur durch Unterbrechungen oder Nebengeräusche geachtet werden. Durch diese Probleme gestaltet sich die Spracherkennung als sehr schwierig, wodurch die Möglichkeiten der Anwendung sehr stark eingeschränkt werden. Nicht zuletzt durch die Tatsache, dass in den letzten Jahren sehr viel auf dem Gebiet der Spracherkennung geforscht wurde, konnte im vergangenen Jahr ein Erfolg auf diesem Gebiet erreicht werden. Einem Forschungsteam der Firma Microsoft ist es gelungen, eine *Word error rate* von 5,9% zu erreichen, was ungefähr der Fehlerrate entspricht, die bei der normalen zwischenmenschlichen Kommunikation auftritt [Xio+16].

2.2.2 Vorverarbeitung und Information Extraction

Sobald Text in maschinenlesbarer Form vorliegt (z.B. ASCII-Code) kann auf den syntaktischen bzw. den semantischen Inhalt geprüft werden. Dazu gibt es verschiedene Möglichkeiten für die sogenannte *Information Extraction*. Ein Beispiel dafür ist die sogenannte *Named Entity Recognition (NER)*. *NER* ermöglicht das Verarbeiten der Texte und liefert Informationen über den Inhalt der Texte. Ergebnis der *NER* sind die sogenannten *Entities*. Es handelt sich dabei um Personen oder Objekte, welche in dem zu analysierenden Text enthalten sind. Es werden außerdem Attribute der *Entities*, wie zum Beispiel das Alter, extrahiert. Für eine genauere Analyse können auch Beziehungen zwischen *Entities* oder Ereignisse in welche *Entities* einbezogen sind ausgelesen werden. Diese Informationen können für die weitere Verarbeitung und Interpretation verwendet werden und je nach Genauigkeit der ausgelesenen Daten kann die Qualität der Aussage die über den Inhalt getroffen werden kann variieren. In der untenstehenden Grafik ist der Ablauf eines *NER* Zyklus dargestellt.

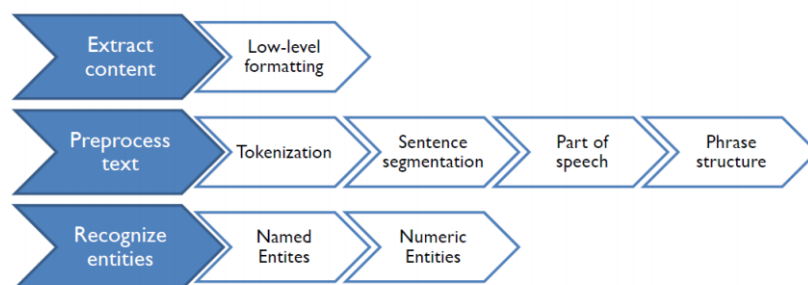


Abbildung 2.1: Named Entity Recognition Prozess

Der in Abbildung 2.1 dargestellte NER Prozess zeigt sehr gut, wie der normale Ablauf bei der Texterkennung erfolgt. Zuerst wird der Text verarbeitet, dazu werden unnötige Zeichen entfernt; Bilder, Werbung und andere Objekte mit keiner semantischen Bedeutung werden ebenfalls entfernt. Danach wird der Text vorverarbeitet. Es werden dazu einzelne Tokens gebildet, wobei jeder Token ein Wort repräsentiert. Diese Tokens werden wiederum zu Sätzen zusammengefasst, sodass

die semantische Bedeutung nicht verloren geht. Sobald Sätze gebildet wurden, erfolgt das sogenannte *Part-of-speech-tagging*. Beim *Part-of-speech-tagging* wird festgestellt, welche verschiedenen Wortarten die Tokens darstellen und die Tokens bekommen ein Label mit der jeweiligen Wortart. Dies dient vor allem der besseren semantischen Interpretation. Im letzten Schritt der Vorverarbeitung wird die grammatikalische Struktur des Satzes in Form eines Baumes abgebildet. Sobald alle Schritte abgeschlossen sind werden die einzelnen sogenannten Entities ausgelesen. Man unterscheidet hierbei zwischen *Named Entities*, es handelt sich dabei um Objekte wie z.B. Personen, oder um *Numeric Entities*, welche z.B. Jahreszahlen repräsentieren. Mit diesen Informationen kann schließlich eine weitere Interpretation auf den semantischen Inhalt dieses Textes erfolgen.

2.2.3 Interpretation

Der letzte Schritt bei der Verarbeitung von natürlicher Sprache ist die Interpretation des Inhaltes. Dabei werden die extrahierten Informationen mittels Algorithmus so vernetzt, dass Aussagen über den Inhalt getroffen werden können. Dieser Teilausschnitt des *NLP* ist sehr vielseitig und abhängig von der Komplexität der Anforderung. Es können hier von einfachen logischen Überprüfungen bis zu der Anwendung von künstlicher Intelligenz ein breites Spektrum an Algorithmen verwendet werden.

2.2.4 Ausgabe und Ergebnis

Am Ende sollte je nach Anforderung bzw. Anwendung ein Text ausgegeben, ein Ereignis getriggert, oder weitere Schritte eingeleitet werden. In den nächsten Abschnitten werden einige Möglichkeiten der Anwendung von *NLP* erläutert, die bereits in IT-Prozessen Anwendung finden oder eventuell verwendet werden könnten.

2.3 Anwendungen von Natural Language Processing

Nachdem im letzten Abschnitt grob skizziert wurde, wie die Verarbeitung und Interpretation von natürlicher Sprache stattfindet, wird im Folgenden auf verschiedene Beispiele für Anwendungen von *NLP* im IT-Bereich eingegangen.

2.3.1 Chatbots

Ein Bereich in dem die automatisierte Texterkennung verstärkt Anwendung findet, sind sogenannte Bots. Die Bots könnten dabei z.B. den First-Level Support ersetzen und grundlegende Fragen klären, wie die Art des Problems, etwaige Kontaktdaten oder auch die Dringlichkeit des Problems. Es findet dabei eine bidirektionale Kommunikation zwischen dem Kunden und einem Bot statt. Der Kunde schreibt eine Nachricht, der Bot versucht diese zu interpretieren und weitere Fragen zu erschließen. Moderne Bots bieten hier zusätzlich die Möglichkeit, dass der aktuelle Kontext persistent ist, was zur Folge hat, dass die Konversation durch die bereits gestellten Fragen und erhaltenen Antworten angepasst wird.

In Abbildung 2.2 ist beispielhaft dargestellt, wie mit Hilfe der Texterkennung analysiert wird, welche Beiträge der Anwender gerne sehen möchte. Es werden da-

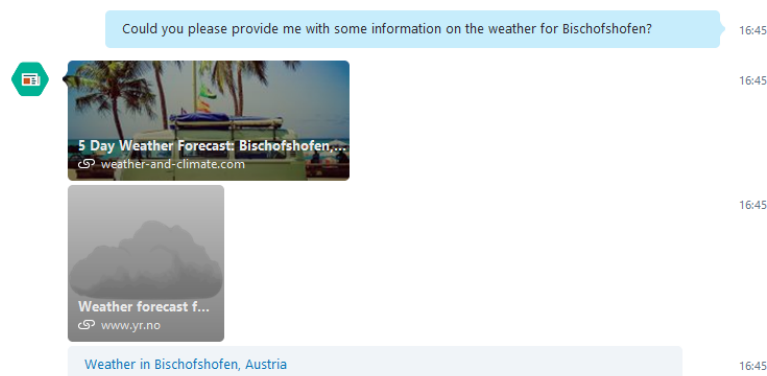


Abbildung 2.2: Bingnews Skype bot

nach die Artikel übermittelt, welche am besten mit den gesuchten Begriffen übereinstimmen.

2.3.2 Spam-Filter

Eine äußerst bekannte und sehr weit verbreitete Anwendung von *NLP* sind Spam-Filter. In der Vergangenheit wurde bei Spam-Filtern lediglich ein Whitelisting bzw. Blacklisting für verschiedene Absender oder Wörter betrieben. Sobald eine Email von einem Absender kommt, der sich auf einer Blacklist befindet, wird diese als Spam markiert. Selbiges gilt für Emails mit gewissem Inhalt, der darauf schließen lässt, dass es sich bei der Email höchstwahrscheinlich um Spam handelt. In den letzten Jahren konnte bei der Verarbeitung von Emails durch die Anwendung von *NLP* eine Verbesserung bei der Erkennung und Verwaltung von Spam erreicht werden wie z.B. in [GD13] erläutert.

2.3.3 Sprachassistenten

Bereits seit einigen Jahren erfreuen sich die zahlreichen Sprachassistenten immer größter Beliebtheit. Siri (Apple), Alexa (Amazon) und Cortana (Microsoft) sind dabei nur einige wenige Produkte, welche durch einen immer weiterwachsenden Funktionsumfang und eine immer tiefere Integration in vorhandene Systeme verstärkt auch im Unternehmensumfeld Anwendung finden. Dabei wird vor allem Cortana im Windows Umfeld sehr stark von Microsoft beworben und zahlreiche neue Möglichkeiten geschaffen, wie der Arbeitsalltag erleichtert werden sollte. Das Spektrum reicht dabei von der Suche nach Dokumenten, bis zum Planen von Terminen inkl. einer Rückmeldung, falls der gewünschte Zeitpunkt nicht möglich ist. Dabei ist eine sehr starke Verknüpfung mit den unterschiedlichen Systemen wie z.B. Exchange oder Office 365 erforderlich.

Im Folgenden wird ein Beispiel skizziert, welches mit Hilfe des von Microsoft entwickelten Sprachassistenten Cortana getestet wurde. Prinzipiell ist dieses Szenario auch mit anderen Sprachassistenten möglich.

Der Anwender möchte beispielsweise einen Termin für den 30. Oktober 2017

um 10:00 Uhr planen. Cortana reagiert dabei auf die Spracheingabe und nach einer erfolgreichen Verarbeitung der Eingabe, wird in Exchange überprüft, ob der Termin belegt ist. Für diesen Tag ist bereits ein Termin eingeplant und Cortana startet eine bi-direktionale Kommunikation, indem dem Anwender mitgeteilt wird, dass der Termin bereits belegt ist und einige verschiedene Optionen aufgezeigt werden. Der Anwender entscheidet sich dafür, dass der Termin der bereits vorhanden ist verschoben wird. In einer Rückmeldung informiert Cortana den Anwender, dass der Termin verschoben und der neue Termin eingetragen wurde. Cortana erkundigt sich, ob alle Teilnehmer des verschobenen Termins informiert werden sollten. Der Anwender antwortet mit Ja und Cortana leitet im Hintergrund die nötigen Schritte ein.

Dieser Fall zeigt nur eines der Beispiele, in denen ein Sprachassistent auch im Unternehmensumfeld Anwendung finden kann.

Kapitel 3

Anwendung von NLP in IT-Prozessen

3.1 Allgemeines

In diesem Kapitel wird erläutert, wie *Natural Language Processing* in ausgewählten IT-Prozessen Anwendung finden kann, sowie die Verbesserung die dadurch erwartet werden. Es wird darauf eingegangen, welche technischen Limitierungen bzw. welche Probleme bei der Anwendung von *NLP* entstehen, sowie eine kurze Bewertung bezüglich der Machbarkeit abgegeben.

Für die Analyse der Anwendungen von *NLP*, wird im ersten Schritt der Ist-Zustand des Prozesses analysiert. Dazu wird der Ablauf mit Hilfe der *BPMN*-Notation dargestellt. Es wird erläutert wie die einzelnen Schritte im Detail abgearbeitet werden und welche Ergebnisse bei den einzelnen Schritten erwartet werden. Dabei werden am Anfang eher einfachere Maßnahmen beschrieben, deren Umsetzung nur wenige Änderungen nach sich ziehen und einen eher geringen Aufwand bedeuten, sowie einige eher komplexere Maßnahmen die teilweise weitere Voraussetzung, wie zum Beispiel, das Vorhandensein einer Wissensbasis, erfordern. Es wird ein Überblick gegeben, welche Schwierigkeiten sich ergeben und welche Vorteile durch die Anwendung von *NLP* für die ProzessteilnehmerInnen und den Prozess erzielt werden können.

3.2 Das Beispielunternehmen

Das Unternehmen *XYZ Softwarehaus*, welches in den folgenden Abschnitten skizziert wird, ist ein Softwareentwicklungsunternehmen. Das *XYZ Softwarehaus* entwickelt und vertreibt dabei seine Software für die Planung von Meetings. Dabei wurden einige zentrale Geschäftsprozesse festgelegt, die zwar mit Softwareunterstützung abgewickelt werden, jedoch keine Anwendung von *NLP* stattfindet. Vor allem der Supportprozess stellt einen enorm wichtigen Prozess im Unternehmen dar, bei dem sich zahlreiche Aufgaben automatisieren lassen.

3.3 Supportprozess

Für die Abwicklung von Probleme oder allgemeinen Anfragen wurde im Unternehmen ein Supportprozess definiert. Dieser gibt vor, welche Schritte zu unternehmen sind, sobald eine Supportanfrage eines Kunden oder einer Kundin gestellt wird. Der Prozess endet mit einer Lösung für das Problem und dem Informieren des Kunden über diese.. In Abbildung 3.1 ist der Prozess in *BPMN-Notation* dargestellt.

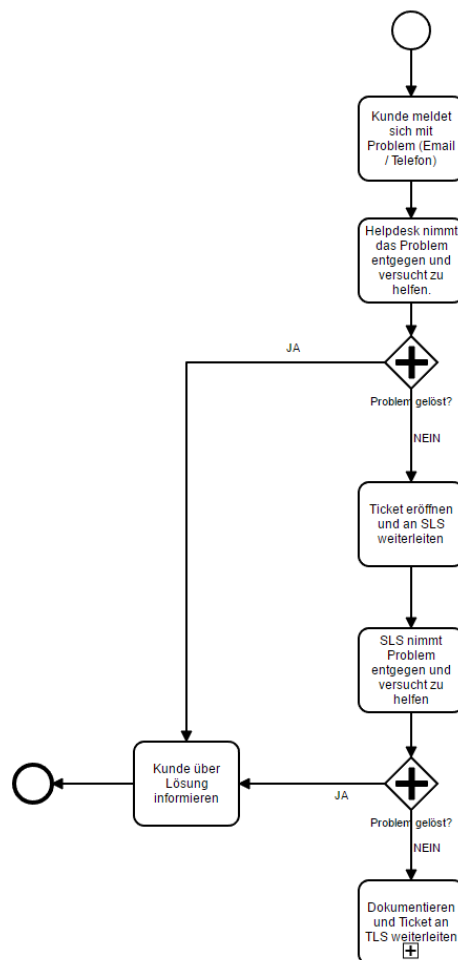


Abbildung 3.1: Supportprozess des XYZ Softwarehauses

3.3.1 Ablauf

Der erste Schritt des Prozesses besteht aus der Anfrage eines Kunden oder einer Kundin. Dies geschieht meist per Email, kann aber auch durch einen Telefonanruf erfolgen. Dabei entstehen abhängig vom gewählten Medium unterschiedliche Daten. Diese stellen sich entweder in Form von Lautsprache, oder als geschriebener Text dar. Im Folgenden wird ausschließlich auf die Anfrage per Email eingegangen.

Der Helpdesk nimmt die Anfragen des Kunden entgegen und versucht das Problem zu lösen. Dazu stehen dem Helpdesk mehrere Möglichkeiten zur Verfügung. In den meisten Fällen wird der Mitarbeiter oder die Mitarbeiterin am Helpdesk versuchen, Informationen zu dem gemeldeten Problem in der Wissensbasis zu finden. Dazu wird die Wissensbasis durchsucht. Es werden Textausschnitte oder Schlagwörter verwendet die das Anliegen des Kunden möglichst gut wiedergeben. Wurde eine Lösung für das Problem gefunden, wird diese Lösung vom Helpdeskmitarbeiter oder der Helpdeskmitarbeiterin angewendet oder die Informationen direkt an den Kunden oder die Kundin weitergegeben. Falls keine Information in der Wissensbasis gefunden werden kann, wird versucht, das Problem durch implizites Wissen des Mitarbeiters oder der Mitarbeiterin zu lösen. Falls auf diesem Weg eine Lösung gefunden werden kann, wird der Mitarbeiter oder die Mitarbeiterin die Informationen zur Lösung in der Wissensbasis hinzufügen und den Kunden oder die Kundin über die Lösung des Problems informieren. Für jede Lösung wird dem Kunden oder der Kundin zusätzlich ein Link zu dem Artikel mit der Lösung in der Wissensbasis übermittelt. Dies führt im Idealfall dazu, dass der Kunde oder die Kundin in Zukunft ohne Hilfe des Helpdesks zu einer Lösung findet.

Kann auf Grund der Komplexität des Problems im ersten Schritt vom Helpdesk keine Lösung gefunden werden, wird überprüft, ob bereits ein Ticket für dieses Problem geöffnet wurde. Möglicherweise ist das gemeldete Problem bereits bei anderen Kunden aufgetreten und daher auch schon im Ticketsystem vermerkt. Dieses manuelle Auffinden der Tickets ist sehr häufig mit einem vermehrten Zeitaufwand verbunden und häufig werden doppelt erstellte Tickets erst spät entdeckt. Falls das Problem noch nicht in Form eines Tickets beschrieben wurde, wird ein Ticket erstellt, welches alle bereits im Helpdesk erhaltenen Informationen enthält und an den sogenannten *Second Level Support (SLS)* weitergeleitet wird. Wenn im SLS eine Lösung gefunden werden kann, folgen die gleichen Schritte wie bei erfolgreicher Abarbeitung im Helpdesk: Der Kunde oder die Kundin wird informiert und die Wissensbasis ergänzt. Zusätzlich wird dem Helpdesk eine Information gegeben, dass die Wissensbasis um die nötigen Informationen erweitert wurde, sodass bei zukünftigen Anfragen dieser Art eine Lösung möglicherweise schon von einem Mitarbeiter oder einer Mitarbeiterin des Helpdesks gefunden werden kann.

Wenn das Problem auch im SLS nicht gelöst werden kann, geht es schließlich zum sogenannten *Third Level Support (TLS)*. Dieser Teil des Prozesses ist in Abbildung 3.1 nur als Subprozess modelliert, da er sich sehr ähnlich gestaltet wie der Prozessschritt im SLS. Das Ergebnis dieses Prozessschrittes ist erneut, die Rückmeldung an den Kunden oder die Kundin und die Erweiterung der Wissensbasis, sowie das Informieren des Helpdesks und des SLS über die Erweiterung.

Im Endeffekt sollte durch die kontinuierliche Erweiterung und Verbesserung der Wissensbasis ein System geschaffen werden, welches es ermöglicht, dass immer mehr Probleme direkt vom Kunden oder der Kundin selbst gelöst werden können. Es sollte auch dazu beitragen, dass mehr einfache Probleme direkt vom Helpdesk gelöst werden können, sodass der Kunde oder die Kundin schnellere Rückmeldungen bekommen und der SLS und der TLS entlastet werden. Vor allem technische Probleme werden sehr häufig nur im SLS oder im TLS lösbar sein, da im Helpdesk schlichtweg keine Zeit für die Lösung dieser bleibt.

3.3.2 Verbesserung 1 - Kategorisierung der Anfrage

Die erste Verbesserung die im Zuge des Supportprozesses vorgenommen werden kann ist eine Kategorisierung der Anfrage. Die meisten Email Programme und Mailserver bieten bereits eine grundlegende Kategorisierung über sogenannte Regeln an. Diese Regeln entsprechen dabei sehr stark vereinfachten Booleschen Abfragen. Es kann zum Beispiel festgelegt werden, dass alle Emails die vom gleichen Absender kommen, in ein bestimmtes Unterverzeichnis verschoben werden. In Abbildung 3.2 ist ein Dialog für das Festlegen von Regeln in *Microsoft Office 2016* dargestellt. Diese regelbasierte Ansatz bietet bereits eine sehr einfache Möglichkeit für die Vorsortierung der Emails. Die Problematik die sich mit diesem Ansatz ergibt ist der Konfigurationsaufwand für das Erstellen dieser Regeln. Vor allem bei sehr vielen Regeln und großen Systemen wird dieser sehr groß und teilweise unüberschaubar. Auf Grund der Tatsache, dass mit diesen Regeln nur sehr einfache Regeln abgebildet werden können, ist es zum Beispiel nicht möglich, alle Emails von einer gemeinsamen Domain (microsoft.com) in ein Unterverzeichnis weiterzuleiten. Weiters gibt es keine Möglichkeiten die Mails auf den Inhalt zu prüfen und Regeln anzugeben, die dazu führen würden, dass Mails abhängig vom Inhalt kategorisiert und weitergeleitet werden.

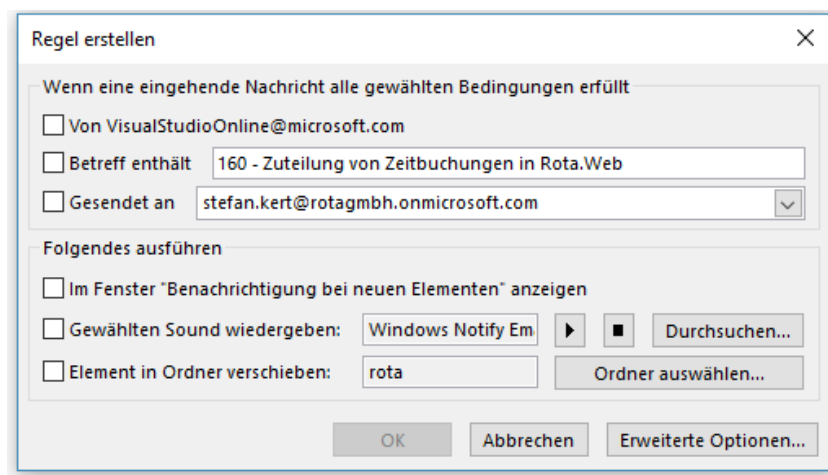


Abbildung 3.2: Dialog zum Festlegen von Regeln in MS Outlook

Hier kann eine Anwendung von *NLP* zum Beispiel dazu dienen, dass abhängig vom Inhalt der Email, eine Kategorie gefunden wird, in welche diese Email eingeordnet werden kann. Hier einige Beispiele für solche Kategorien:

- Hardwareproblem
- Softwareproblem
- Verkauf Lizenzerweiterung

Die Email kann schließlich abhängig von der Kategorie zugeordnet werden und direkt an die gewünschten Stellen weitergeleitet werden. Das folgende Beispiel sollte zur Veranschaulichung dienen:

Kunde A stellt eine Anfrage über die Möglichkeiten der Lizenzerweiterung. Die Email geht direkt in den sogenannten *Office-Posteingang* der als gemeinsames Postfach für das Unternehmen verwendet wird. Dieser Posteingang wird vom Helpdesk verwaltet. Nach dem Erhalt der Email wird ein Mitarbeiter oder eine Mitarbeiterin des Helpdesks diese Überprüfen und anschließend manuell in die Kategorie *Verkauf Lizenzerweiterung* einordnen, wodurch die Email schließlich zum Vertrieb weitergeleitet wird.

Dieses Beispiel zeigt klar das Verbesserungspotential für diesen Prozess auf. Im Moment wird die Kategorisierung der Email manuell von einem Mitarbeiter oder einer Mitarbeiterin des Helpdesks vorgenommen. Diese Aufgabe könnte durch *NLP* automatisiert erledigt werden, sodass der Aufwand für den Helpdesk hier sehr stark verringert wird.

Vorteile

Für diese Verbesserung ergeben sich folgende Vorteile:

- Bessere Vorverarbeitung von Anfragen
- Automatisches Zuordnen und Kategorisieren von Anfragen
- Komplexere Regeln möglich

Machbarkeit

Die Kategorisierung von Daten ist eine Standardanwendung von *NLP* und es gibt bereits zahlreiche Gebiete wo die Kategorisierung im Zusammenhang mit Text erfolgreich eingesetzt wird. Bei Kategorisierungs- oder Zuordnungsaufgaben werden im *NLP* verschiedene sogenannte *Clustering*- und *Classification*-Methoden verwendet.

Als Clustering wird die Möglichkeit bezeichnet, Daten (meist auch als Dokumente bezeichnet) in Klassen oder Kategorien, sogenannte Cluster, von gleichen Daten zu gruppieren. Dabei sind beim Clustering zwei wichtige Kriterien zu erfüllen:

1. Daten in einem Cluster sollten ähnlich sein
2. Daten aus anderen Clustern sollten sich nicht ähneln.

Das Zusammenfassen gleicher Daten zu neuen Gruppen ist das Ziel des *Clusterings*. Die Aufgabe bei der *Classification* besteht darin, dass im Gegensatz zum *Clustering* eine Gruppierung der Daten auf Basis von vorgegebenen Kategorien statt. Die Gruppierung erfolgt also nicht auf Grund der vorhandenen Daten sondern die neuen Daten werden in vorgegebene Kategorien eingeordnet.

Ein Beispiel für die Anwendung von Clustering ist die Kategorisierung von wissenschaftlichen Artikeln, welche in dem Artikel *Reconstructing the Giant: Automating the Categorization of Scientific Articles with Deep Learning Techniques* [DHH17] beschrieben wird. Die *Clustering*-Methode, die im Artikel beschrieben wird, ist das *Hierarchical Agglomerative Clustering (HAC)*. Dieses Verfahren zeichnet sich dadurch aus, dass es am Anfang alle Dokumente der Datenbasis in eigene *Cluster* einordnet. Es wird für jeden dieser Cluster der ähnlichste Cluster gewählt und diese beiden Cluster formen einen neuen Cluster. Diese Verfahren wird rekursiv

durchgeführt und verarbeitet alle Cluster, bis alle Artikel zu einem gemeinsamen Cluster zusammengefasst wurden. In dem genannten Artikel wird das Modell außerdem durch einen *Deep-Learning* Technik trainiert.

Eine Anwendung von *Clustering* wäre für die Verbesserung in diesem Prozess grundsätzlich möglich, da bereits vordefinierten Kategorien von Anfragen existieren, ist die Anwendung von *Classification*-Methoden hier zu empfehlen.

In dem Artikel *Classification of Text, Automatic* wird eine Möglichkeit für die Anwendung von Classification-Methoden im Zusammenhang mit Texten beschrieben [Seb06]. Bei der *Classification* wird grundsätzlich zwischen der *Single-Label-Classification* und der *Multi-Label-Classification* unterschieden. Der Unterschied zwischen diese beiden Verfahren besteht darin, dass bei der *Single-Label-Classification* nur eine Klasse zugeordnet wird, wohingegen bei der *Multi-Label-Classification* mehrere Klassen zugeordnet werden können. In dem aktuellen Anwendungsfall, der dazu dienen sollte die Zuordnung von Anfragen zu bestimmten Kategorien zu finden, ist in erster Linie die *Single-Label-Classification* von Interesse, da für diesen Prozess davon ausgegangen wird, dass die Anfragen immer nur an eine Stelle weitergeleitet und somit nur einer Kategorie zugeordnet werden. Die *classification* Methoden die in dem genannten Artikel empfohlen werden sind die folgenden:

- Vektor Maschinen
- Boosting

Die Hauptgründe für die Auswahl dieser beiden Verfahren sind die Genauigkeit, sowie die sehr gute Performance im Zusammenhang mit der Klassifizierung von Text.

Auch hier ist es wieder erforderlich, dass das initiale Modell trainiert wird, sodass eine zukünftige Klassifizierung der Anfragen mit einer hohen Genauigkeit erfolgen kann. Die Datenbasis für das initiale Trainieren des Modells besteht dabei aus den bereits manuell zugeordneten Anfragen. Dabei ist es von enormer Wichtigkeit, dass für die Trainingsdaten richtig zugeordnete Anfragen verwendet werden, um das Primärziel, eine genaue Zuordnung zukünftiger Anfragen, zu erfüllen. Es sollte schlussendlich möglich sein, weitere Zuordnung basierend auf dem trainierten Modell zu finden und automatisiert zu verarbeiten.

3.3.3 Verbesserung 2 - Hinzufügen von Informationen aus der Wissensbasis und Verknüpfen mit Anfragen aus der Vergangenheit

Ein zentraler Bestandteil des Prozesses, ist die kontinuierliche Erweiterung der Wissensbasis, sowie die Verwendung der darin enthaltenen Informationen. Im Moment wird versucht, durch manuelles Durchsuchen dieser Wissensbasis, Informationen, die zum Lösen eines Problems beitragen, zu erhalten. Dies ist meist mit einem erhöhten Aufwand verbunden. Der Mitarbeiter oder die Mitarbeiterin des Helpdesks muss die Wissensbasis nach den Schlagwörtern, oder Inhalten die in der Anfrage enthalten sind durchsuchen, die Informationen ihrer Relevanz nach beurteilen und schließlich, die Ergebnisse in eine Antwort für den Kunden oder die Kundin umformen. Diese Vorgehensweise liefert sehr häufig gute Ergebnisse die genau auf das Problem des Kunden zugeschnitten sind, benötigt aber mehr Zeit für die Abarbei-

tung.

Ein weiterer Teil dieser Verbesserung besteht in der Zuordnung von Fällen aus der Vergangenheit die ähnlich sind wie die aktuelle Anfrage. Ähnlich wie beim Auffinden der Informationen in der Wissensbasis kann hier durch die Anwendung von *NLP* ein Vergleich zu den Anfragen aus der Vergangenheit gezogen werden und diese für den Mitarbeiter oder die Mitarbeiterin des Helpdesks zur Verfügung gestellt werden. Der Vorteil besteht darin, dass selbst wenn keine Informationen in der Wissensbasis vorhanden sind, trotzdem Informationen aus bereits gelösten Fällen bezogen werden können.

Mit Hilfe von *NLP* kann eine teilweise Automatisierung des Durchsuchen der Wissensbasis, beziehungsweise Zuordnung zu Anfragen aus der Vergangenheit erreicht werden. Die Anfragen, welche für den Support bestimmt sind und Probleme darstellen, werden mit Informationen aus der Wissensbasis und älteren Anfragen verglichen. Dieser Vergleich kann mehrere unterschiedliche Ergebnisse liefern. Vereinfacht gesagt, werden die Anfragen und die Informationen aus der Wissensbasis und den älteren Anfragen in Kategorien aufgeteilt. Dies ist grundsätzlich sehr ähnlich zu der Verbesserung in Abschnitt 3.3.2, ein sehr zentraler Unterschied besteht aber darin, dass die Anfragen mehreren Kategorien zugeordnet werden müssen und dass die Kategorien initial noch nicht vorhanden sind. Die Kategorien müssen erst aus den vorhandenen Informationen aus der Wissensbasis und aus Anfragen aus der Vergangenheit gebildet werden.

Die Zuordnung zu mehreren Kategorien hat in erster Linie den Hintergrund, dass Probleme sich häufig nicht eindeutig in eine Kategorie passen. Meist erstreckt sich ein Problem über mehrere Kategorien. In diesem Fall, würde eine sogenannten *harte Zuordnung*, wodurch eine Zuordnung nur zu einer Kategorie möglich ist nicht ausreicht. Weiters kommt es sehr häufig vor, dass Kunden mehrere Probleme in einer Anfrage beschreiben, oder die Probleme einen Zusammenhang mit anderen Kategorien haben, die für das Ermitteln der Lösung von Bedeutung ist.

Wenn die Zuordnung zu einer oder mehreren Kategorien erfolgt ist, können die Informationen aus der Wissensbasis, sowie die Informationen aus den älteren Anfragen, welche der gleichen Kategorie wie die Anfrage zugeordnet sind, ermittelt werden. Hier wäre eine weitere Möglichkeit zur Reihung dieser Ergebnisse, dass eine Sortierung nach der Relevanz der einzelnen gefunden Informationen erfolgt. Das Ergebnis ist eine der Relevanz nach sortierte Liste an Informationen.

Der Mitarbeiter oder die Mitarbeiterin des Helpdesks hat nun die Möglichkeit, dem Kunden oder der Kundin eine Information aus dem erhaltenen Ergebnis zurückzusenden, wodurch noch eine weitere manuelle Überprüfung der erhaltenen Informationen auf Korrektheit möglich ist.

Hier wäre es möglich, durch die Rückgabe der Auswahl der Information durch den Mitarbeiter oder die Mitarbeiterin des Helpdesks, das Modell für die Kategorisierung weiter zu trainieren. Wenn beispielsweise keiner der Artikel zu der gestellten Anfrage passt, kann durch zwei Schritte Einfluss auf das Modell genommen werden. Im ersten Schritt sollte die Kategorie überprüft werden, welcher die Anfrage zugeordnet wurde. Falls diese Zuordnung falsch ist, muss eine manuelle Zuordnung zu einer passenden Kategorie erfolgen. Diese Zuordnung sollte in jedem Fall wieder ins Modell zurückfließen, sodass die Anfragen in Zukunft besser zugeordnet

werden. Der zweite Schritt ist eine manuelle Suche durch den Mitarbeiter oder die Mitarbeiterin des Helpdesks. Das Ergebnis dieser manuellen Suche kann ebenfalls zum trainieren des Modells verwendet werden. Im Idealfall führt dies zu besseren Suchergebnissen in der Zukunft.

Dieses Trainieren des Modells kann schließlich auch bei der manuelle Suche in der Wissensbasis unterstützen, von dem schlussendlich auch Kunden und die Mitarbeiter und Mitarbeiterinnen des Helpdesks profitieren, da die Ergebnisse, die zu Anfragen gefunden werden, durch das Trainieren des Modells immer genauer werden.

Als weiterer Optimierungsschritt für den Prozessablauf wird dem Mitarbeiter oder der Mitarbeiterin des Helpdesks die Möglichkeit gegeben, aus der erhaltenen Informationen, welche als Ergebnis der automatischen Suche geliefert wurde, eine Email generieren zu lassen, die alle Informationen, wie Links, Beschreibungen oder weiterführende Informationen enthält.

Ein weiterer Schritt, der zu einer Verbesserung des Modells und der Wissensbasis beitragen kann, ist das miteinbeziehen der Rückmeldung des Kunden oder der Kundin. Dem Kunden oder der Kundin wird über eine Bewertungs- und eine Kommentarfunktion die Möglichkeit gegeben, die erhaltenen Informationen zu bewerten. Diese Information kann nach einer Prüfung durch einen Mitarbeiter oder eine Mitarbeiterin des Helpdesks wieder ins Modell integriert werden und so das Modell weiter trainieren. Hier, wie auch bei der Auswahl durch den Mitarbeiter oder die Mitarbeiterin des Helpdesks, besteht natürlich die Gefahr, dass das Modell mit falschen Informationen trainiert wird und dadurch die Ergebnisse in Zukunft verschlechtert werden. Um diese Problematik so gering wie möglich zu halten, erfordert es eine hohe Genauigkeit beim weitergeben der Informationen an das Modell.

Vorteile

Für diese Verbesserung ergeben sich folgende Vorteile:

- Schnelles und automatisches Erhalten von Informationen aus der Wissensbasis
- Geringer manueller Aufwand für Helpdesk
- Manuelle Suche wird durch Trainieren des Modells verbessert
- Auffinden bereits behandelter Anfragen

Machbarkeit

Wie bereits in Abschnitt 3.3.2 kann auch hier auf bereits vorhandenen Algorithmen und Vorgehensweisen für die Kategorisierung von Texten zurückgegriffen werden. Auf Grund der Tatsache, dass es sehr viele Kategorien von Problemen gibt und Anfragen auch mehrere Probleme beinhalten können, sowie die Tatsache, dass sich Probleme auf mehrere Kategorien verteilen können, ist die Komplexität dieser Verbesserung signifikant höher als die Verbesserung in Abschnitt 3.3.2.

Grundsätzlich wäre hier das Verwenden einer *Classification*-Methode möglich. Um die Anforderung der Zuordnung zu mehreren Kategorien zu erfüllen, kann hier auf eine sogenannte *Multi-Label-classification* gesetzt werden. Da für diese

Verbesserung keine Datenbasis vorhanden ist, die eine Zuordnung zu vorgegebenen Kategorien ermöglicht, ist für diese Verbesserung die Anwendung einer *Clustering*-Methode empfehlenswert.

Für den Fall, dass eine Zuordnung mittels Clustering erfolgen sollte, ist die Anwendung von *Fuzzy-Clustering*-Methoden notwendig. *Fuzzy-Clustering*-Methoden ermöglichen die Zuordnung von Daten zu mehreren verschiedenen Clustern. Ein Beispiel für die Anwendung von *Fuzzy-Clustering*-Methoden im Zusammenhang mit Text ist in dem Artikel *A Scalable Hierarchical Fuzzy Clustering Algorithm for Text Mining* beschrieben. [RS04]. Das in diesem Artikel beschriebene Problem ist dem Problem, welches mit dieser Verbesserung gelöst werden sollte sehr ähnlich. Die Anwendung, die beschrieben wird, behandelt die Zuordnung von Themen zu unterschiedlichen Domänen. Dabei hängen Themen, die sich auf eine bestimmte Domäne beziehen, mit anderen Themen aus dieser Domäne zusammen und stehen in einer Beziehung zu diesen. Häufig beziehen sich Themen auch auf andere Domänen und sind daher auch für diese anderen Domänen mehr oder weniger relevant. Gleich wie bei dem Problem, welches sich im Zusammenhang mit den Problemkategorien ergibt, müssen also Themen mehreren Domänen gleichzeitig zugeordnet werden. Durch die Anwendung von *Fuzzy-Clustering* Methoden kann eben dieses Problem gelöst werden. Die Themen können gleichzeitig mehreren Domänen zugeordnet werden, wodurch sich teilweise hilfreiche Beziehungen zwischen den einzelnen Domänen und Themen ergeben, die bei einem normalen *clustering* nicht ersichtlich wären.

Durch das *Clustering* können neue Anfragen in Cluster eingeordnet werden. Diese Cluster beinhalten dabei Artikel aus der Wissensbasis, sowie Anfragen aus der Vergangenheit. Durch diese Zuordnung können die erwähnten Vorteile erzielt werden, und es ist dem Mitarbeiter oder der Mitarbeiterin des Helpdesks möglich, abhängig von der Anfrage, sich auf Informationen aus der Wissensbasis, sowie auf Informationen aus Anfragen aus der Vergangenheit zu beziehen.

Eine Herausforderung die sich im Zusammenhang mit dieser Verbesserung ergibt, ist das Befüllen der Wissensbasis. Wenn diese nur unzureichend befüllt ist, oder überhaupt nicht vorhanden, werden sich die durch diese Verbesserung beschriebenen Vorteile für das Unternehmen nur sehr begrenzt ergeben. Für diesen Fall bietet diese Verbesserung immer noch die Möglichkeit, Lösungen für neue Anfragen, in Anfragen aus der Vergangenheit zu erhalten, da diese dem Cluster der aktuellen Anfrage zugeordnet sind.

Um den erwähnten Vorteil der Relevanz zu erzielen, muss für die *Clustering* Methode zusätzlich eine Möglichkeit gefunden werden, die Relevanz der Information mit einfließen zu lassen. Dies kann eben über den erwähnten manuellen Input durch Mitarbeiter oder Mitarbeiterinnen des Helpdesks oder auch den Kunden erfolgen. Das Modell für die Suche der Daten sollte im Idealfall laufend trainiert werden, sodass die Ergebnisse im Laufe der Zeit immer besser werden. Dies bedeutet vor allem am Anfang einen erhöhten manuellen Aufwand. Häufig gibt es bereits eine Auswertung wie hilfreich gewisse Informationen aus der Wissensbasis sind, dies kann natürlich für das Reihensortieren der Informationen nach Relevanz verwendet werden.

3.3.4 Verbesserung 3 - Erkennen von duplizierten Tickets

Kann das Problem von einem Mitarbeiter oder einer Mitarbeiterin des Helpdesk nicht gelöst werden, wird ein sogenanntes Ticket erstellt. Dieses Ticket enthält Informationen über das Problem, den Kunden oder die Kundin, wo das Problem aufgetreten ist und weitere Metainformationen wie Betriebssystem, Version der Applikation, oder auch ob das Problem nur zu einer bestimmten Uhrzeit auftritt. Vor der Erstellung dieses Tickets, wird das Ticketsystem noch auf ähnliche Probleme durchsucht. In manchen Fällen lassen sich bei dieser Suche bereits ähnliche Fälle für dieses Problem finden. Ist dies der Fall wird das neue Ticket mit dem alten Ticket verknüpft, sodass beide gemeinsam gelöst werden können. Dieser Schritt erfordert erneut eine manuelle Suche die durch einen Mitarbeiter oder eine Mitarbeiterin des Helpdesks erfolgen muss. Hier könnte eine Verbesserung durch die Anwendung von *NLP* dadurch erreicht werden, dass sobald ein Ticket erstellt wird, die Wissensbasis automatisch nach ähnlichen oder gleichen Tickets durchsucht wird. Es sollten also Duplikate von Tickets erkannt werden und dem Mitarbeiter oder der Mitarbeiterin eine Rückmeldung gegeben werden, dass dieses oder ein ähnliches Ticket bereits erstellt wurde und ob diese Tickets verknüpft werden sollten. Auch hier ist eine manuelle Eingabe des Mitarbeiters oder der Mitarbeiterin erforderlich. Der Mitarbeiter oder die Mitarbeiterin prüft, ob die ermittelten Ergebnisse wirklich übereinstimmen. Ist dies der Fall, wird das Ticket zugeordnet.

Eine weitere Möglichkeit im Zuge dieser Verbesserung, ist das in Abschnitt 3.3.3 beschriebene Verfahren zum Auffinden von ähnlichen Anfragen. Die Tickets, welche diesen ähnlichen Anfragen zugeordnet sind können als potentielle Duplikate des aktuellen Tickets in Betracht gezogen werden. Die Zuordnung erfolgt abhängig davon, ob das Ticket bereits abgeschlossen ist oder nicht. Falls das Ticket bereits abgeschlossen wurde, wird dem Mitarbeiter oder der Mitarbeiterin die Information gegeben, dass bereits eine Lösung vorhanden ist. Falls nicht kann erneut eine Verknüpfung der Tickets durch ein manuelles Eingreifen des Mitarbeiters oder der Mitarbeiterin erfolgen.

Vorteile

Für diese Verbesserung ergeben sich folgende Vorteile:

- Schnelles und automatisches Auffinden von duplizierten Tickets
- Weniger Tickets werden zum SLS oder zum TLS weitergeleitet, da duplizierte Tickets bereits zuvor zusammengefasst werden

Machbarkeit

Für das Auffinden von Duplikaten ergeben sich die gleichen Anforderungen wie für die Verbesserung in Abschnitt 3.3.3. Auch hier müssen ähnliche Dokumente (Tickets) gruppiert werden, sodass diese verlinkt werden können. Dabei kann wieder auf *Clustering*- oder auf *Classification*-Methoden gesetzt werden. Auch in diesem Fall empfiehlt sich erneut die Verwendung einer *Clustering*-Methode. Genau wie bei der Verbesserung in Abschnitt 3.3.3, beschränken sich die Tickets meist nicht auf einen Cluster, sondern können mehreren Clustern zugeordnet sein.

Beim Training kann für diese Anwendung auf die gesamte Datenbasis der vorhandenen Tickets zurückgegriffen werden. Mit Hilfe einer *Clustering*-Methode werden sämtliche Tickets in Cluster eingeordnet, sodass das initiale Modell entsteht. Neue Tickets werden diesen Clustern zugeordnet und werden somit gemeinsam mit ähnlichen Tickets gruppiert. In einer einfachen Ausbaustufe dieser Verbesserung könnten auf eine Anfrage nach ähnlichen Tickets, alle Tickets eines Clusters gelistet werden. Der Mitarbeiter oder die Mitarbeiterin des Helpdesks hat daraufhin die Möglichkeit, eine manuelle Auswahl der Duplikate vorzunehmen. In einer späteren Ausbaustufe, kann zusätzlich zwischen den einzelnen Tickets verglichen werden und kleinere Cluster gebildet werden, sodass nur noch bei einer sehr starken Ähnlichkeit eine Zuordnung erfolgen kann.

Kapitel 4

Fazit

Grundsätzlich lässt sich sagen, dass die Anwendung von *Natural Language Processing* vor allem in Prozessen in denen viele Daten in Form von Text verarbeitet werden eine Verbesserung darstellen. Dies ist vor allem dort der Fall, wo eine Kommunikation mittels Chat, Email oder Onlineformular stattfindet. Für die meisten Anwendungen von *NLP* ist eine manuelle Überprüfung durch einen Prozessteilnehmer oder eine Prozessteilnehmerin empfehlenswert, da die erhaltenen Ergebnisse häufig vom erwarteten Ergebnis abweichen können. Durch diese manuelle Überprüfung kann das System außerdem trainiert werden, sodass die gelieferten Ergebnisse in Zukunft möglicherweise besser dem Erwarteten entsprechen.

Es wäre für Unternehmen von großem Vorteil, sich mit der Thematik des *NLP* zu beschäftigen. Bereits rudimentäre Anwendungen, bringen Vorteile für die Prozesse und die Teilnehmer des Prozesses mit sich. Der Supportprozess ist hierfür ein gutes Beispiel. Bereits durch eine einfache Texterkennung kann eine Weiterleitung der Anfragen an die richtigen Stellen passieren. Der Aufwand für die Implementierung hält sich in Grenzen und erspart den Mitarbeitern die sich um die Anfragen kümmern häufig einiges an Arbeit.

Die skizzierten Verbesserungen sind dabei unabhängig von der Sprache bereits jetzt sehr gut einsetzbar, da kein Parsen des Textes im eigentlichen Sinn, sondern ein Vergleich des Inhaltes vorgenommen wird.

Für komplexere Anwendungsszenarien, wie dem Interpretieren eines Textes, gibt es vor allem im englischsprachigen Raum sehr stark Fortschritte. Im deutschen ist die Unterstützung teilweise nur rudimentär und nur in sehr speziellen Szenarien verwendbar. Dies liegt unter anderem daran, dass die englische Sprache auf Grund ihrer geringeren Komplexität leichter zu verarbeiten ist. Ein weiterer Grund ist das wachsende Interesse von großen Unternehmen wie Microsoft, Apple oder Amazon an der Interpretation von Texten und Sprache im Allgemeinen und den Investitionen, die vor allem für die Unterstützung von englischer Sprache getätigt werden.

Quellenverzeichnis

Literatur

- [DHH17] David Dann, Matthias Hauser und Jannis Hanke. „Reconstructing the Giant: Automating the Categorization of Scientific Articles with Deep Learning Techniques“ (2017) (siehe S. 1, 13).
- [GD13] Rohit Giyanani und Mukti Desai. „Spam detection using natural language processing“. *IOSR Journals (IOSR Journal of Computer Engineering)* 1.16 (2013), S. 116–119 (siehe S. 7).
- [Nob+16] Chikashi Nobata u. a. „Abusive language detection in online user content“. In: *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2016, S. 145–153 (siehe S. 1).
- [RDP14] Cristina Rădulescu, Mihaela Dinsoreanu und Rodica Potolea. „Identification of spam comments using natural language processing techniques“. In: *Intelligent Computer Communication and Processing (ICCP), 2014 IEEE International Conference on*. IEEE. 2014, S. 29–35 (siehe S. 2).
- [RS04] MESM Rodrigues und L Sacks. „A scalable hierarchical fuzzy clustering algorithm for text mining“. In: *Proceedings of the 5th international conference on recent advances in soft computing*. 2004, S. 269–274 (siehe S. 17).
- [Seb06] Fabrizio Sebastiani. „Classification of text, automatic“. *The encyclopedia of language and linguistics* 14 (2006), S. 457–462 (siehe S. 14).
- [Xio+16] Wayne Xiong u. a. „Achieving human parity in conversational speech recognition“. *arXiv preprint arXiv:1610.05256* (2016) (siehe S. 5).