

**Thesis zur Erlangung des akademischen Grades
Bachelor of Science (B. Sc.)**

**AUTOMATISIERUNG DER
INFORMATIONSGEWINNUNG IN
BEDARFSMELDUNGEN**

von

Ricardo Valente de Matos

geboren am 30.10.1999

Matrikelnummer: 7203677

im Studiengang Wirtschaftsinformatik

der Fachhochschule Dortmund

im Fachbereich Informatik

Erstprüfer: Prof. Dr.-Ing. Guy Vollmer

Zweitprüfer: Stephan Schmeißer, M. Sc., Adessoplatz 1, 44269 Dortmund

Dortmund, den 25. Juli 2024

Sperrvermerk

Diese Bachelorarbeit zum Thema „Automatisierung der Informationsgewinnung in Bedarfsmeldungen“ enthält interne Informationen der *adesso SE*. Aus diesem Grund ist die Arbeit mit einem Sperrvermerk versehen.

Diese Bachelorarbeit ist der Öffentlichkeit nicht in ihrer vollständigen Form zugänglich zu machen. Die enthaltenen Informationen sind vertraulich zu behandeln.

Dortmund, den 25. Juli 2024

Ricardo Valente de Matos

Abstract

Das Ziel in der vorliegenden Arbeit ist es, zu klären, welche effektiven Methoden und Techniken des Information Retrieval und Data-Mining genutzt werden können, um automatisiert relevante Inhalte aus *Bedarfsmeldungen* im spezifischen Kontext der Software Entwicklung zu extrahieren und somit die Effizienz, Genauigkeit und Geschwindigkeit der Informationsgewinnung für Führungskräfte zu verbessern. Um diese Frage zu beantworten, wurde zu Beginn eine qualitative Inhaltsanalyse von Experteninterviews durchgeführt, um zu verstehen, welche die relevanten Inhalte einer *Bedarfsmeldung* darstellen. Die Ergebnisse zeigen, dass die Nutzung von den Techniken *TF-IDF*, *N-Gramm*, *Named Entity Recognition*, *POS-Tagging* und diversen Vorverarbeitungsschritten in einem hybriden System zu vielversprechenden Ergebnissen in der Strukturierung von *Bedarfsmeldungen* darstellt.

ToDo: Abstract erstellen

<-hier

Lesehinweis

Aus Gründen der besseren Lesbarkeit werden Wörter und Wortgruppen, die hervorgehoben werden oder mehrfach auftauchen, durch *kursiven* Text kenntlich gemacht. Zudem wird in dieser Ausarbeitung die Sprachform des generischen Maskulinums angewandt. Sämtliche Ausführungen sind jedoch geschlechtsunabhängig und beziehen sich damit auf alle Geschlechter.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Problemstellung	2
1.2. Ziele und Ergebnisse der Arbeit	3
1.3. Aufbau der Arbeit	4
2. Strukturierte Bedarfsmeldung	5
2.1. Bedarfsmeldungen	5
2.2. Experteninterviews	5
2.3. Ablauf der Interviews	7
2.4. Übersicht der Experten	7
2.5. Auswertungsmethode	9
2.6. Durchführung der qualitativen Inhaltsanalyse	11
2.7. Interpretation der Ergebnisse	16
2.8. Strukturierung von Bedarfsmeldungen	19
3. Konzeptionierung	21
3.1. Idee des Recommender System zur Mitarbeiterempfehlung	21
3.2. Anforderungsanalyse	24
3.2.1. Anwendungsfälle	25
3.2.2. Anforderungen	26
3.2.3. Ablauf des Systems	28
4. Literaturüberblick	29
4.1. Techniken	29
4.2. Ergebnis	37
5. Umsetzung	38
5.1. Beschreibung	38
5.2. Konkreter Ablauf der Pipeline	38
5.3. Implementationsdetails	40
5.3.1. Pipeline	40
5.3.2. Projektstruktur	40
5.3.3. Modulimplementationen	40

6. Evaluierung	48
6.1. Versuchsdurchführung	48
6.2. Beschreibung des verwendeten Datensatzes	50
6.3. Präsentation und Diskussion der Ergebnisse	51
7. Zusammenfassung und Ausblick	60
A. Anhang	62
A.1. Interviewtranskripte	62
A.1.1. Vorabtest Fragen und Antworten	62
A.1.2. Interview 2	64
A.1.3. Interview 3	68
A.1.4. Interview 4	78
A.1.5. Interview 5	82
A.2. Paraphrasierung, Generalisierung und Reduktion	87

Abbildungsverzeichnis

2.1. UML-Aktivitätsdiagramm zum Ablaufmodell der qualitativen Inhaltsanalyse nach Mayring [22]	10
2.2. UML-Aktivitätsdiagramm zum Ablaufmodell der zusammenfassenden Inhaltsanalyse nach Mayring [22]	10
2.3. Vereinfachte Darstellung zur Erfassung und Durchführung eines Projektes bei <i>adesso</i>	16
2.4. Mockup des Aufbaus einer <i>Bedarfsmeldung</i> in Jira.	18
2.5. Mockup einer strukturierten <i>Bedarfsmeldung</i>	19
3.1. Abstrakte Darstellung des Recommender System zur Mitarbeiterempfehlung.	21
3.2. Use-Case Diagramm zum System zur Strukturierung von Bedarfsmeldungen.	25
3.3. Abstrakte Darstellung des Systems zur Strukturierung von <i>Bedarfsmeldungen</i>	28
5.1. UML-Aktivitätsdiagramm der Pipeline.	39
6.1. Diagramm zur Anzahl der Wörter im Verhältnis zum Score.	52
6.2. Diagramm zum Threshold im Verhältnis zum Score.	54
6.3. Diagramm zur Anzahl von zu entfernenden Wortartenkombinationen im Verhältnis zum Score.	56

Tabellenverzeichnis

2.1. Übersicht der Experten	8
2.2. Ergebnisse der qualitativen Inhaltsanalyse.	15
5.1. Übersicht der Datenfelder	41
5.2. Untypische Wortartenkombinationen in der englischen Sprache.	44
6.1. Gegenüberstellung der Ergebnisse mit 4000 und 8070 Wörtern im Textkorpus.	53
6.2. Gegenüberstellung der Ergebnisse mit dem Threshold 0.0 und 0.05.	55
6.3. Gegenüberstellung der Ergebnisse mit 1 und 2 untypischen Wortkom- binationen.	57
6.4. Gegenüberstellung der Ergebnisse mit und ohne Übersetzungsschritt.	58
A.1. Erstellung der Kategorien aus den Experteninterviews.	97

Listings

5.1. Implementation der Filterung für Schlüsselwörter in einer Bi-Gramm Liste	43
5.2. Implementation der Filterung von Wortartenkombinationen	45
5.3. Umformung der Bi-Gramm Liste in Stichpunkte	46

1. Einleitung

In einer globalisierten und dynamischen Wirtschaftswelt sind Unternehmen zunehmend auf Projekte angewiesen, um ihre Ziele zu erreichen und Wettbewerbsvorteile zu erlangen. Die Personalbeschaffung für solche Projekte erfordert oft spezialisiertes Fachwissen und vielfältige Fähigkeiten, um erfolgreich umgesetzt zu werden. Es ist entscheidend für den Projekterfolg, dass die Personalbeschaffung die passenden Mitarbeiter für ausgewählte Projekte findet. Hier setzt die Entwicklung eines Recommender Systems zur Mitarbeiterempfehlung an. Ein solches System kann Unternehmen dabei unterstützen, den Prozess der Mitarbeiterrekrutierung und -auswahl zu optimieren. Durch die Berücksichtigung verschiedener Kriterien wie Qualifikationen, Fähigkeiten und Erfahrungen kann das Recommender-System dazu beitragen, die Auswahl effektiv zu filtern und diejenigen herauszufiltern, die am besten zu einem Projekt im Unternehmen passen. Ein solches System bietet außerdem den Vorteil, den Prozess der Mitarbeiterempfehlung zu automatisieren und zu beschleunigen. Dies ermöglicht Unternehmen, schneller auf offene Stellen zu reagieren und potenzielle Kandidaten zeitnah zu identifizieren. Dadurch wird die Effizienz der Mitarbeitersuche verbessert und die Qualität der Einstellungsentscheidungen erhöht.

Das Potenzial von Recommender Systems wurde auch bei *adesso* entdeckt und nun wird nach und nach Wege gesucht, KI-gestützte Systeme in die eigenen Prozesse zu integrieren. Im internen Projekt *adMatch* wird an einem Recommender-System zur Mitarbeiterempfehlung für ausgewählte Projekte gearbeitet. Die Umsetzung der Recommender Systems bedient sich verschiedener KI-basierten Ansätze. Als IT-Dienstleister wird *adesso* von Kunden unter anderem mit der Entwicklung individueller Softwarelösungen beauftragt. Derzeit verbringen Führungskräfte jedoch viel Zeit damit, interne Mitarbeiterinnen und Mitarbeiter manuell für Kundenprojekte zu suchen und diese dann aufgrund ihrer Erfahrungen und Fähigkeiten auszuwählen und entsprechend einzusetzen. Ein ganz entscheidender Schritt im Prozess der Mitarbeiterempfehlung ist die Vorverarbeitung der *Bedarfsmeldungen*. Diese beinhalten Informationen zu Projekten und sind eine wertvolle Informationsquelle, die Führungskräften helfen kann, die Empfehlungen effizienter zu gestalten, um dadurch wettbewerbsfähig zu bleiben. Allerdings sind diese oft umfangreich, unsortiert und komplex, was ihre effektive Nutzung erschwert. Dieser Prozess soll durch eine KI-Lösung unterstützt werden. Da es sich bei der Personalsuche um einen geschäftskritischen Prozess handelt, ist der Spielraum für Fehler gering. Im internen Projekt

adMatch wird eine durch Large Language Model-gestützte Anwendung entwickelt, die Führungskräfte bei der Suche nach geeignetem Personal für ausgewählte Projekte unterstützt. Der Ansatz des Large Language Modeling ist jedoch nicht deterministisch. Es besteht die Gefahr, dass bei gleichem Input unterschiedliche Ergebnisse erzielt werden. Somit versucht *adesso* durch den Einsatz von Methoden und Technologien neben dem Large Language Model-Ansatz deterministische Ergebnisse zu erzielen, die auf einem ähnlichen Niveau liegen. Ein Ansatz ist die Anwendung von effizienten Methoden und Techniken des Information Retrieval, um so relevante Informationen schnell und präzise aus *Bedarfsmeldungen* zu extrahieren. Die Extraktion wichtiger Schlüsselwörter, Phrasen und Themen ermöglicht es einen besseren Einblick in die Ziele, Methoden und Ergebnisse der Projekte zu bekommen. Dadurch können fundierte Entscheidungen bezüglich der Personalbesetzung getroffen und Ressourcen effizient genutzt werden.

1.1. Problemstellung

Der Staffing-Prozess kann ausschließlich von ausgewählten Mitarbeitenden von *adesso* durchgeführt werden. Um das Entlastungspotenzial für Führungskräfte durch das Gesamtsystem eines Recommender Systems für Mitarbeiterempfehlungen zu realisieren, ist eine technische Abbildung des Prozesses erforderlich. Dazu sind mehrere Schritte notwendig. Eine Informationsgewinnung aus den unstrukturierten Projekt- und Mitarbeiterdaten ist unerlässlich, um schließlich den Ähnlichkeitsvergleich für die Empfehlungen durchführen zu können. Diese Ausarbeitung befasst sich mit dem Schritt der Strukturierung und Informationsextraktion der *Bedarfsmeldungen*. Somit steht *adesso* vor der Herausforderung, relevante Informationen effizient aus umfangreichen *Bedarfsmeldungen* zu extrahieren. Obwohl diese Beschreibungen wichtige Einblicke in Ziele, Methoden und Ergebnisse liefern, können sie aufgrund ihres Umfangs und ihrer Komplexität schwer durchsuchbar und analysierbar sein. Die manuelle Identifizierung und Extraktion relevanter Inhalte ist zeitaufwendig und fehleranfällig. Daher stellt sich die Problemstellung:

Wie können wir effektive Methoden und Techniken des Information Retrieval und Data-Mining nutzen, um automatisiert relevante Inhalte aus *Bedarfsmeldungen* im spezifischen Kontext der Software Entwicklung zu extrahieren und somit die Effizienz, Genauigkeit und Geschwindigkeit der Informationsgewinnung für Führungskräfte zu verbessern.

In der Vergangenheit wurden bereits Methoden im Bereich des automatisierten Recruitings untersucht. Im Projektgeschäft sehen wir uns mit einem Problem konfrontiert, dessen Umfang jedoch präziser definiert werden kann, da die Kandidatenauswahl einem begrenzten Pool unterliegt. Besondere Relevanz hat hierbei die Erstellung einer Standardisierung der *Bedarfsmeldung*, da diese häufig unstrukturiert und mit fehlenden Informationen vorliegt.

1.2. Ziele und Ergebnisse der Arbeit

Diese Ausarbeitung präsentiert eine umfassende Untersuchung zur Entwicklung eines automatisierten Systems zur Extraktion relevanter Inhalte aus *Bedarfsmeldungen* im Software-Entwicklungs-Kontext.

- In der Ausarbeitung wird zunächst ein Konzept einer standardisierten *Bedarfsmeldung* erarbeitet. Dazu wird eine klare Erwartungshaltung hinsichtlich der Anforderungen und Bedürfnisse der Stakeholder entwickeln. Hierfür werden Interviews mit Führungskräften durchgeführt, um die Erwartungen bezüglich einer „perfekten“ *Bedarfsmeldung* herauszuarbeiten. Dieses Konzept dient als Grundlage für die weiteren Entwicklungs- und Evaluierungsphasen.
- Es wird an einer ausführbaren prototypischen Software gearbeitet, die *Bedarfsmeldungen* effizient verarbeitet und wichtige Informationen extrahiert. Hierfür wird eine Pipeline in Python aufgebaut und strukturell durch ein Use-Case- und UML-Aktivitätsdiagramme dokumentiert. Es werden Modelle des Information Retrieval und Data-Mining implementiert die dazu beitragen, eine *Bedarfsmeldung* in die vorher definierte Struktur umzuformen. Dabei erfolgt zunächst eine eingehende Analyse der Techniken *TF-IDF*, *N-Gramm*, *Named Entity Recognition*, *POS-Tagging*, und Hybride Ansätze, um die besten Ansätze zur Extraktion relevanter Inhalte zu identifizieren. Diese Analyse bildet die Grundlage für die Umsetzung des Software-Prototypen, das eine Kombination der erforschten Ergebnisse darstellt.
- Um die Leistungsfähigkeit des entwickelten Systems zu evaluieren, werden Testfälle mit synthetischen Daten definiert. Dabei wird überprüft, inwieweit das Ergebnis einer definierten Erwartungen entspricht. Mit Hilfe der Kosinus-Ähnlichkeit werden Abweichungen, Ähnlichkeiten und Anpassungen analysiert, um Erkenntnisse über die inhaltliche Leistung des Systems und die Tech-

niken zu gewinnen, die in Kombination mit mehreren Ansätzen die wichtigsten Informationen aus den semi-strukturierten *Bedarfsmeldungen* herausfiltern.

1.3. Aufbau der Arbeit

Kapitel 1 Einleitung befasst sich mit der Problemstellung und die Zielsetzung der Ausarbeitung. Zudem wird der Aufbau der Arbeit erläutert.

In **Kapitel 2 Strukturierte Bedarfsmeldung** werden Experteninterviews durchgeführt und analysiert, um eine Standardisierung von *Bedarfsmeldung* zu entwickeln. Dies legt die Grundlage, wohin das zu entwickelnde System gelangen soll.

Kapitel 3 Konzeptionierung befasst sich mit der Grundsätzlichen Idee eines Recommender Systems zur Mitarbeiterempfehlung und der Historie von Recommender Systems. Außerdem wird auf abstrakter Ebene das zu Entwickelnde System konzipiert und Anforderungen zusammengetragen.

Kapitel 4 Literaturüberblick wird Literatur zu Methodiken und Ansätze analysiert, die bei der Umsetzung des konzipierten Systems beitragen.

In **Kapitel 5 Umsetzung** werden Implementationsdetails des System zur Strukturierung von *Bedarfsmeldungen* dargestellt.

In **Kapitel 6 Evaluierung** wird das in Kapitel 5 entwickelte System mithilfe einer vordefinierten Erwartungshaltung evaluiert.

Das abschließende **Kapitel 7 Zusammenfassung und Ausblick** fasst die wichtigsten Ergebnisse zusammen und gibt einen Ausblick auf mögliche weiterführende Forschungen und Anpassungsmöglichkeiten des entwickelten Systems.

2. Strukturierte Bedarfsmeldung

In diesem Kapitel werden Methodiken und die Durchführung von Experteninterviews erörtert, mit dem Ziel, wichtige Aspekte einer *Bedarfsmeldung* durch eine qualitative Inhaltsanalyse zu identifizieren. Auf Basis dieser Erkenntnisse wird eine strukturierte Vorlage von *Bedarfsmeldungen* entwickelt, die die Anforderungen an das zu entwickelnde System zur Extraktion relevanter Informationen aus semistrukturierten *Bedarfsmeldungen* definiert.

2.1. Bedarfsmeldungen

Eine *Bedarfsmeldung* bezeichnet eine Projektbeschreibung, die Anforderungen an ein zu entwickelndes System enthält. Sie umfasst mehrere wesentliche Aspekte, die dafür sorgen klar und effektiv zu sein. Dazu gehört eine Überschrift, die die Projektart und den Bedarf kurz zusammenfasst. Die Beschreibung spiegelt den Bedarf wider und durch den Einsatzkontext können wichtige Hintergrundinformationen zum Umfeld des Projekts erklärt werden. Das Datum gibt Aufschluss über den Zeitplan, in welchem das Projekt durchgeführt werden soll. Durch das Volumen können benötigte Ressourcen und den Umfang des Bedarfs dokumentiert werden. Eine *Bedarfsmeldung* sollte den Tagessatz sowie den Einsatzbereich klar definieren. Zudem ist es wichtig, die Dauer des Einsatzes und den Technologie-Stack anzugeben, der zum Einsatz kommt. Darüber hinaus sollten die Muss- und Kann-Kriterien für die gesuchten Fachkräfte klar formuliert werden. Einarbeitungszeiträume und Lieferverpflichtungen sind ebenfalls entscheidende Bestandteile, die in einer *Bedarfsmeldung* nicht fehlen dürfen.

2.2. Experteninterviews

Im Rahmen der vorliegenden Ausarbeitung werden halbstrukturierte Interviews mit Experten aus dem Bereich des *Staffings* durchgeführt. Sinn und Zweck von Experteninterviews ist die Rekonstruktion spezifischer Wissensbestände oder besonders exklusiver, detaillierter oder umfassender Kenntnisse über bestimmte Wissensbestände und Praktiken. Der Begriff *Experte* bezeichnet hier eine Person, die über

einen privilegierten Zugang zu Informationen verfügt [26].

Die Experteninterviews in dieser Ausarbeitung zielen darauf ab, qualitative Daten zu erheben. Die Interviews werden als Einzelinterviews durchgeführt, wodurch der Fokus auf das spezifische Wissen jedes Befragten gerichtet werden kann. Die Ergebnisse der Interviews bilden die Grundlage für die Formulierung der Anforderungen einer optimalen *Bedarfsmeldung*, die als Basis für das zu entwickelnde System eingesetzt wird. Alle Interviews wurden in Microsoft Teams abgehalten. Die Experten sind einerseits vielfach unterwegs und nicht immer in der Geschäftsstelle anzufinden. Andererseits verfolgen sie einen strikten Zeitplan wodurch Meetings auch spontan verschoben werden. Zur Zeiteinsparnis und Spontanität wurden diese dadurch Online abgehalten. Jedes Interview wird zu Dokumentationszwecken aufgezeichnet. Um die Interviews strukturiert für die qualitative Inhaltsanalyse vorzubereiten, ist es erforderlich sie in schriftliche Transkripte umzuwandeln. Zur ersten Textumwandlung wurde das Transkriptionstool von Teams verwendet. Ungenaue Umwandlungen wurden mit der Videoaufnahme nachgebessert. Grundsätzlich wird die einfache Transkribierung nach Dresing und Pehl angewandt, da nur Wert auf den Inhalt der Interviews gelegt wird [10]. Im Rahmen dieses Schritts wird der Text vom Umgangssprachlichen in einen gut lesbaren Text ohne Lücken übersetzt [10]. Der Interviewer wurde mit einem „I“ und die jeweils befragte Person mit „B“ gekennzeichnet. Im Rahmen des Transkriptionsprozesses werden die Aussagen der Interviews anonymisiert. Personenbezogenen Informationen werden somit durch neutrale Bezeichnungen ersetzt.

Im Vorfeld der Durchführung der Interviews wird eine Überprüfung der inhaltlichen Verständlichkeit der Fragen sowie ihrer Beantwortbarkeit vorgenommen. Zudem wird Feedback zur Reihenfolge der Fragen eingeholt. Zu diesem Zweck sind die Fragen vorab an eine Führungsperson geschickt und schriftlich beantwortet worden. Die Fragen werden den Experten vorab inklusive Kontext des Interviews geschickt, damit diese sich bei Bedarf Gedanken machen können. Zur zeitlichen Begrenzung wird das Interview auf zehn Fragen reduziert. Trotz der vorgegebenen Strukturierung des Interviews wird Raum für spontane Fragen gelassen, um eine natürliche Gesprächsführung zu ermöglichen. Die Fragen dienen als Orientierungshilfe und Leitfaden durch das Interview. Der Leitfaden ist dabei lediglich als inhaltliche Richtlinie zu verstehen, von der gegebenenfalls abgewichen werden kann.

2.3. Ablauf der Interviews

Im Rahmen der Interviews erfolgt zunächst eine Einführung in die Problemstellung sowie das Ziel der Ausarbeitung. Auf diese Weise wird sichergestellt, dass die Interviewpartner den Sinn und Zweck des Interviews nachvollziehen können. Des Weiteren erfolgt eine definitorische Erläuterung des Begriffs *Bedarfsmeldung*. Auch wenn dies die fachliche Bezeichnung darstellt, ist sie nicht jedem einzelnen Experten geläufig. Im Rahmen der Einführung erfolgt zunächst eine Erörterung der Rolle des Experten bei *adesso*. Im Rahmen dessen erfolgt eine Klärung der genauen Aufgaben des Experten bei *adesso*. Im Anschluss erfolgt eine Erörterung der Rolle des Experten im Kontext des *Staffing*-Prozesses sowie der Bearbeitung der *Bedarfsmeldung*. Die Interviewfragen sind so konzipiert, dass sie zunächst allgemein gehalten sind und im Verlauf des Interviews zunehmend präziser werden. Im Anschluss an die Erörterung der Frage, welche Art von Projekten über *Bedarfsmeldungen* erfasst wird, wird die Frage aufgeworfen, auf welche Weise diese kommuniziert und dokumentiert werden. Im Anschluss erfolgt eine Klärung der für die Erfassung von *Bedarfsmeldungen* relevanten Informationen. Dabei wird erörtert, welche Informationen von besonderer Bedeutung sind und folglich nicht fehlen dürfen. Die Herausforderungen und Bewertungskriterien geben Aufschluss über die bereits genutzten Ansätze zur Standardisierung der *Bedarfsmeldungen*. Des Weiteren können bereits unternommene Maßnahmen zur Qualitätssicherung als hilfreicher Ansatz zur Identifizierung von Anforderungen der *Bedarfsmeldungen* herangezogen werden. Die Beantwortung der Fragen erfordert insbesondere die Entwicklung eigener Ideen und Konzepte, die im weiteren Verlauf des Prozesses zur Erfassung von *Bedarfsmeldung* gegebenenfalls noch nicht zum Einsatz gekommen sind.

2.4. Übersicht der Experten

Die Übersicht der interviewten Experten ist in der Tabelle 2.1 aufgeführt. Die erste Spalte der Tabelle 2.1 zeigt die vergebene Nummerierung sowie das Datum der Durchführung. Dies dient der vereinfachten Referenzierung innerhalb der Ausarbeitung. Die zweite Spalte der Tabelle enthält die Rolle bzw. Tätigkeit der Befragten bei *adesso*. Alle Befragten haben, oder hatten eine leitende Rolle mit Erfahrungen in der Personaleinsatzplanung. Dementsprechend hat jeder Befragte in irgend einer Form Berührungspunkte mit *Bedarfsmeldungen* gehabt. Interview 2 und 3 haben zusätzlich noch die Perspektive zur Erstellung und Verwaltung von *Bedarfsmeldungen*, da ihre Hauptaufgaben genau in diesem Bereich fallen.

Nr. und Datum	Profil der Befragten	Durchführungsart	Dauer	Transkripte (Anhang)
1. 28.03.2024	CC-Leiter und Softwarearchitekt	schriftlicher Vorabtest	-	A.1.1
2. 29.04.2024	CC-Leiter und Projektleiter	Video-Interview	15min	A.1.2
3. 29.04.2024	CC-Leiter und Delivery Manager	Video-Interview	45min	A.1.3
4. 30.04.2024	CC-Leiter und Softwarearchitekt	Video-Interview	15min	A.1.4
5. 6.05.2024	Bereichsleiter	Video-Interview	25min	A.1.5

Tabelle 2.1.: Übersicht der Experten

Die 3. Spalte zeigt die Art der Befragung. Abgesehen von dem Vorabtest wurden alle Interviews in einem Video-Call abgehalten. Die vierte Spalte zeigt die Dauer der Interviews. Die letzte Spalte ist eine Referenzierung zu den Transkripten der Interviews im Anhang. Die befragten Personen wurden so ausgewählt, dass diese durch unterschiedliche Aufgabenbereiche differenzierte Bezüge zu *Bedarfsmeldungen* pflegen. Alle befragten Personen hatten zu einem Zeitpunkt in ihrer Laufbahn die Rolle einer Führungsperson übernommen und somit Berührungspunkte mit *Bedarfsmeldungen* zur Projektfindung für die eigenen Mitarbeiter hatten. Darüber hinaus hat jeder Experte durch die weiteren Aufgaben innerhalb von *adesso* eine andere Sicht auf die *Bedarfsmeldungen*. Softwarearchitekten legen potenziell mehr Wert auf technischere Aspekte wie benötigte Expertise in Technologien. Ein Delivery Manager, der die Schnittstelle zwischen Unternehmen und Kunde darstellt versucht wiederum wichtige Aspekte aus beiden Sichten zusammenzuführen. Ein Projektleiter hat potenziell wiederum ein anderen Anspruch an eine *Bedarfsmeldung*, da dieser seine eigenen Vorstellungen und Ansprüche an das Projekt durchbringen möchte. Wie sich das genau verhält, wird erst durch die Interviews deutlicher.

Die **erste** Befragte Person aus dem schriftlichen Vorabtest ist studierter Informatiker mit Diplomabschluss. Bei *adesso* nimmt er die Rolle des CC-Leiters ein. Dies

ist die Bezeichnung für Führungspersonen mit Zuständigkeiten für Mitarbeiter bei *adesso*. CC-Leiter haben mit Bedarfsmeldungen zu tun, da sie ihre eigenen Mitarbeiter auf Projekte einstellen müssen. Neben der Tätigkeit als Führungsperson ist er Softwarearchitekt und leitet teilweise Projekte.

Die **zweite** befragte Person hat eine Ausbildung zum Fachinformatiker für Anwendungsentwicklung in einem Softwarehaus abgeschlossen. Nach 7 Jahren Erfahrungen als Consultant wurde er innerhalb von 12 Jahren vom Softwareentwickler zum IT-Leiter einer mittelständigen Bank. Nun ist er als CC-Leiter und Projektleiter bei *adesso* tätig.

Die **dritte** befragte Person war während seiner Laufbahn bei *adesso* als CC-Leiter tätig. Aktuell ist er in einer Mischform von Projektleitung und Produktunterschiede um interne Projekte. Zudem übernimmt er auch noch die Rolle des Delivery Managers, bei dem er eine Schnittstellenposition zwischen *adesso* und den Kunden einnimmt. Dabei geht es unter anderem auch um die Verwaltung der *Bedarfsmeldungen*.

Die **vierte** Befragte Person ist Studierter Kerninformatiker mit einem Diplomabschluss. 2003 startete die Karriere als Software Entwickler und seitdem ist er Software Architekt und CC-Leiter bei *adesso*.

Die **fünfte** Person ist promovierter Informatiker mit einem Diplomabschluss. Mit Umwegen ist er im Bereich der Dienstleistung gewechselt. Bei *adesso* ist er CC-Leiter und Bereichsleiter. Er übernimmt das Kunden Management und ist für die Gemeinsame Gestaltung der Zusammenarbeit mit Kunden, aber nicht so sehr für einzelne Bedarfsmeldungen zuständig.

2.5. Auswertungsmethode

Zur systematischen Analyse der Experteninterviews wurde die qualitative Inhaltsanalyse nach Philipp Mayring angewendet. Das Ziel der Inhaltsanalyse in dieser Ausarbeitung ist die Zusammenfassung der Informationen aus den Interviews. Grundsätzlich unterscheidet sich die Inhaltsanalyse in drei Grundtechniken: (i) *Zusammenfassung* (Materialreduktion, um die wesentlichen Inhalte zu erhalten), (ii) *Explikation* (Verständniserweiterung durch Herantragung von Zusatzmaterial) und (iii) *Strukturierung* (Herausfilterung von Aspekten und Einschätzung von Kriterien) [20].

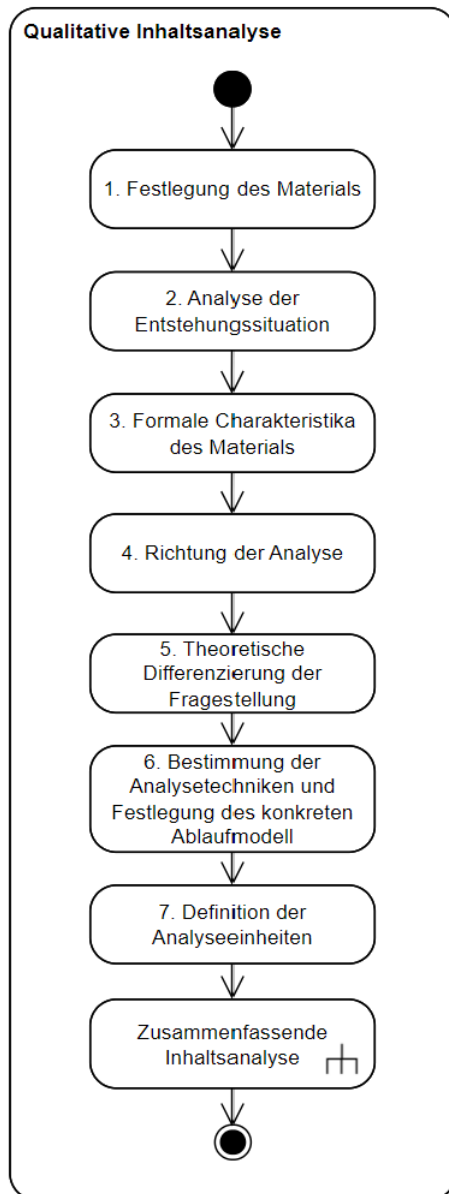


Abbildung 2.1.: UML-Aktivitätsdiagramm zum Ablaufmodell der qualitativen Inhaltsanalyse nach Mayring [22]

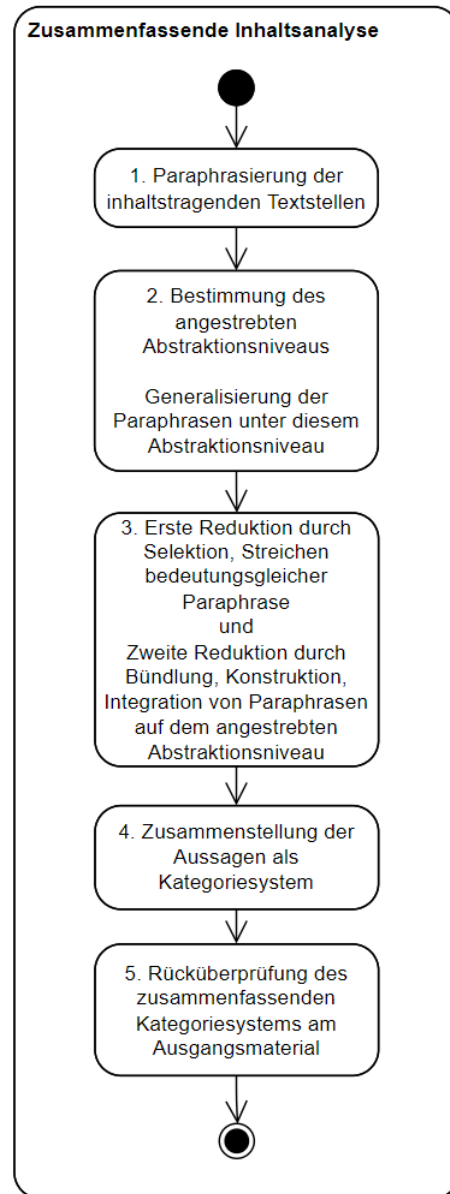


Abbildung 2.2.: UML-Aktivitätsdiagramm zum Ablaufmodell der zusammenfassenden Inhaltsanalyse nach Mayring [22]

Im Folgenden wird sich auf die Technik der Zusammenfassung konzentriert, da die Anforderungen an *Bedarfsmeldungen* nur durch eine Übersicht der Interviewergebnisse erstellt werden kann. Zur strukturierten Analyse werden die Arbeitsschritte mit Hilfe des Ablaufmodells von Mayring aufgeteilt [22]. Mayring beschreibt

ein allgemeines Ablaufmodell mit Schritten für die qualitative Inhaltsanalyse [22]. Das UML-Aktivitätsdiagramm aus der Abbildung 2.1 beschreibt diesen Ablauf bis zum Schritt *Analyseschritte mittels Categoriesystem*. In diesem Schritt wird sich auf eine konkrete Analyseform festgelegt. Diese Analyse durchläuft die zusammenfassende Inhaltsanalyse nach Mayring [22]. Dadurch wird der Ablauf spezifischer und kann somit nicht mehr mit dem allgemeinen Ablaufmodell dargestellt werden. Die Schritte der zusammenfassenden Inhaltsanalyse nach Mayring werden im UML-Aktivitätsdiagramm in Abbildung 2.2 dargestellt.

2.6. Durchführung der qualitativen Inhaltsanalyse

Die Struktur aus den Abbildungen 2.1 und 2.2 wird nachfolgend Schrittweise auf die Interviews angewendet.

1. Festlegung des Materials

Die Materialien sind aus 4 Experteninterviews und einem schriftlichen Vorabtest entnommen worden. Dazu wurden 10 Interviewfragen als Leitfaden angewendet.

2. Analyse der Entstehungssituation

Die Interviewteilnehmer sind Führungskräfte, die bereits mit *Bedarfsmeldungen* Berührungspunkte hatten. Vereinzelt haben diese auch näheren Kontakt mit *Bedarfsmeldungen*. Dies ist in Kapitel 2.4 näher erläutert.

3. Formale Charakteristika des Materials

Die Interviews wurden über Microsoft Teams mit der Aufnahmefunktion aufgezeichnet und mit der Teams Transkriptionsfunktion transkribiert. Undeutliche Transkriptionsbereiche wurden mit der Aufnahme nachgebessert. Die Interviews liegen in Textform im Anhang zur Verfügung.

4. Richtung der Analyse

Das Ziel der Analyse ist die Informationsgewinnung aus den Interviews, um die wichtigsten Aspekte von *Bedarfsmeldungen* zu identifizieren. Dabei soll der Fokus auf den Inhalt gelegt werden, wodurch emotionale und sprachliche Faktoren nicht einbezogen werden.

5. Theoretische Differenzierung der Fragestellung

Die Interviews sind auf die Aussagen der Experten aus Teilbereichen der Fachkräf-

teorganisation und der Akquirierung und Bearbeitung von Projekten und Projektbedarfen zugeschnitten. Es geht um den Informationsgehalt von *Bedarfsmeldungen*. Dafür wurden Themenfelder festgelegt, die Aussagen über die Relevanz und Konsistenz der *Bedarfsmeldungen* behandeln. Der Ablauf und die dazugehörigen Themenfelder wurden bereits in Kapitel 2.3 erläutert und spiegeln sich in den Fragen wider.

6. Bestimmung der Analysetechnik und Festlegung des konkreten Ablaufmodells

Zur Extraktion der relevanten Informationen wurde die zusammenfassende Inhaltsanalyse nach Mayring verwendet. Innerhalb der Analyse wird der Text auf wesentliche Inhalte reduziert und in Kategorien unterteilt. Die einzelnen Schritte des Ablaufmodells für eine zusammenfassende Inhaltsanalyse ist in Abbildung 2.2 abgebildet.

7. Bestimmung der Analyseeinheiten

Mayring beschreibt drei Analyseeinheiten. Die (i) *Auswertungseinheit* definiert, welche Textteile jeweils nacheinander kodiert werden [20]. Die (ii) *Kodiereinheit* definiert, welche minimale Materialmenge ausgewertet werden darf und welcher Kategorie sie zugeordnet werden kann [20]. Die (iii) *Kontexteinheit* definiert die maximale Textmenge, die unter eine Kategorie fallen kann [20].

Die Auswertungseinheit umfasst alle Interviewtranskripte. Die kleinste Textmenge, die kodiert wird betragen einzelne Wörter aus einzelnen Absätzen. Die Kontexteinheit bezieht sich auf die jeweiligen Absätze zu den einzelnen Fragen aus den Interviews.

Zusammenfassende Inhaltsanalyse

Die Interviews werden auf die wichtigsten Informationen heruntergebrochen und mit Berücksichtigung der Verfahrensregeln nach Mayring paraphrasiert, generalisiert, reduziert und als Categoriesystem zusammengetragen [20]. Die Verfahrensregeln beschreiben Arbeitsschritte bei der Durchführung der zusammenfassenden Inhaltsanalyse [20]. Die Ergebnisse der Schritte aus dem Ablaufmodell der Abbildung 2.2 sind im Anhang A.2 in der Tabelle A.1 dargestellt. Diese Tabelle beinhaltet in der ersten Spalte den *Fall*, der das jeweilige Interview widerspiegelt. Fall 1 entspricht Interview 1, usw. Die zweite Spalte ist die *Nummer* der jeweiligen Zeile der Tabelle, bei der paraphrasiert, generalisiert, reduziert wurde. Diese ist dazu da um inhaltlich

zusammengehörende Abschnitte referenzieren zu können. Die dritte Spalte enthält die Paraphrasierung. Die vierte Spalte umfasst die Generalisierung und die fünfte Spalte enthält die Kategorien mit den reduzierten Informationen. Die Paraphrasierung, Generalisierung, sowie erste und zweite Reduktion wurden in einem Schritt durchgeführt. Dies kann bei größeren Datenmengen getan werden [22].

1. Paraphrasierung der inhaltstragenden Textstellen

Zu Beginn wurden die Interviews paraphrasiert. Dabei wurden die Verfahrensregeln für die Paraphrasierung nach Mayring beachtet [22]. Dabei wurden alle nicht oder wenig inhaltstragende Textbestandteile wie ausschmückende, wiederholende oder verdeutlichende Wendungen aus den Interviews gestrichen [22]. Außerdem wurden die inhaltstragenden Textstellen auf eine einheitliche Sprachebene gebracht und auf eine grammatikalische Kurzform transformiert [22]. Das Abstraktionsniveau ist dabei hoch. Die Aussagen wurden sinngemäß aus den Interviews extrahiert.

2. Generalisierung der Paraphrasen

Bei der Generalisierung wurden die Verfahrensregeln für die Generalisierung auf das Abstraktionsniveau nach Mayring angewendet [22]. Dabei wurden die Gegenstände der Paraphrasen auf die definierte Abstraktionsebene generalisiert [22]. Alle Satzaussagen wurden auf die gleiche Weise generalisiert [22]. Paraphrasen, die über dem angestrebten Abstraktionsniveau liegen wurden belassen [22]. Bei Zweifelsfällen wurden theoretische Vorannahmen zu Hilfe genommen [22].

3. Reduktion

Die Verfahrensregeln für die erste und zweite Reduktion wurden bei der Durchführung beachtet [22]. Bei der ersten Reduktion wurden bedeutungsgleiche Paraphrasen innerhalb der Auswertungseinheiten gestrichen [22]. Zudem wurden Paraphrasen gestrichen, die nicht wesentlich inhaltstragend sind [22]. Paraphrasen die weiterhin inhaltstragend sind wurden übernommen [22]. Die zweite Reduktion umfasst die Zusammenfassung von Paraphrasen mit ähnlichem Gegenstand oder mehreren Aussagen [22]. Zudem wurden Paraphrasen mit gleichem Gegenstand und verschiedener Aussagen zusammengefasst [22].

4. Zusammenstellung der Aussagen als Categoriesystem

Die Kategorien sind induktiv aus dem Material erstellt worden [21]. Dabei wurden alle Interviews heruntergebrochen, paraphrasiert, generalisiert, reduziert und in

Kategorien (K1-K7) überführt. Bei der Durchführung der Schritte 1-3 der zusammenfassenden Inhaltsanalyse haben sich Themenschwerpunkte durch die Interviewfragen entwickelt. Aussagen konnten zusammengefasst werden, wodurch sich daraus die Kategorien gebildet haben. In jeder Kategorie sind Stichpunktartig die Informationen aus den Paraphrasen enthalten.

Nr.	Überschrift	Stichpunkte
K1	Arten von Projekten	<ul style="list-style-type: none"> - Kundenprojekte - Softwareentwicklungsprojekte - Time Material - Festpreis
K2	Stakeholder	<ul style="list-style-type: none"> - Maitre - Führungskräftenetzwerk - Sales - Projektleiter - Fachverantwortliche - Entscheider - Geschäftsführer - Delivery Manager - Account Manager
K3	Wichtige Informationen	<ul style="list-style-type: none"> - Tagessatz - Einsatz - Dauer - Tech Stack - Muss/Kann Kriterien - Einarbeitungszeiträume - Lieferverpflichtung

K4	Bedarfsmeldung	<ul style="list-style-type: none"> - Überschrift - Beschreibung - Einsatzkontext - Datum - Volumen - in Jira gespeichert - gewichtete Fähigkeiten - keine feste Struktur
K5	Qualitätsbewertung	<ul style="list-style-type: none"> - keine vorhanden - Erfahrung - über Projektleitung - intensives Lesen - Rückfragen stellen - grober Rahmen durch Jira - regelmäßige Meetings
K6	Qualitätsverbesserung	<ul style="list-style-type: none"> - klare Vorgaben - weniger Freitext - Reviewprozess - Verständnisübereinstimmung - Beseitigung Missverständnisse - strukturierte Datenerfassung - KI-gestützte Prüfungen
K7	Auswirkungen	<ul style="list-style-type: none"> - Überblick verlieren - längerer Staffing-Prozess - Umbesetzung - erhöhter Aufwand - Mehrfachbeantwortung - Missverständnis - unpassendes Personal

Tabelle 2.2.: Ergebnisse der qualitativen Inhaltsanalyse.

In Tabelle 2.2 sind die Ergebnisse der qualitativen Inhaltsanalyse als Kategorien abgebildet. Die Spalte eins beinhaltet die Kategoriennummer (K1-K7) der jeweiligen Kategorie. Die zweite Spalte enthält die Überschrift der Kategorien. Die dritte Spalte umfasst die stichpunktartigen Inhalte und Aussagen aus der zusammenfassenden Inhaltsanalyse.

5. Rücküberprüfung des Categoriesystems am Ausgangsmaterial

In diesem Schritt wurde das Ausgangsmaterial ein weiteres Mal durchgesehen. Dadurch wurde sichergestellt, dass alle relevanten Inhalte durch das Categoriesystem abgedeckt werden. Zudem wurden Unklarheiten und Lücken identifiziert und ausgebessert.

2.7. Interpretation der Ergebnisse

Die Kategorien zeigen, dass *Bedarfsmeldungen* Bestandteil eines komplexen und vielschichtigen Prozesses sind, der verschiedene Arten von Projekten, eine Vielzahl von Stakeholdern und detaillierte Informationen umfasst.

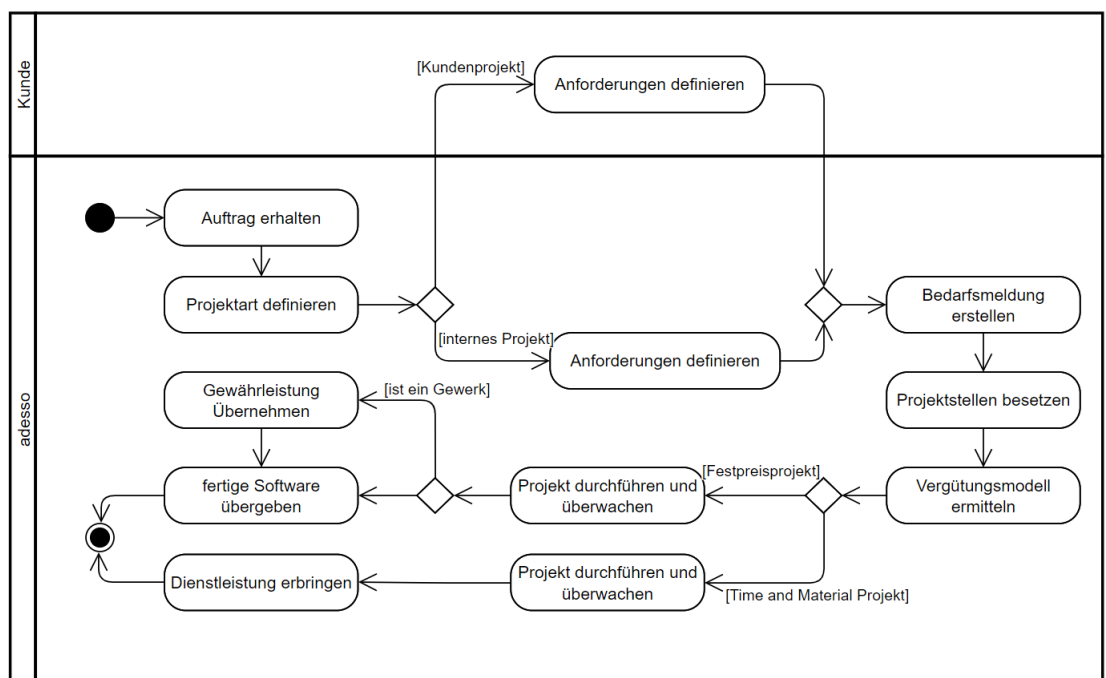


Abbildung 2.3.: Vereinfachte Darstellung zur Erfassung und Durchführung eines Projektes bei *adesso*.

Die Abbildung 2.3 zeigt eine vereinfachte Darstellung des Prozesses zur Erfassung und Durchführung von Projekten bei *adesso*. Dieser Prozess ist in der Hinsicht vereinfacht, dass dieser im Detail von Projekt zu Projekt unterschiedlich sein kann.

Grundsätzlich wird nach dem Erhalt eines Auftrages zwischen einem Kunden- oder internen Projekt unterschieden. Dabei wird differenziert, ob die Anforderungen vom Kunden oder intern definiert werden. Aus den Anforderungen wird die *Bedarfsmeldung* erstellt. Anschließend wird auf Basis dieser *Bedarfsmeldung* passende Mitarbeiter manuell identifiziert. Das Vergütungsmodell unterscheidet sich zwischen einem Festpreis- und Time and Material Projekt. Das Projekt wird mit den ermittelten Mitarbeitern durchgeführt und überwacht. Entsprechend des Vergütungsmodells und ob es ein Gewerk ist, wird entweder eine fertige Software oder eine Dienstleistung erbracht. Dieser Prozess ist im Detail vielschichtig und komplex. Viele Schritte können und dürfen nicht von Software übernommen werden. Dennoch kann der Prozess zur Besetzung der Projekte durch Systeme unterstützt werden.

Die Qualität der *Bedarfsmeldungen* ist entscheidend für den Erfolg der Projekte, und es existieren Möglichkeiten zur Verbesserung durch Strukturierung, Automatisierung und klarer Kommunikationswege. Die Auswirkungen von unzureichender Pflege und fehlendem Informationsgehalt der *Bedarfsmeldungen* sind weitreichend und können zu erheblicher Ineffizienz und Problemen innerhalb von *adesso* führen.

Aktuelle Struktur einer Bedarfsmeldung

Die *Bedarfsmeldungen* werden in Jira gespeichert und erhalten dadurch einen groben Rahmen, wie diese gepflegt werden.

Solution Architect für das Entwickeln einer serverless Webapplication

Status:

Labels:

Rolle:

Aufgaben:

Skills:

Skill-Level:

Kunde:

Einsatzort:

Beginn Einsatz:

Ende Einsatz:

Option auf Verlängerung:

Tagessatz:

Reisekosten vergütet:

ANÜ:

Freelancer:

Smartshore-fähig:

EU-Shoring:

Deutschsprachig:

Bemerkung/Sonstiges:

Abbildung 2.4.: Mockup des Aufbaus einer *Bedarfsmeldung* in Jira.

Diese Struktur ist in Abbildung 2.4 abgebildet. Enthalten sind vordefinierte Felder zur Beschreibung von *Bedarfsmeldungen*. Der *Status* beschreibt in welchem Bearbeitungslage das Projekt zur *Bedarfsmeldung* ist. Dabei wird zwischen *Offen*, *Eskaliert* und *Geschlossen* unterschieden. Dadurch wird signalisiert in welcher Form die *Bedarfsmeldung* Aufmerksamkeit benötigt. Die *Labels* helfen bei der Suche nach den *Bedarfsmeldungen* und geben eine abstrakte Übersicht des Themengebiets. Die *Rolle* stellt die Anforderungen bezüglich der Verantwortung an die gesuchten Mitarbeiter. Die Felder *Aufgaben* und *Skills* beinhalten konkrete Anforderungen an die Mitarbeiter in Form von Kriterien und Fähigkeiten. Diese bestehen aus unstrukturiertem Informationsgehalt. Die Informationen sind insofern unstrukturiert, als dass sie ohne vordefinierte Struktur in Form eines Volltexts vorliegen. Infolgedessen kann es zu Abweichungen hinsichtlich der Ausgestaltung von *Bedarfsmeldungen* kommen. Das *Skill-Level* gibt Auskunft über die benötigte Kompetenzstufe in der gesuchten Rolle.

Das Feld *Kunde* enthält Informationen zum Auftraggeber der *Bedarfsmeldungen*. Die Felder *Einsatzort*, *Beginn Einsatz* und *Ende Einsatz* beinhalten Zeitliche und ortsspezifische Informationen, in welchem die *Bedarfsmeldung* durchgeführt wird. Der *Tagessatz* gibt Auskunft über die Vergütung des Mitarbeiters. Des weiteren existieren spezifischere Informationen wie *Option auf Verlängerung*, *Reisekosten vergütet*, *ANÜ*, *Freelancer*, *Smartshore-fähig*, *EU-Shoring*, *Deutschsprachig* und *Bemerkung/-Sonstiges* die individuell spezifisch zu bestimmten *Bedarfsmeldungen* ausgefüllt werden. Diese Felder werden dennoch nicht regelmäßig ausgefüllt, wodurch diese oftmals leer bleiben. Zudem gelten diese für den Staffing nicht als relevante Felder.

2.8. Strukturierung von Bedarfsmeldungen

Die Problemstellung umfasst eine Reihe von Punkten, die im Rahmen der Ausarbeitung zu behandeln sind. Aufgrund der unstrukturierten und mit fehlenden Informationen versehenen *Bedarfsmeldungen* ist eine Strukturierung von besonderer Relevanz. Dies würde die Extraktion relevanter Informationen erleichtern und somit die Effizienz des Systems verbessern. Auf Basis der Informationen aus den Interviews und der Kategorien im Kapitel 2.6 wurde die Semi-Strukturierung aus der Abbildung 2.4 angepasst, reduziert und konkretisiert.

Solution Architect für das Entwickeln einer serverless Webapplication

Einsatz:	13.06.2024 - 20.10.2024
Rolle:	Solution Architect
Skill-Level:	Senior
Tagessatz:	1020 EUR
Kunde:	Sample GmbH
Einsatzort:	Remote
Aufgaben:	-fundiertes Know How -Webtechnologien -serverless Webapplications ~ ...
Skills:	-HTML -CSS -JavaScript -Angular -AWS Lambda -AWS Amplify ~...

Abbildung 2.5.: Mockup einer strukturierten *Bedarfsmeldung*.

In Abbildung 2.5 ist die Struktur dargestellt, in die die Bedarfsmeldungen durch das zu entwickelnde System überführt werden sollen. Der Einsatzbeginn und das Einsatzenende werden in einem Feld zusammengefasst. Des Weiteren werden die Aufgaben und Skills nicht als Freitext, sondern als stichpunkthaltige Zusammenfassungen reduziert. Fachexperten benötigen laut den Kategorien aus Kapitel 2.6 prägnante Listen mit gekürzten Informationen wie benötigte Expertise in Technologien und Muss- und Kann-Kriterien. Mitarbeiterprofile sind abstrakt gesehen auch Listen mit Skills. Dadurch soll die *Bedarfsmeldung* in eine vergleichbare Struktur überführt werden. Im Rahmen der Auswertung der Interviews wurden diejenigen Felder ermittelt, die als besonders Relevant deklariert sind. Neben diesen Feldern wurden alle weiteren irrelevanten und potenziell leeren Felder entfernt.

Im Rahmen der Entwicklung einer Software zur automatisierten Strukturierung einer *Bedarfsmeldung* ist die Extraktion der erforderlichen Informationen aus den Volltexten erforderlich. In diesem Kontext existiert bereits eine Reihe an Methoden und Ansätzen, die sich in der Forschung bewährt haben. Die unstrukturierten Volltexte müssen annäherungsweise in eine strukturierte inhaltliche Aufteilung in einzelne Sequenzen und Stichpunkte überführt werden.

3. Konzeptionierung

In diesem Kapitel wird auf Basis der Ergebnisse aus Kapitel 2 ein System konzipiert, das als Teilsystem des Recommender System zur Mitarbeiterempfehlung dienen soll.

3.1. Idee des Recommender System zur Mitarbeiterempfehlung

Das zu unterstützende Recommender System zur Mitarbeiterempfehlung besteht aus mehreren Schritten.

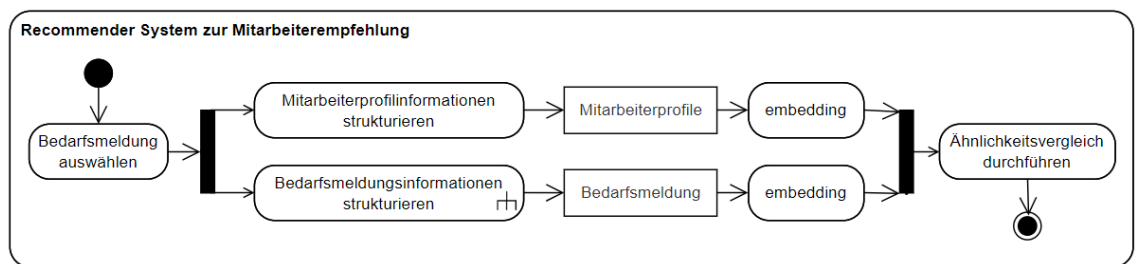


Abbildung 3.1.: Abstrakte Darstellung des Recommender System zur Mitarbeiterempfehlung.

Diese Schritte sind abstrakt als UML-Aktivitätsdiagramm in der Abbildung 3.1 dargestellt. Nach Auswahl einer *Bedarfsmeldung* wird diese einerseits auf die relevanten Informationen reduziert und strukturiert. Diesen Schritt gilt es in dieser Arbeit zu konzipieren, entwickeln und evaluieren. Neben diesem Schritt erfolgt eine Vorverarbeitung der Mitarbeiterprofile, bei dem die verfügbaren Informationen in eine vergleichbare Struktur wie die Bedarfsmeldung reduziert werden. Die strukturierte *Bedarfsmeldung* und Mitarbeiterprofile werden vektorisiert und am Ende mit einer Vergleichsmetrik auf Ähnlichkeit geprüft. Die Komplexität der Vergleichsmetrik ist dabei skalierbar. So können verschiedene Aspekte, wie z.B. Verfügbarkeit, Skills, etc., Bestandteil des Ähnlichkeits-Scoring sein. Das Profil, bzw. die Profile mit der höchsten Ähnlichkeit zur *Bedarfsmeldung* wird am Ende Empfohlen.

Recommender Systems Historie und aktueller Stand der Forschung

Auch wenn die Erstellung eines vollständigen Recommender Systems nicht Gegen-

stand der vorliegenden Ausarbeitung ist, stellt die Nutzung von Information Retrieval und Filtering ein entscheidender Schritt in Richtung eines funktionierenden Recommender Systems dar. Das Verständnis der Funktionsweise eines Recommender Systems sowie dessen Entwicklung in den vergangenen Jahren ist daher für das Verständnis des Teilbereichs dieser Thematik essentiell.

Recommender Systems existieren bereits seit vielen Jahren [9]. Im Jahr 1992 führten Belkin und Croft eine Analyse und einen Vergleich des Information Retrievals und Filtering durch [9]. Das Information Retrieval behandelt die grundlegende Technologie der Suchmaschine [9]. Das Recommender System basiert hauptsächlich auf der Technologie des Information Filtering. Im selben Jahr präsentierte Goldberg das Tapestry-System, das das erste System zur Informationsfilterung darstellt, das auf kollaboratives Filtern durch menschliche Bewertung basiert [9]. Die Mehrheit der frühen Empfehlungsmodelle basiert auf kollaborativer Empfehlungen, wobei K-Nearest-Neighbor (KNN)-Modelle eine besondere Rolle einnehmen. Diese Modelle prognostizieren die Nachbarn eines Zielnutzers, indem sie eine Ähnlichkeit zwischen den vorherigen Präferenzen und den Präferenzen der anderen Nutzer berechnen [9]. Die Studie von Goldberg inspirierte einige Forscher des Massachusetts Institute of Technology (MIT) und der University of Minnesota (UMN) dazu, einen Nachrichtempfehlungsdienst mit dem Namen *GroupLens* zu entwickeln. Die Hauptkomponente dieses Dienstes ist ein Modell zur kollaborativen Filterung zwischen Nutzern [9]. Das gleichnamige Forschungslabor kann somit als Pionier auf dem Gebiet der Recommender Systems bezeichnet werden. Die dort durchgeführten Forschungen bilden die Grundlage für nachfolgende Musik- und Video-Ähnlichkeitsempfehlungen [9].

Recommender Systeme haben in den letzten Jahren verschiedene Definitionen erhalten. Eine dieser Definitionen wird in dem Artikel von Resnick und Varian (1997) sinngemäß so beschrieben, dass ein typisches Recommender System Empfehlungen durch Personen als Eingabe erhält, die das System dann zusammenschließt und an geeignete Empfänger weiterleitet [4]. In einigen Fällen besteht die primäre Transformation in der Zusammenführung, in anderen Fällen liegt die Fähigkeit des Systems darin, gute Übereinstimmungen zwischen Empfehlungsgebern und Empfehlungsempfängern herzustellen [4]. Empfehlungssysteme stellen ein Instrument zur Interaktion mit umfangreichen und vielschichtigen Informationen dar [4]. Sie ermöglichen eine personalisierte Sicht auf diese Informationen, indem sie die für den Nutzer wahrscheinlich relevanten Inhalte aufbereiten [4]. Besonders im Handels-

verkehr im Internet sind Recommender Systeme ein häufiger Einsatzgebiet. Dabei werden Recommender Systeme als Werkzeuge zum Suchen und Filtern von Informationen verwendet, die dem Benutzer Vorschläge unterbreiten, die für ihn nützlich sein könnten [4]. Sie sind in einer Vielzahl von Internetanwendungen weit verbreitet und helfen den Nutzern, bessere Entscheidungen bei der Suche nach Nachrichten, Musik, Urlaubsangeboten oder Geldanlagen zu treffen [30]. Ein spezifisches Recommender System konzentriert sich typischerweise auf eine Art von Themengebiet wie z. B. Filme oder Nachrichten [30]. Darüber hinaus sind sie zu einem entscheidenden Faktor in der Entscheidungsfindung von Organisationen geworden [5]. Unternehmen wie *adesso* bauen immer weiter auf Recommender System unterstützte System auf, um Prozesse zu beschleunigen oder zu vereinfachen. Grundsätzlich können die Methoden in die Typen (i) *collaborative Filtering-based* (kollaborative Empfehlungssysteme), (ii) *content-based* (inhaltsbasierte Empfehlungssysteme), (iii) *knowledge-based* (wissensbasierte Empfehlungssysteme) und (iv) *hybrid* (hybride Empfehlungssysteme) unterteilt werden.

Jede Empfehlungsmethode hat ihre Vorteile und Grenzen [17]. Insbesondere das inhaltsbasierte Empfehlungssystem bringt eine hohe Relevanz für das Mitarbeiterempfehlungssystem. Die Grundprinzipien inhaltsbasierter Empfehlungssysteme sind zum einen die Analyse der Beschreibung der von einem bestimmten Benutzer bevorzugten *Items*, um die gemeinsamen Hauptattribute (Präferenzen) zu identifizieren, die diese *Items* unterscheiden. Diese Präferenzen werden in einem *Benutzerprofil* gespeichert [17]. Zusätzlich werden die Eigenschaften jedes *Items* mit dem *Benutzerprofil* verglichen, so dass nur *Items* empfohlen werden, die eine hohe Ähnlichkeit mit dem *Benutzerprofil* aufweisen [17]. Bei der Idee der Mitarbeiterempfehlung kann also die *Bedarfsmeldung* mit den benötigten Projektskills und Anforderung als *Benutzerprofil* angesehen werden. Die Mitarbeiterprofile sind dabei die *Items*. Die Attribute werden verglichen (Skills der Mitarbeiter mit den Skills und Anforderungen der *Bedarfsmeldung*) und ähnliche *Items* werden vorgeschlagen. Mit Hilfe traditioneller Methoden des Information Retrievals, wie z.B. dem Kosinus-Ähnlichkeitsmaß, werden dann Empfehlungen generiert [17]. Darüber hinaus generieren sie Empfehlungen mit Hilfe von statistischen und maschinelle Lernverfahren, die in der Lage sind, Nutzerinteressen aus historischen Nutzerdaten zu lernen [17].

Information Retrieval und Information Filtering

Im Allgemeinen wird einem Informationssystem die Funktion zugeschrieben, den Benutzer zu den Dokumenten zu führen, die seinen Informationsbedarf am besten

decken [3]. Allgemeiner ausgedrückt ist das Ziel eines Informationssystems, dem Benutzer Informationen aus der Wissensressource zur Verfügung zu stellen, die ihm helfen, ein Problem zu lösen [3]. Auf der anderen Seite ist unter Filtern das Entfernen von Daten aus einem eingehenden Datenstrom zu verstehen und nicht das Auffinden von Daten in diesem Datenstrom [3]. Filtersysteme verarbeiten große Datenmengen [3]. Typische Anwendungen betreffen Gigabytes von Text oder weitaus größere Mengen anderer Medien [3]. Während es bei dem Information Retrieval typischerweise um die einmalige Nutzung des Systems durch eine Person mit einem einmaligen Ziel und einer einmaligen Anfrage geht, befasst sich die Informationsfilterung mit der wiederholten Nutzung des Systems durch eine oder mehrere Personen mit langfristigen Zielen oder Interessen [3].

3.2. Anforderungsanalyse

Wie in Kapitel 3.1 erklärt funktionieren Recommender Systeme so, dass Attribute miteinander Verglichen werden. Dementsprechend ist es notwendig die *Bedarfsmeldungen* in eine Struktur zu bringen, bei dem einzelne Attribute für sich genommen strukturiert werden. Es wird eine Lösung gesucht, die eine effiziente Verarbeitung von *Bedarfsmeldungen* durchführen kann. Welche Aspekte in einer *Bedarfsmeldung* relevant sind, wurde bereits im Kapitel 2 näher erläutert. Die Idee ist es, ein System zu entwickeln, die Möglichkeiten zum laden von *Bedarfsmeldungen* bietet. Das System überführt die Informationen aus Jira in eine Struktur, die für die weitere Nutzung im Recommender System Kontext funktioniert. In den *Bedarfsmeldungen* existieren semi strukturierte Volltexte. Diese gilt es durch Ablauf verschiedener Schritte innerhalb des System zu bearbeiten. Dabei soll der Volltext von unngewünschten Zeichen und Formatierungen befreit werden. Dadurch entsteht ein Absatz bestehend aus Wörtern. Durch die Vorverarbeitung verliert der Text an Zusammenhänge durch beispielsweise Satztrennungen durch einen Punkt. Um zusammengehörende Wörter zusammenzubringen, werden verschiedene Schritte durchlaufen. Einerseits sollen vorab Datums-Daten und Zeiten extrahiert werden, da diese durch die Vorverarbeitung durch die Entfernung von Zeichen ebenfalls entfernt werden würden. Durch eine Erstellung von Wortketten können Verbindungen zwischen zwei Wörtern untersucht werden. Eine Wortkette soll hier bedeuten, dass Wörter nebeneinanderstehende Wörter entweder verbunden oder nicht verbunden sind. Zum besseren Verständnis kann folgendes Beispiel betrachtet werden: Die drei Wörter *Ich bin hier* sind durch *Ich bin* und *bin hier* miteinander verkettet. Würde eine Verbindung zwischen *bin* und *hier* getrennt werden, würden zwei Sätze entstehen, nämlich *Ich*

bin und *hier*. Diese Zusammenhänge sollen mithilfe von zwei Schritten unterbrochen werden. Einerseits sollen untypische Wortkombinationen identifiziert und entfernt werden, da diese im ursprünglichen Volltext ebenfalls nicht nebeneinander standen. Der Grund dafür ist, dass bei der Vorverarbeitung der Kontext zusammengehöriger Sätze und Abschnitte verloren geht und dadurch ein Teil der Zusammengehörigkeit wiederhergestellt werden kann. Zudem sollen Wortketten ohne Schlüsselwörter herausgenommen werden, da diese dadurch nicht relevant sind. Um die Anforderungen an das System genauer zu beschreiben, wird ein Use-Case-Diagramm und ein UML-Aktivitätsdiagramm dargestellt und beschrieben. Zudem werden funktionale und nichtfunktionale Anforderungen des Systems erfasst.

3.2.1. Anwendungsfälle

In diesem Kapitel werden die Interaktionen zwischen Benutzer und System beschrieben. Dazu wird ein Use-Case-Diagramm angefertigt, das eine grafische Übersicht über alle Anwendungsfälle bietet.

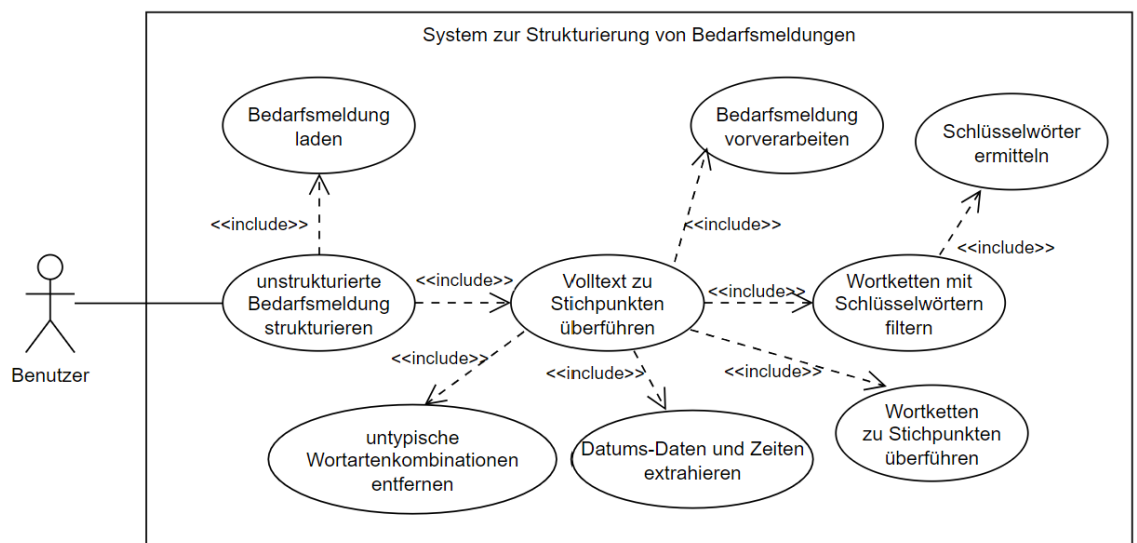


Abbildung 3.2.: Use-Case Diagramm zum System zur Strukturierung von Bedarfsmeldungen.

In der Abbildung 3.2 ist das Use-Case Diagramm zum System zur Strukturierung von *Bedarfsmeldungen* dargestellt. Es existiert ein Akteur, der die Bezeichnung *Benutzer* hat. Dieser hat die Möglichkeit unstrukturierte *Bedarfsmeldungen* zu strukturieren. Dazu kann das System einerseits eine *Bedarfsmeldung* laden und im Falle der unstrukturierten Volltextfelder eine Reihe an Aufgaben durchführen. Um die Volltexte

in Stichpunkte zu überführen, kann der Text mit Vorverarbeitungsverfahren reduziert und angepasst werden. Im Rahmen der Überführung von Stichpunkten können die aus den Volltext erstellten Wortketten auf die gekürzt werden, die mindestens ein Schlüsselwort enthalten. Die Schlüsselwörter können zudem ermittelt werden. Im Anwendungsfall der Überführung des Volltextes in Stichpunkte, können Wortartkombinationen, die nicht zusammengehören in den Wortketten entfernt werden. Dadurch trennen sich neben der Filterung von Schlüsselwörter weitere Verbindungen innerhalb der Wortketten. Datums- und Zeitangaben können neben den anderen Anwendungsfällen separat herausgefiltert werden. Schließlich können die Wortketten zu Stichpunkten überführt werden.

3.2.2. Anforderungen

Im Folgenden werden die funktionalen sowie nichtfunktionalen Anforderungen des Systems beschrieben. Diese Informationen wurden aus den Ergebnissen des Kapitels 2 und dem Kapitel 3.2 hergeleitet und bilden den Rahmen des Systems.

Funktionale Anforderungen

Die funktionalen Anforderungen beschreiben konkrete Funktionalitäten des Systems. Dazu werden zusammengehörende Anforderungen nummeriert und in detaillierte Unterpunkte aufgelistet und beschrieben.

- 1.1 Die *Bedarfsmeldungen* sollen geladen werden können.
- 1.2 Beim laden wird eine *Bedarfsmeldung* ausgewählt.
- 1.3 Die ausgewählte *Bedarfsmeldung* muss in die vordefinierte Struktur überführt werden.
- 2.1 Das System soll Methoden der Vorverarbeitung zur Bereinigung eines Volltextes anwenden können.
- 2.2 Die Vorverarbeitung soll ein Volltext als Eingabe erhalten.
- 2.3 Das Vorverarbeitung soll den bereinigten Volltext als Ausgabe zurückgeben.
- 3.1 Das System soll eine Methode zur Identifizierung von Schlüsselwörtern anwenden können.
- 3.2 Das Modul soll eine Liste mit Schlüsselwörtern anlegen.

- 4.1 Das System soll Wortketten aus einem Volltext erstellen können.
- 4.2 Das Modul soll einen Volltext als Eingabe für diese Methode erhalten.
- 4.3 Als Rückgabe soll das Modul eine Liste mit Wortketten zurückgeben.
- 4.4 Das System soll eine Wortketten-Liste auf die Wörter reduzieren, die mindestens ein Schlüsselwort enthalten.
- 5.1 Das System soll eine Methode zur Extraktion von Zeitangaben anwenden können.
- 5.2 Das Modul soll aus einem Volltext Datums- und Zeitangaben extrahieren.
- 6.1 Das System soll eine Methode zur Identifikation von untypischenn Wortkombinationen anwenden können.
- 6.2 Das Modul soll Wortkombinationen aus der Wortkette erhalten und überprüfen ob die Wörter nebeneinander stehen dürfen.
- 6.3 Wortketten werden entfernt, die nicht nebeneinander stehen sollen.
- 7.1 Das System soll die Wortketten in zusammengehörende Stichpunkte zusammenführen.
- 7.2 Als Ergebnis wird eine Liste mit Stichpunkten zurückgegeben.
- 8.1 Die in Stichpunkte umgebauten Volltexte werden der *Bedarfsmeldung* beigelegt.
- 8.2 Die Datums- und Zeitangaben werden den Stichpunkten beigelegt.
- 8.3 Das Ergebnis wird dem Benutzer ausgegeben.

Nichtfunktionale Anforderungen

Hierbei handelt es sich um qualitätsbezogene Anforderungen. Diese umfassen nicht konkrete Funktionen des Systems, sondern stellen Rahmenbedingungen des Systems im Ganzen zusammen.

- 1. Das System soll modular aufgebaut sein, um die einzelnen Schritte und Methoden austauschen zu können.
- 2. Das System soll in der Lage sein, mehrere *Bedarfsmeldungen* laden zu können.

3. Das System muss in der Lage sein, die *Bedarfmeldungen* in einer akzeptablen Zeitspanne zu verarbeiten.
4. Das Ergebnis muss deterministisch sein.

3.2.3. Ablauf des Systems

Der Ablauf des Systems zur Strukturierung von *Bedarfmeldungen* wird mit Hilfe eines UML-Aktivitätsdiagramm dargestellt, das eine Erweiterung des Aktivitätsdiagramms aus der Abbildung 3.1 durch eine hierarchische Schachtelung darstellt.

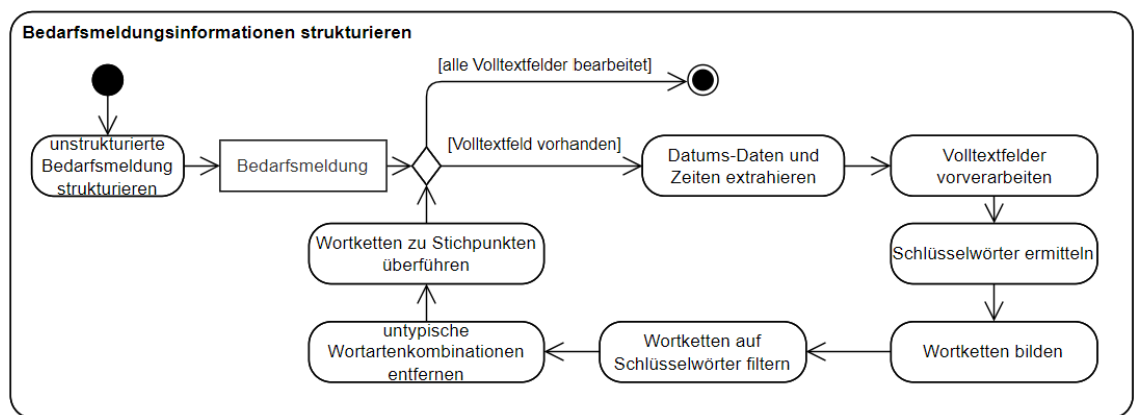


Abbildung 3.3.: Abstrakte Darstellung des Systems zur Strukturierung von *Bedarfmeldungen*.

In Abbildung 3.3 ist der abstrakte Ablauf des Systems dargestellt. Zu Beginn wird der unstrukturierte Informationsfelder mit *Bedarfmeldungsinformationen* in eine strukturierte Form gebracht. Anschließend werden die Volltextfelder durch eine Reihe an Schritten geleitet. Angefangen mit der Extraktion von Datums-Daten und Zeiten. Die Volltextfelder werden vorverarbeitet, wodurch ungewollte Formatierungen entfernt werden. Darauffolgend werden Schlüsselwörter im Volltext ermittelt. Aus dem Volltext werden Wortketten gebildet und auf die Wörter reduziert, die Schlüsselwörter enthalten. Dann werden untypische Wortartenkombinationen entfernt, wodurch die Wortketten weiter separiert werden. Zum Schluss werden die einzelnen Wortketten in Stichpunkte überführt und in die *Bedarfmeldung* aufgenommen. Dieser Prozess wird für alle Volltextfelder durchgeführt.

4. Literaturüberblick

Es gibt eine Reihe an verwandten Arbeiten, die sich mit unterschiedlichen Aspekten des *Staffing*-Prozesses und der Nutzung von Information Retrieval und Filtering zur Informationsgewinnung beschäftigen. Dennoch beschäftigt sich keine Arbeit mit dem spezifischen Problem der Informationsgewinnung aus *Bedarfsmeldungen*, in der Art, wie sie die *adesso* verwendet. Es wird ein Einblick in die Art und Weise gegeben, wie andere Autoren Information Retrieval einsetzen und kombinieren.

4.1. Techniken

Nachfolgend werden Details zu verschiedene Arbeiten absatzweise dargestellt. Jeder Absatz befasst sich mit unterschiedlichen Techniken, die Lösungen für die Implementierungsebene des zu entwickelnden Systems bieten.

Automatisiertes Staffing

In der Arbeit “Information retrieval, fusion, completion, and clustering for employee expertise estimation”[11] beschreiben Horesh, Varshney und Yi einen Ansatz zur Ableitung von Unternehmensdaten und digitalen Fußabdrücken von Mitarbeitern. Mit Hilfe eines Big-Data-Workflows, der die Komponenten Information Retrieval und Suche, Datenfusion, Matrixvervollständigung und ordinale Regression nutzt, können Informationen zur Expertise automatisch zusammengeführt und für die Nutzung durch Experten aufbereitet werden. Das System soll Fähigkeiten, Talente und Fachwissens der Mitarbeiter in einem breiten Bereich wie cloud computing oder cybersecurity einschätzen. Beim Ansatz des Information Retrieval und -fusion wird eine Liste von Suchbegriffen erstellt, die sich auf das breite Fachgebiet der Mitarbeiter beziehen. Die Suche wird nach jedem dieser Abfragebegriffe durchgeführt, um Zusammenhänge zwischen Mitarbeiter und Datenquellen zu finden. Die verschiedenen Zusammenhänge werden miteinander zusammengefügt, gewichtet und nach der Abfrage sortiert. Die Mitarbeiter werden nach Daten gewichtet und bewertet, um einen einzigen Wert (sehr niedrig, niedrig, moderat, etwas, begrenzt) für ihr Fachwissen in diesem breiten Bereich zu erhalten.

Vorverarbeitung

Der Autor Jain und Srivastava der Arbeit “Data mining techniques: a survey paper”[12] beschreibt Data-Mining als ein interdisziplinäres Teilgebiet der Informatik, das sich mit der rechnergestützten Entdeckung von Mustern in großen Datenbeständen befasst [12]. Ziel dieses fortgeschrittenen Analyseverfahrens ist es, Informationen aus einem Datensatz zu extrahieren und in eine für die weitere Verwendung verständliche Struktur umzuwandeln [12]. Die verwendeten Methoden liegen an der Schnittstelle zwischen künstlicher Intelligenz, maschinellem Lernen, Statistik, Datenbanksystemen und Business Intelligence [12]. Beim Data Mining geht es um die Lösung von Problemen durch die Analyse von Daten, die bereits in Datenbanken vorhanden sind [12].

In der Arbeit “Review of data preprocessing techniques in data mining”[1] zeigen die Autoren Alasadi und Bhaya Wege und Schritte zur Aufbereitung von Datensätzen auf. Die Arbeit umfasst Data-Mining Vorverarbeitungsmethoden, um die Qualität der Daten zu verbessern. Diese weisen wichtige Schritte auf, um die Effizienz in der Datensammlung zu verbessern. Die Datenvorverarbeitung (preprocessing) stellt eine der essenziellen Data-Mining-Aufgaben dar, die die Vorbereitung und Umwandlung von Daten in eine geeignete Form umfasst. Die Datenvorverarbeitung zielt unter anderem darauf ab, die Datenmenge zu reduzieren und Daten zu standardisieren. Die Datenvorverarbeitung umfasst eine Reihe von Techniken, darunter Datenbereinigung, -integration, -transformation und -reduktion.

Die Arbeit mit dem Titel “Preprocessing of requirements specification”[14] von dem Autor Kroha wird der Teil des Anforderungsspezifikationsprozesses diskutiert, der zwischen der textuellen Anforderungsdefinition und den dazugehörigen Diagrammen der Anforderungsspezifikation liegt. Es wird die These aufgestellt, dass die Erstellung einer textuellen Anforderungsbeschreibung, die das Verständnis des Analysen für das Problem darstellt, die Effizienz der Anforderungvalidierung durch den Benutzer verbessert. Die vorliegende Idee ist aus dem Problem entstanden, dass Software-Entwickler nicht immer über die erforderlichen Kenntnisse in den fachlichen Abläufen der Themengebiete verfügen, die für die Erstellung der Software relevant sind. Im Rahmen der Anforderungsdefinition erfolgt eine textuelle Verfeinerung, die als Anforderungsbeschreibung bezeichnet werden kann. Bei der Arbeit mit dem unterstützten Werkzeug *Tessi* ist der Analytiker durch die genannten Vorgaben gezwungen, Anforderungen zu vervollständigen und zu erklären sowie die Rollen der Wörter im Text im Sinne der objektorientierten Analyse zu spezifizie-

ren. Im Rahmen der Vorverarbeitung erfolgt eine Transformation der Requirements durch Templates.

Schlüsselwörter identifizieren

In der Arbeit “Using tf-idf to determine word relevance in document queries”[29] haben die Autoren Ramos u. a. die *TF-IDF*-Methode (*Term Frequency-Inverse Document Frequency*) zur Ermittlung der Häufigkeit von Wörtern in einem bestimmten Dokument im Vergleich zum Anteil dieses Wortes im gesamten Dokumenten ermittelt. Die Berechnung erlaubt eine Einschätzung der Relevanz eines bestimmten Wortes in einem bestimmten Dokument. Die Grundidee des Ansatzes besteht darin, dass Wörter, die in einem einzigen Dokument oder in einer kleinen Gruppe von Dokumenten häufig vorkommen, tendenziell höhere *TF-IDF*-Werte aufweisen als häufig vorkommende Wörter wie Artikel und Präpositionen. *TF-IDF* stellt laut den Autoren ein effizientes Verfahren zum Abgleich von Wörtern in einer Anfrage mit Dokumenten dar. Bei Eingabe einer Abfrage zu einem bestimmten Thema durch einen Benutzer kann *TF-IDF* relevante Informationen zu dieser Abfrage in Dokumenten finden. Trotz der Stärken von *TF-IDF*, sind auch seine Grenzen zu berücksichtigen. In Bezug auf Synonyme ist zu beachten, dass *TF-IDF* nicht auf die Beziehung zwischen den Wörtern eingeht. Des Weiteren werden unterschiedliche Schreibweisen von Wörtern nicht berücksichtigt, was dazu führen kann, dass Wörter fälschlicherweise als nicht so häufig auftauchend deklariert werden, obwohl sie mit leicht abgewandelter Schreibweise häufiger vorkommen.

Die Autoren haben die *TF-IDF*-Methode anhand von 1400 Dokumenten getestet. Dazu wurden die *TF-IDF*-Werte berechnet und die ersten 100 Dokumente zurückgegeben. Die zurückgegebenen Dokumente werden in absteigender Reihenfolge zurückgegeben, wobei die Dokumente mit höheren Gewichtssummen zuerst erscheinen. Um die Ergebnisse zu vergleichen, wurde zu mehreren Dokumenten die Anzahl einer bestimmten Anfrage ermittelt.

Die Autoren Bafna, Pramod und Vaidya der Arbeit “Document clustering: TF-IDF approach”[2] haben im Rahmen der Textanalyse die *TF-IDF*-Methode angewendet, um häufige Begriffe zu eliminieren und lediglich die relevantesten Begriffe aus einem Textkorpus zu extrahieren. In der Untersuchung wird der *TF-IDF*-Algorithmus zusammen mit dem *Fuzzy K-means* und dem *hierarchischen* Algorithmus verwendet. In einem ersten Schritt werden Experimente mit einem kleinen Datensatz durchgeführt und eine Clusteranalyse vorgenommen. Im Anschluss erfolgt die Anwendung des vorher besten ermittelten Algorithmus auf den erweiterten Datensatz. In Kom-

bination mit den verschiedenen Clustern der verwandten Dokumente werden der resultierende *Silhouettenkoeffizient*, die *Entropie* sowie der *F1-Score* dargestellt, um das Verhalten des Algorithmus für den Datensatz zu veranschaulichen.

Wortketten bilden

Die Autoren Majumder, Mitra und Chaudhuri beschreiben in der Arbeit “N-gram: a language independent approach to IR and NLP”[18] *N-Gramme* als Folgen von Zeichen oder Wörtern, die aus einem Text extrahiert werden. Diese lassen sich in zwei Kategorien unterteilen: i) zeichenbasiert und ii) wortbasiert. Ein Zeichen-*N-Gramm* bezeichnet eine Folge von n aufeinanderfolgenden Zeichen, die aus einem Wort extrahiert werden. Die Hauptmotivation hinter diesem Ansatz besteht darin, dass ähnliche Wörter einen hohen Anteil an *N-Grammen* gemeinsam haben werden. In der Regel umfasst ein *N-Gramm* lediglich die am häufigsten auftretenden Wortpaare und verwendet einen Backoff-Mechanismus, um die Wahrscheinlichkeit zu berechnen, die bei der Suche nach dem gewünschten Wortpaar nicht erfolgreich war. Die Analyse von *N-Grammen* erlaubt die Identifikation häufig vorkommender Phrasen oder Begriffe, die als potenzielle Schlüsselwörter bezeichnet werden können.

In der Durchführung des Experimentes werden *N-Gramme* dazu verwendet, um die indische Sprache aus mehrsprachigen Dokumentensammlungen zu identifizieren. Dazu werden zunächst *N-Gramm*-Profile für 10 indische Sprachen erstellt. Ein *N-Gramm*-Häufigkeitsprofil wird durch Zählen aller *N-Gramme* in einer Reihe von Dokumenten in einer bestimmten Sprache und deren Sortierung in absteigender Reihenfolge erstellt. Im Falle der Identifizierung einer neuen Dokumentsprache wird ein *N-Gramm*-Profil des Dokuments erstellt und anschließend der Abstand zwischen dem neuen Dokumentprofil und den Sprachprofilen berechnet. Der Abstand wird mit dem *out-of-place measure* zwischen den beiden Profilen berechnet. Der kürzeste Abstand wird ausgewählt und es wird vorhergesagt, ob das bestimmte Dokument zu dieser Sprache gehört. Zur Vermeidung einer Fehlklassifikation wurde ein Schwellenwert eingeführt, bei dessen Überschreitung das System die Aussage trifft, dass die Sprache des Dokuments nicht bestimmt werden kann.

Wortgruppen identifizieren

Die Autoren Kumawat und Jain der Arbeit “POS tagging approaches: A comparison”[15] beschreiben unterschiedliche Ansätze von *POS-Tagging* (*Part-of-Speech-*

Tagging), um einen für indische Sprache zu erstellen. Sie beschreiben, dass die Katalogisierung von Wortarten (*POS*) einen Prozess bezeichnet, bei dem jedem einzelnen Wort eines Satzes ein Wortart-Tag oder ein anderes philologisches Klassenzeichen zugeordnet wird. Die Vorverarbeitungsaufgabe des *Taggings* von Sprachbestandteilen stellt einen essenziellen Schritt in *NLP* dar.

NLP (Natural Language Processing) stellt einen zentralen Aspekt im Bereich der künstlichen Intelligenz sowie der Computerwissenschaften dar [13]. Studien in diesem Bereich umfassen Theorien und Methoden, die eine Kommunikation zwischen Menschen und Computern in natürlicher Sprache ermöglichen [13]. *NLP* vereint die Gebiete Informatik, Linguistik und Mathematik mit dem primären Ziel, menschliche Sprache in Befehle zu übersetzen, die von Computern ausgeführt werden können [13].

Die Zuordnung von Wortarten stellt laut den Autoren Kumawat und Jain der Arbeit “POS tagging approaches: A comparison”[15] eine grundlegende Aufgabe bei der Verarbeitung natürlicher Sprache dar. Die Erstellung erfolgt unter Zuhilfenahme linguistischer Theorien, zufälliger Muster sowie einer Kombination aus beidem. Ein *POS-Tagger* ist definiert als ein Teil einer Software, der jedem Wort einer Sprache, das er liest, eine Wortart zuordnet. Die Ansätze des *POS-Tagging* lassen sich in drei Kategorien unterteilen: *regelbasiertes Tagging*, *statistisches Tagging* und *hybrides Tagging*. Im Rahmen der Zuweisung von *POS-Tags* zu Wörtern im regelbasierten *POS-System* erfolgt die Verwendung einer Reihe von handgeschriebenen Regeln in Kombination mit Kontextinformationen. Der Nachteil dieses Systems besteht darin, dass es nicht funktioniert, wenn der Text nicht bekannt ist. Das Problem besteht darin, dass das System nicht in der Lage ist, den passenden Text vorherzusagen. Um eine höhere Effizienz und Genauigkeit in diesem System zu erreichen, ist es daher empfehlenswert, einen umfassenden Satz von handkodierte Regeln zu verwenden. Die Häufigkeit und Wahrscheinlichkeit sind in den statistischen Ansatz einbezogen. Der grundlegende statistische Ansatz basiert auf der am häufigsten verwendeten Markierung für ein bestimmtes Wort in den annotierten Trainingsdaten. Diese Information wird auch zur Markierung dieses Wortes im unannotierten Text verwendet.

Das Experiment aus der Arbeit umfasst ein Datensatz aus 20000 Sätzen mit manuellen Tags in der Sprache Marathi. Damit kann das entwickelte *POS-Tagging-System* getestet werden.

Datum und Zeitangaben identifizieren

Die Arbeit “Named entity recognition approaches”[19] der Autoren Mansouri, Affendey und Mamat vergleichen verschiedene *NER*-Ansätze (Named Entity Recognition) auf die Genauigkeit, um die Stärken und Schwächen einzelner Methoden zu identifizieren. *NER* stellt einen Teilbereich der Informationsextraktion dar und umfasst die Verarbeitung sowohl strukturierter als auch unstrukturierter Dokumente und die Identifizierung von Wörtern, die sich auf Personen, Orte, Organisationen und Unternehmen beziehen. *NER* stellt eine grundlegende Aufgabe eines Systems im Bereich des *NLP* dar. *NER* umfasst zwei Aufgaben. Die erste Aufgabe besteht in der Identifizierung von Eigennamen im Text. Die zweite Aufgabe ist die Klassifizierung dieser Namen in eine Reihe von vordefinierten Kategorien. Dazu zählen (i)Personennamen, (ii)Organisationen, wie Unternehmen, Regierungsorganisationen, Ausschüsse usw., (iii)Orte, wie Städte, Länder, Flüsse usw., (iv)Datumsangaben und (v)Zeitangaben.

Hybride Ansätze

In der Untersuchung “Combining approaches to information retrieval”[7] wird von den Autoren Croft die Entwicklung von Kombinationen im Bereich des Information Retrievals analysiert. Dabei werden sowohl experimentelle Ergebnisse als auch die Retrieval-Modelle, die als formale Rahmen für die Kombination vorgeschlagen wurden, berücksichtigt. Es wird aufgezeigt, dass Kombinationsansätze für die Informationssuche als Kombination der Ergebnisse mehrerer Klassifikatoren auf der Grundlage einer oder mehrerer Darstellungen modelliert werden können. Zudem wird dargelegt, dass dieses einfache Modell Erklärungen für viele der experimentellen Ergebnisse liefern kann.

Die Arbeit “LSTM, VADER and TF-IDF based hybrid sentiment analysis model”[6] von den Autoren Chiny, Chihab, Bencharef u. a. kombiniert drei Ansätze des Information Retrievals mit dem Ziel, relevante Informationen aus Produktreviews zu extrahieren. Der Ansatz *TF-IDF* wird mit einem sogenannten *CLASSIFIER* Modell kombiniert. Das Klassifikationsmodell verarbeitet drei Eingaben der Modelle *LSTM*, *VADER* und *TF-IDF*. Die Werte dieser Eingaben liegen im Bereich von [0,1]. Die Ausgabe des Klassifikationsmodells ist binär und gibt eine Vorhersage des vollständigen Textes der Modelleingabe aus (positiv oder negativ). Aus einem Datensatz wurden 5000 zufällige Bewertungen ausgewählt, die sich von den für die *LSTM*- und *TF-IDF*-Modelle verwendeten Trainings- und Testdatensätzen unterscheiden.

Die Autoren Chiny, Chihab, Bencharef u. a. haben sie durch die Eingabe des globalen Modells laufen lassen, um die Vorhersagen zu erhalten, die von den Modellen *LSTM*-, *VADER*- und *TF-IDF*-Modelle zu erhalten. Anschließend teilten sie diese Ergebnisse in zwei Stapel auf (75 % für die Trainingsmenge und 25 % für die Testmenge), um das binäre Klassifizierungsmodell zu trainieren und zu bewerten. Die Evaluation erfolgt durch den Einsatz der Methoden *Precision*, *Recall* und *F1-Score*.

Die Arbeit “A Hybrid TF-IDF and N-Grams Based Feature Extraction Approach for Accurate Detection of Fake News on Twitter Data”[33] von den Autoren Suhasini und Vimala befasst sich mit der Filterung von Fake news. In diesem Beitrag werden hybride Verfahren zur Gewinnung von Merkmalen untersucht, die in dem Gebiet noch nicht gründlich erforscht wurden. Die Anwendung von Hybrid-systemen hat sich in einer Vielzahl von Anwendungsbereichen als nützlich erwiesen und zeigen eine Tendenz, die Fehlerquote zu reduzieren, indem sie Techniken wie *TF-IDF* und *N-Grams* verwenden. Es wurden Experimente unter Verwendung von Echtzeit-Twitterdaten durchgeführt. Der Datensatz umfasste ca. 5.800 Tweets, die sich auf Donald-Drummond-Geschichten bezogen. Die Sammlung und Verarbeitung der Tweets erfolgt mit Python. Der Datensatz umfasste Original-Tweets, die als gefälscht und echt gekennzeichnet wurden. Die Genauigkeit der Prognose wurde anhand der verschiedenen Nachrichten evaluiert, die für das Training verwendet wurden und am Ende mit *Precision*, *Recall* und *F1-Score* evaluiert.

Im Rahmen der Studie mit dem Titel “Hybrid Keyword Extraction Algorithm and Cosine Similarity for Improving Sentences Cohesion in Text Summarization”[8] wurde von den Autoren Darmawan und Wahono ein hybrider Algorithmus zur Extraktion von Schlüsselwörtern und Kosinusähnlichkeit zur Verbesserung der Satzkohäsion bei der Textzusammenfassung vorgeschlagen. Die vorgeschlagene Methode basiert auf einer Komprimierung von 50 %, 30 % und 20 %, um Kandidaten für die Zusammenfassung zu erstellen. Die Auswertung des Ergebnisses mittels *t-Test* zeigt, dass die vorgeschlagene Methode den Kohäsionsgrad signifikant erhöht.

Der Ablauf umfasst die Analyse eines Dokuments mithilfe eines Extraktionsalgorithmus sowie die Berechnung der *TF-IDF*-Werte für jeden Begriff. Anschließend werden alle *TF-IDF*-Werte für jeden Satz summiert. Im nächsten Schritt werden alle Sätze anhand der Summe von *TF-IDF* eingestuft. Das Kompressionsverhältnis bestimmt die Position des Satzrangs. In dieser Studie wird eine Kompression von 50 % verwendet, was bedeutet, dass die Satzzusammenfassung um 50 % des Originaltextes gekürzt wird. Nach der Auswahl des Satzes wird dessen Berechnung

durchgeführt. Die Ähnlichkeit wird mit der *Cosinus-Ähnlichkeitsmethode* berechnet. Anschließend werden alle Sätze anhand ihrer *Cosinus-Ähnlichkeit* von der höchsten zur niedrigsten sortiert. Der resultierende Text mit neuer Satzanordnung stellt die finale Zusammenfassung dar.

Pipeline

In der Arbeit “Implementierung und Visualisierung N-Gramm-basierter Word-Clouds” [27] wird von dem Autor Pirk eine Pipeline entwickelt, die die *N-Gramm*-Analyse verwendet, um Schlagwörter aus einem Text zu extrahieren und mit verschiedenen Ansätzen von Word-Clouds zu visualisieren. Der Fokus dieser Studie liegt dennoch eher auf der Visualisierung als auf der Informationsgewinnung eines Textes.

Die Arbeit “Analyzing documents with TF-IDF” [16] von dem Autor Lavin präsentiert eine Anleitung zur Erstellung einer Pipeline mit Python und *TF-IDF*. Darüber hinaus wird die Relevanz von *TF-IDF* als Vorverarbeitung beim maschinellen Lernen erörtert. Im Vergleich zur rohen Termhäufigkeit weist *TF-IDF* in der Regel einen höheren Vorhersagewert auf. Die Gewichtung von Themenwörtern wird erhöht, um die Bedeutung von Wörtern zu erhöhen, während die Gewichtung von hochfrequenten Funktionswörtern verringert wird. Es werden Verfahren zur Vorverarbeitung von Texten vorgestellt, die eine Umformung in die gewünschte Darstellungsform ermöglichen. Zudem werden Methoden zur Interpretation der Ergebnisse des *TF-IDF*-Verfahrens erörtert. Die vorliegende Arbeit widmet sich zunächst einer detaillierten Betrachtung der zugrundeliegenden Algorithmen und ihrer Funktionsweise. Im Anschluss erfolgt die Implementierung in Python. Die Verwendung der Bibliothek *sklearn* ist dabei von zentraler Bedeutung.

In dem Beitrag mit dem Titel “Design and implementation of an open source Greek POS Tagger and Entity Recognizer using spaCy” [25] wird von den Autoren Partalidou, Spyromitros-Xioufis, Doropoulos u. a. ein maschineller Lernansatz für die Bereiche *POS-Tagging* und *NER* für die griechische Sprache unter Verwendung von *spaCy* erarbeitet und evaluiert. Die Verarbeitung natürlicher Sprache wirft insbesondere bei der Analyse unüblicher Sprachen wie Griechisch Schwierigkeiten auf. Der Datensatz wurde aus Texten einer griechischen Zeitung extrahiert. Die Artikel der Zeitung wurden in verschiedene Kategorien wie beispielsweise Sport, Gesundheit, Wirtschaft und politische Nachrichten eingeteilt. Die Daten bestehen aus einer Reihe von XML-Dateien, die Informationen auf der Ebene von Absätzen, Sätzen

und Wörtern enthalten. Im Rahmen der Evaluation wurden verschiedene Parameter getestet, um das optimale Ergebnis zu erzielen. Der Datensatz wurde gemischt und in einen Trainingssatz, einen Testsatz und einen Validierungssatz aufgeteilt. Zur Evaluierung wurden die Methoden des *Precision*, *Recalls* und *F1-Scores* angewendet.

4.2. Ergebnis

In der Literatur finden sich verschiedene Ansätze zur Extraktion relevanter Stichpunkte aus einem Volltext. Eine Methode zur Ermittlung wichtiger Schlüsselwörter in Texten stellt die *TF-IDF*-Methode dar. Innerhalb einer oder mehrerer *Bedarfsmeldungen* lassen sich mit dieser Methode häufig auftauchende Wörter ermitteln. Es besteht somit die Möglichkeit, Wörter aus einer *Bedarfsmeldung* mit anderen *Bedarfsmeldungen* zu vergleichen und die Häufigkeit der Wörter zu berechnen, um somit potenzielle Schlüsselwörter zu ermitteln. Ein Ansatz zur Bildung von Wortketten ist die Nutzung von *N-Grammen*. Dabei lassen sich Sätze und Absätze zur weiteren Verarbeitung separieren, wodurch Wörter und ihre Zusammenhänge identifizieren lassen. Auch grammatische Kategorien von Wörtern können Rückschlüsse auf potenzielle Schlüsselwörter zulassen. Die *POS-Tagging*-Methode stellt eine Möglichkeit dar, um dieses Ziel zu erreichen. Des Weiteren kann *NER* dazu beitragen Zeitangaben zu identifizieren. Hybride, die sich durch besondere Zusammensetzungen auszeichnen, zeigen Überlegenheiten gegenüber den einzelnen Ansätzen, die für sich genommen jeweils nur eine Teilkompetenz abdecken.

5. Umsetzung

Auf Basis welcher Methodiken und Ansätze die relevanten Informationen extrahiert werden können, wurde in Kapitel 4 dargestellt. Nun gilt es diese Ansätze in einem System zu implementieren. In diesem Kapitel werden die Funktionalitäten des Systems zusammengetragen. Dabei werden verwendete Technologien und Implementationsaspekte genauer beschrieben.

5.1. Beschreibung

Das System durchläuft verschiedene Schritte, weswegen es als Pipeline implementiert wird. Damit alle Ansätze und Methoden zur Extraktion von Informationen gut funktionieren und vergleichbar bleiben, wird eine Übersetzungsfunktion der *Bedarfsmeldungen* hinzugefügt. Auch wenn die *Bedarfsmeldungen* in den meisten Fällen auf Deutsch sind, hilft es diese zu übersetzen, damit keine Unterschiede in der Ergebnisqualität resultiert, da einige Methoden und Ansätze auf Basis von Englischen Trainingssätzen trainiert wurden. Schließlich müssen alle aus Kapitel 4 untersuchten Ansätze implementiert und nutzbar sein. Sie sollen die Möglichkeit haben *Bedarfsmeldungen* als Input zu erhalten und eine strukturierte *Bedarfsmeldung* als Ausgabe zurückzugeben. Zur vereinfachten Entwicklung soll das System modular sein, damit Methoden und Ansätze nach belieben durchgetauscht und verwendet werden können.

5.2. Konkreter Ablauf der Pipeline

Dieses Kapitel beschreibt den Ablauf der Pipeline. Dazu wird das Aktivitätsdiagramm aus der Abbildung 3.3 verfeinert und mit allen analysierten Komponenten aus Kapitel 4 genauer beschrieben. Die Abbildung 5.1 zeigt das UML-Aktivitätsdiagramm der Python Pipeline zur Strukturierung von *Bedarfsmeldungen*. Im ersten Schritt *unstrukturierte Bedarfsmeldung strukturieren* im Ablauf der Pipeline wird eine *Bedarfsmeldung* ausgewählt und in die festgelegte *Bedarfsmeldungsstruktur* aus Kapitel 2.8 umgebaut. Die Felder *Einsatzbeginn* und *Einsatzende* werden zu einem Feld *Einsatz* zusammengefügt. Neben den Feldern *Aufgaben* und *Skills* bleiben alle weiteren Felder bis zum letzten Punkt Ausgabe unverändert. Die Felder *Aufgaben* und *Skills* werden jeweils in die Schleife weitergeleitet und auf Stichpunkte reduziert.

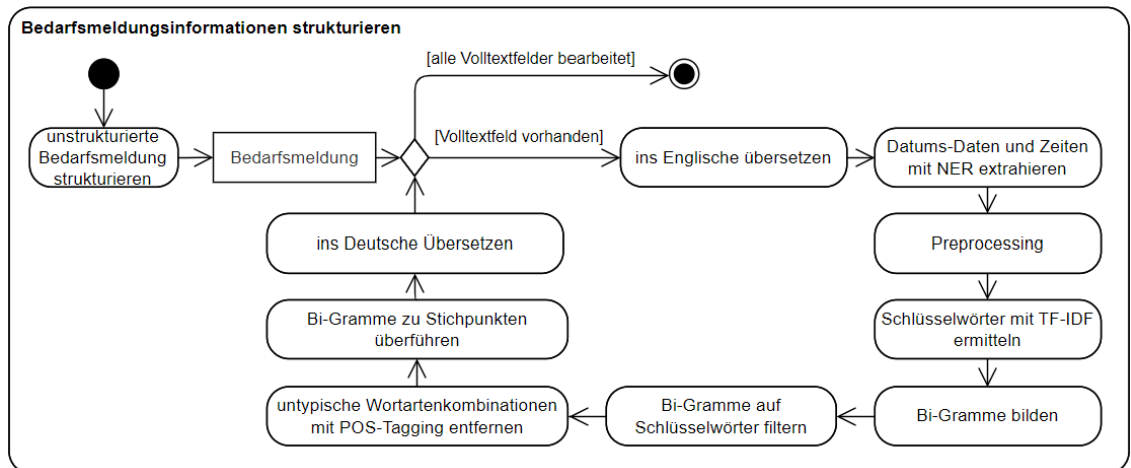


Abbildung 5.1.: UML-Aktivitätsdiagramm der Pipeline.

Der Prozess zur Reduktion durchläuft mehrere Schritte. Zu Beginn werden die Volltexte aus den Feldern im Punkt *ins Englische übersetzen* übersetzt. Anschließend erfolgt im Schritt *Datums-Daten und Zeiten mit NER extrahieren* die Extrahierung der zeitbezogenen Daten. Darauf folgt die Vorverarbeitung des Volltexts im Punkt *Preprocessing*. Der Grund, warum die Extraktion mit *NER* nicht nach der Vorverarbeitung durchgeführt werden kann, ist, dass im Vorverarbeitungsschritt alle Nummern und somit auch alle zeitbezogenen Daten entfernt werden. Die Resultate des *NER* werden zum Ende hin zurück ins Deutsche übersetzt und der fertigen Stichpunktliste beigefügt. Zur Erstellung der Stichpunktliste werden relevante Schlüsselwörter durch die *TF-IDF*-Methode ermittelt. Anschließend wird der vorverarbeitete Text zu *Bi-Grammen* umgeformt. Beim Punkt *Bi-Gramme auf Schlüsselwörter filtern* werden alle *Bi-Gramme* entfernt, die kein Schlüsselwort aus der *TF-IDF*-Methode enthalten. Somit erhält der Schritt *untypische Wortartenkombinationen mit POS-Tagging entfernen* eine reduzierte Liste mit *Bi-Grammen*, bei dem mindestens eines der beiden *Bi-Gramm*-Wörter ein Schlüsselwort darstellt. Jedes Wort der *Bi-Gramm*-Liste durchläuft eine weitere Filterung. Im Englischen existieren Wortarten, die typischerweise nicht nebeneinander stehen. Durch Entfernung dieser *Bi-Gramme* durch POS-Tagging-Kombinationen erfolgt eine weitere Trennung der Wortketten in der *Bi-Gramm*-Liste. Wörter, die nicht zusammengehören, verlieren dadurch die Verbindung zueinander. Im darauffolgenden Schritt *Bi-Gramm zu Stichpunkten überführen* werden die *Bi-Gramme* jeweils zu einem String zusammengefügt, bei dem im darauffolgenden *Bi-Gramm* ebenfalls ein relevantes Wort enthalten ist. Dadurch formt sich eine Liste mit Stichpunkten bestehend aus eins bis x vielen Wörtern, die Bezug zueinander haben. Im

Schritt *ins Deutsche übersetzen* wird die Liste mit Stichpunkten zurück ins übersetzt. Dieser Ablauf wird für die beiden Felder *Aufgaben* und *Skills* durchlaufen, da diese Volltextfelder darstellen.

5.3. Implementationsdetails

Dieses Kapitel beschreibt den technischen Entwicklungsprozess zur Umsetzung der Anforderungen des Systems. Die Implementierung fokussiert sich auf die Umsetzungen von Technologien und Funktionsweisen verschiedener Anforderungen.

5.3.1. Pipeline

Die Pipeline wurde in der Programmiersprache Python umgesetzt. Python hat sich zu einer der populärsten interpretierten Programmiersprachen entwickelt [23]. Die Programmiersprache eignet sich insbesondere für die Erstellung kleiner Programme und Skripte, die zur Automatisierung von Aufgaben eingesetzt werden können [23]. Python hat eine große und aktive Community für wissenschaftliche Berechnungen und Datenanalysen hervorgebracht und hat sich in den letzten Jahren zu einer der wichtigsten Sprachen für Data Science, maschinelles Lernen und allgemeine Softwareentwicklung in Wissenschaft und Industrie entwickelt [23]. Python unterstützt Modularität, wodurch ein Teil der Anforderungen somit abgedeckt werden kann. Die Implementierungen der einzelnen Verfahren aus den Anforderungen werden nicht manuell, sondern auf Basis von bereits existierende Bibliotheken umgesetzt.

5.3.2. Projektstruktur

Das Projekt wird in einem git Repository gespeichert und versioniert. Die einzelnen Module sind in dem Verzeichnis `modules/` in separaten `.py` Dateien gelagert. Die *Bedarfsmeldungen* werden im `requirements/`-Verzeichnis gespeichert. Alle relevanten *Bedarfsmeldungen* sind in einer `jiraTickets.json`-Datei als Liste gespeichert. Die Datei `app.py` ist der Kern der Pipeline. Diese importiert alle Module und implementiert die Struktur der Pipeline. Das Projekt kann über den Befehl `> py app.py` ausgeführt werden.

5.3.3. Modulimplementationen

Nachfolgend werden Implementationsdetails zu den einzelnen Modulen gegeben. Dabei werden verwendete Bibliotheken und Code-Details näher erläutert.

Strukturierung von Bedarfsmeldungen

Die *Bedarfsmeldungen* wurden über die Jira-Schnittstelle extrahiert und im Verzeichnis `requirements/jiraTickets.json` gespeichert. Zur Eingrenzung der Datenmenge wurde ein Filter angewendet, der nur die *offenen* und *eskalierten Bedarfsmeldungen* zurückgibt. Diese sind die relevanten und noch aktuellen *Bedarfsmeldungen*. Zudem wurden alle irrelevanten Felder mit einem weiteren Filter herausgenommen. Die Daten aus der Jira-API bestehen namentlich aus *customfields* mit einer angehängenden ID. Der Softwareprototyp lädt in dem Modul *readRequirements.py* die unstrukturierten Fields und formt diese in dem Modul *transformRequirements.py* in *Bedarfsmeldungs*-Objekte um.

Display-Felder	Jira-API-Felder	Objekt-Felder
Überschrift	summary	header
Rolle	customfield_15321	role
Aufgaben	customfield_10288	tasks
Skills	customfield_10296	skills
Skill-Level	customfield_15322	skillLevel
Kunde	customfield_10279	customer
Einsatzort	customfield_10297	location
Beginn	customfield_10293	timePeriod
Ende	customfield_10294	timePeriod
Tagessatz	customfield_10298	dailyRate

Tabelle 5.1.: Übersicht der Datenfelder

In der Tabelle 5.1 ist in der ersten Spalte eine Übersicht der Datenfelder, wie diese in der Abbildung 2.5 mit dem Mockup einer standardisierten *Bedarfsmeldung* auftauchen. In der zweiten Spalte sind die dazugehörigen Feldernamen, die in der `jiraTickets.json` enthalten sind. Die dritte Spalte spiegelt die jeweiligen Namen innerhalb des *Bedarfsmeldungs*-Objekts im Prototypen wieder. Das *Bedarfsmeldungs*-Objekt dient der strukturierten Handhabung der *Bedarfsmeldungs*-Daten innerhalb des Systems.

Übersetzung

Das Modul *translate.py* ist dazu da, um die *Bedarfsmeldungen* zu übersetzen. Für die Übersetzung wurde die Python Bibliothek *deep-translator* verwendet. Diese bietet Implementationen unterschiedlicher Übersetzungs-APIs von diversen Anbietern. Der Vorteil ist dabei die vereinfachte Möglichkeit Anbieter bei Bedarf zu wechseln. Es wurde sich für den Google Translator entschieden, da hierfür kein API-Key be-

nötigt wird. Die Methode erhält einen Text als Parameter. Der Google Translator erhält die Parameter *source* und *target*, beidem *source* angibt in welcher Sprache der Eingabetext ist. Durch Angabe von *'auto'* wird die Sprache ermittelt. Der Grund dafür ist, dass grundsätzlich anderssprachige *Bedarfsmeldung* enthalten sein können. Der Parameter *target* ist die Zielsprache in welche der Input übersetzt werden soll. Die Zielsprache ist hier Englisch (*'en'*).

NER

Um die Methode *NER* zu Implementieren wurde die Bibliothek *spaCy* und das Modell *en_core_web_sm* verwendet. Die zu extrahierende Kategorie ist Datum (DATE). Innerhalb der Methode `useNER()` wird der Volltext als Parameter übergeben und zu Tokens umgeformt. Anschließend werden alle Tokens durchlaufen und nach ihren Kategorien überprüft. Ist ein Token die definierte Kategorie, wird der Token in eine separate Liste gespeichert und zurückgegeben. Die extrahierten Tokens werden aus dem Volltext entfernt, um am ende keine doppelten Stichpunkte zu erhalten.

Preprocessing

Vor der weiteren Nutzung der Daten innerhalb einer *Bedarfsmeldung*, ist es erforderlich diese von irrelevanten Wörtern, Zeichen und Formatierungen zu befreien. Zur Eliminierung wiederaufgetretener Wörter, die keine Relevanz für den Informationsgehalt aufweisen, wurde die Bibliothek *nlTK* verwendet. Diese beinhaltet eine Liste an sogenannten *stopwords*. Es werden alle mit einem Leerzeichen getrennten Wörter aus der übergebenen *Bedarfsmeldung* in einer Liste aufgeteilt. Dabei wird jeder Listeneintrag mit der *stopword*-Liste von *nlTK* verglichen. Stimmt das Wort nicht mit einem Eintrag der *stopwords* überein, wird diese in die Liste *words* hinzugefügt. Zum Schluss werden die Wörter wieder zu einem String zusammengetragen und zurückgegeben.

Um weitere Formatierungen und ungewünschte Zeichen zu entfernen wird die Bibliothek *re* verwendet. Diese kann Regular Expression-Patterns anwenden und Bereiche, die zum Pattern passen entfernen. Die Methode `removeTags()` erhält die Expression `</?.*?>`. Dabei werden `<Tags>` ermittelt und mit der Methode `sub()` entfernt. Zur Entfernung von Ziffern und Nicht-Alphanummerischen Zeichen wird die Expression `(\\d|\\W)+` angewendet. Als Ergebnis des Preprocessing wird ein gesäuberter String ohne Zeichen und Tags zurückgegeben. Zum schluss werden alle Wörter in Kleinbuchstaben umgewandelt. Der Grund dafür ist, dass somit die Vergleichbarkeit der Wörter gefördert wird. Es könnte sonst vorkommen, dass Beispielsweise beim Schlüs-

selwörterabgleich zwei Wörter nicht als identisch identifiziert werden, da sie einmal mit großem und einmal mit kleinem Anfangsbuchstaben geschrieben wurde.

TF-IDF

Vorbereitend für die *TF-IDF*-Methode wurden die *Skills* und *Aufgaben* aus allen *Bedarfsmeldungen* ins Englische übersetzt, Preprocessed und in einem String zusammengefasst. Der Grund dafür ist, dass für die *TF-IDF*-Methode ein Textkorpus benötigt wird, woraus die Schlüsselwörter durch *Term Frequency* ermittelt werden. Damit dieser Prozess nur einmal erfolgen muss, wurden die Ergebnisse in die `requirements/preprocessedRequirements.json` gespeichert. Für die Implementierung des *TF-IDF* wurden die Bibliotheken *sklearn* und *numpy* verwendet. Der Textkorpus wird dem *TfidfVectorizer* von *sklearn* beigefügt und die *TF-IDF*-Werte werden berechnet. Anschließend werden alle durchschnittlichen *TF-IDF*-Werte für jedes Wort im gesamten Textkorpus berechnet. Daraus wird eine Liste mit Wörtern und ihren durchschnittlichen *TF-IDF*-Werten erstellt und in absteigender Reihenfolge sortiert. Durch ein vordefinierten Score Threshold können darunterliegende Schlüsselwörter entfernt werden. Dies ist relevant, um Wörter mit niedrigem Scoring und diese beispielsweise nur einmal im Textkorpus auftauchen nicht als Schlüsselwörter erfasst werden. Als Ergebnis wird eine Liste mit Schlüsselwörtern zurückgegeben.

N-Gramm

Damit wie in Kapitel 3.2 Wortketten gebildet werden können ist es notwendig *bi-Gramme* zu verwenden, da somit immer zwei nebeneinanderstehenden Wörter betrachtet werden können. Bei einem höheren n der *n-Gramme* würde somit das Prinzip einer Wortkette, wie sie im System benötigt wird nicht funktionieren. Die Methode der *n-Gramme* wurde mit der Bibliothek *nlTK* implementiert. Eine Variable n ist definiert, die die Größe eines *n-Gramms* widerspiegelt. Da in der Pipeline *Bi-Gramme* benötigt werden, liegt der Wert von n auf 2. Mit der Methode `ngrams()` können die *n-Gramme* generiert werden. Diese erhalten einen *String* und die Variable n als Parameter. Der zu überführende Text wird als Parameter übergeben und durch die Methode `split()` in einer List auf die einzelnen Wörter umgeformt. Die einzelnen Tupel mit den *Bi-Grammen* werden am Ende zurückgegeben.

Entfernung von Bi-Grammen ohne Schlüsselwörter

Zur Entfernung von Bi-Gramm-Tupel, die keine Schlüsselwörter enthalten wurde die Methode `containsKeywords()` erstellt.

```

1  def containsKeywords(biGramList, keywordsList):
2      filteredBiGram = []
3      for tupel in biGramList:
4          if any(word in tupel for word in keywordsList):
5              filteredBiGram.append(tupel)
6      return filteredBiGram

```

Listing 5.1: Implementation der Filterung für Schlüsselwörter in einer Bi-Gramm Liste

Das Listing 5.1 implementiert diese Methode. Als Parameter wird die *Bi-Gramm*-Liste und die Schlüsselwörterliste übergeben. In Zeile 3 ist zu sehen, dass jeder Tupel traversiert wird. Dabei wird in Zeile 4 jedes Wort im Tupel mit der Schlüsselwörterliste verglichen. Wenn das aktuelle Wort in der Liste enthalten ist, so wird dieser Tupel in eine neue Liste aufgenommen. Diese wird am Ende zurückgegeben, wodurch eine gefilterte Liste mit ausschließlich enthaltenen Schlüsselwörtern vorhanden ist.

POS-Tagging

In der Englischen Sprache existieren verschiedene Wortgruppenkombinationen, die zusammen ungewöhnlich klingen und dadurch beim sprechen und schreiben nicht oder nur selten verwendet werden. Wenn diese Kombinationen in den *Bi-Grammen* identifiziert und entfernt werden, können somit Verkettungen unterbrochen werden. Dadurch stehen Wörter und spätere Stichpunkte nicht nebeneinander, die sprachlich im betrachteten Kontext wenig Sinn machen. *Bedarfsmeldungen* wurden im professionell seriösen Kontext angefertigt, wodurch einige nebeneinanderstehenden Wörter im Englischen nicht häufig zusammen auftauchen.

Wortarten	POS-Tagging Abkürzungen	Beispiel
Nomen + Adjektiv	NN JJ	"time beautiful"
Verb + Adjektiv	VB JJ	"think beautiful"
Verb + Pronomen	VB PRP	"think him"
Verb + Verb	VB VB	"think understood"
Adjektiv + Adjektiv	JJ JJ	beautiful strange
Adjektiv + Verb	JJ VB	beautiful think
Adjektiv + Adverb	JJ RB	"beautiful fast"
Adverb + Adjektiv	RB JJ	"fast beautiful"

Tabelle 5.2.: Untypische Wortartenkombinationen in der englischen Sprache.

In der Tabelle 5.2 sind einige Beispiele für solche untypischen Zusammensetzungen. In Spalte eins sind die jeweiligen Wortarten beschrieben. Die zweite Spalte zeigt die

dazugehörigen Tags, wie diese mit *POS-Tagging* ermittelt werden. Die dritte Spalte zeigt jeweils ein Beispiel dieser Wortkombinationen. Die Identifikation erfolgte durch Betrachtung der Grundregeln der Englischen Grammatik aus der Arbeit “Basic English: A general introduction with rules and grammar”[24] und den Syntaxregeln aus der Arbeit “Syntax parsing: Implementation using grammar-rules for English language”[34]. Diese Regeln beinhalten Kombinationen aus Wortarten, die zusammen Erlaubt sind. Dadurch konnten Kombinationen ausgeschlossen werden. Schließlich wurden durch trial and error weitere Kombinationen identifiziert, die schließlich in die Tabelle aufgenommen wurden. Es kann nicht ausgeschlossen werden, dass weitere Kombinationen und Sonderfälle existieren. Dies stellt ein Optimierungspotenzial außerhalb dieser Ausarbeitung dar. Somit werden nur die Kombinationen aus der Tabelle 5.2 im System beachtet. Dennoch ist dieses so konzipiert, dass weitere Kombinationen in das System hinzugefügt werden können.

Zur Implementierung der Methode *POS-Tagging* wurde die Bibliothek *nlTK* verwendet. Die *Tags* werden mit dem vortrainierten Modell *averaged_perceptron_tagger* ermittelt. Als Parameter wird ein String übergeben, der zwei Wörter aus den *Bi-Grammen* enthält. Dieser String wird in Tokens umgewandelt. Diese werden in die Methode *pos_tag()* übergeben und eine Liste mit Tupeln wird zurückgegeben, bei dem das Wort und das dazugehörige *Tag* enthalten ist.

Entfernung von Bi-Grammen mit untypischen Wortartenkombinationen

Die *Bi-Gramm*-Liste muss im nächsten Schritt weiter reduziert werden. Hierbei werden alle *POS-Tagging*-Kombinationen aus der Tabelle 5.2 mit den Tupeln aus der *Bi-Gramm*-Liste verglichen werden.

```

1 combinationsToRemove = ["NN JJ", "VB JJ", "VB PRP", "VB VB", "JJ
    JJ", "JJ VB", "JJ RB", "RB JJ"]
2 def removeWordCombination(biGramList):
3     filteredList = []
4     for left, right in biGramList:
5         biGramString = f"{left} {right}"
6         biGramStringsWithTag = usePosTagging(biGramString)
7         tagString = ' '.join(tag for _, tag in biGramStringsWithTag)
8         if tagString not in combinationsToRemove:
9             filteredList.append((left, right))
10    return filteredList

```

Listing 5.2: Implementation der Filterung von Wortartenkombinationen

Das Listing 5.2 implementiert diesen Schritt. Dazu werden in Zeile 1 alle Kombinationen als String in eine Liste zwischengespeichert. Als Parameter wird die zu filternde

Bi-Gramm-Liste übergeben. Diese wird in Zeile 4 traversiert und die Tupel mit den beiden Wörtern aus den *Bi-Grammen* wird in Zeile 5 als String zusammengefügt. Dieser String wird in die *POS-Tagging*-Methode übergeben und als Rückgabewert erhält das System eine Liste mit zwei Tupeln. Diese Tupel beinhalten jeweils eines der Wörter und das dazugehörige Tag aus dem *POS-Tagging*. Die beiden Tags werden in Zeile 7 zusammengefügt, sodass ein String entsteht, wie es in der *combinationsToRemove*-Liste in Zeile 1 ist. In Zeile 8 wird überprüft ob die Kombination aus Tags einer aus der Liste in Zeile 1 entspricht. Ist dies nicht der Fall, wird der Tupel in eine neue Liste eingefügt, die am Ende als Ausgabe zurückgegeben wird.

Überführung von Bi-Grammen in Stichpunkte

Da der Volltext aus den Bedarfsmeldungen nicht einzelne Wörter, sondern inhaltlich aufeinander bezogene Stichpunkte entstehen sollen, werden die um die Schlüsselwörter herumliegenden Wörter aneinander gefügt. Dazu werden diejenigen *Bi-Gramme* zu einem Satz zusammengefügt, die im ersten Tupel auf der rechten Seite und im zweiten Tupel auf der linken Seite das gleiche Wort enthalten. Hinzu kommt eine Überprüfung ob das darauffolgende linke Wort ein Schlüsselwort ist. Somit werden nur die Wörter zu Sätzen verkettet, die aus Schlüsselwörtern bestehen. In dem Fall, bei dem im darauf folgenden Tupel kein Schlüsselwort auf der linken sondern nur auf der rechten besteht, entsteht ein neuer Stichpunkt. Als Beispiel können folgende *Bi-Gramme* betrachtet werden: (“expertise“, “aws“) (“aws“, “technologie“) (“technologie“, “pflicht“) (“pflicht“, “englisch“) (“englisch“, “skills“). Als Schlüsselwörter wurden die Wörter “aws“, “pflicht“, “englisch“ ermittelt. Die Idee ist es, die zusammengehörigen *Bi-Gramme* an diesen Schlüsselwörtern zusammenzufügen. Daraus werden die Stichpunkte “expertise aws technologie“ und “technologie pflicht englisch skills“.

```

1  def combineWords(filteredBiGram, keywords):
2      combinedWords = []
3      i = 0
4      while i < len(filteredBiGram):
5          keyPhrase = recursiveCombine(filteredBiGram, i, "", keywords,
6                                     "")
7          combinedWords.append(keyPhrase)
8          wordsCount = len(keyPhrase.split(" "))
9          wordsCount = max(1, wordsCount - 1)
10         i += wordsCount
11         if(i > len(filteredBiGram)):
12             break
13     return combinedWords

```

```

14 def recursiveCombine(filteredBiGram, index, currentString,
15     keywords, lastRightItem):
16     left, right = filteredBiGram[index]
17     if not currentString:
18         currentString = left
19     else:
20         currentString += " " + left
21     if lastRightItem in {left, ""}:
22         if right in keywords and index + 1 < len(filteredBiGram):
23             currentString = recursiveCombine(filteredBiGram, index + 1,
24                 currentString, keywords, right)
25         else:
26             currentString += " " + right
27     return currentString

```

Listing 5.3: Umformung der Bi-Gramm Liste in Stichpunkte

Die Implementierung zu dieser Idee ist im Listing 5.3 dargestellt. Die Methode `combineWords` in Zeile 1 erhält die *Bi-Gramm*-Liste und die Schlüsselwörterliste als Parameter. Die *Bi-Gramm*-Liste wird traversiert und ein string aus zusammengehörigen Wörtern wird rekursiv in Zeile 5 mit der Methode `recursiveCombine` erstellt. Dazu wird in jedem rekursiven Schritt überprüft, ob das darauf folgende rechte Wort ein Schlüsselwort und das gleiche wie das linke Wort im aktuellen Tupel entspricht. Ist dies der Fall werden die *bi-Gramm* an der Stelle zusammengefügt. Dies wiederholt sich bis dieser Fall nicht mehr eintritt. Dadurch ist die Rekursion vorbei und wird in Zeile 7 bis 11 geprüft wie viele Wörter in einem String gelandet sind. Das ist wichtig um herauszufinden an welcher Stelle in der *bi-Gramm*-Liste weitergearbeitet werden muss. Sind alle *bi-Gramme* zusammengesetzt worden, wird eine Liste mit allen zusammengesetzten Wörtern zurückgegeben.

6. Evaluierung

Das vorliegende Kapitel befasst sich mit der Evaluierung des entwickelten Systems zur Strukturierung von *Bedarfsmeldungen*. In dem Kapitel 4 wurde dargelegt, dass gängige Methoden zur Evaluierung von den einzelnen Information Retrieval Ansätzen und daraus entstehenden Pipelines und Hybriden die Methoden *Precision*, *Recall* und *F1-Score* angewendet werden. Grundsätzlich wäre dies ein valider Ansatz zur Evaluation. Dennoch bestehen keine Grundvoraussetzungen zur Durchführung dieser Methoden. Es liegt kein Datensatz mit ausreichenden Trainings und Testdaten vor. Die Erstellung eines Datensatzes auf Basis der originalen *Bedarfsmeldungen* ist zeitaufwändig und erfordert Präzision. Dies ist im Rahmen der Ausarbeitung nicht weiter möglich. Stattdessen wird eine optimierende Evaluation auf Basis einer vordefinierten Erwartungshaltung durchgeführt, um Verbesserungspotenzial innerhalb des Systems zu ermitteln. Durch Modifikationen der Parameter der Pipeline wird versucht, eine vordefinierte Erwartungshaltung zu approximieren.

6.1. Versuchsdurchführung

Der nachfolgende Versuch dient der Überprüfung der Ergebnisqualität des entwickelten Systems in Bezug auf die Strukturierung eines Volltextes in reduzierte Stichpunkte. Dazu wird eine synthetische Eingabe eines Skillfeldes einer *Bedarfsmeldung* erstellt. Aus dieser wird manuell die wichtigsten Punkte auf Basis der Erkenntnisse aus Kapitel 2 herausgearbeitet. Dieses dient als Erwartungshaltung, zu dem das System annäherungsweise gelangen soll. Zum semantischen Ähnlichkeitsvergleich wird die Kosinus-Ähnlichkeit verwendet. Dabei werden die Stichpunkte des Systems und der Erwartungshaltung vektorisiert und durch die Berechnung des *Cosine-Similarity* ein Ähnlichkeitsscoring zwischen 0 und 1 ermittelt. Dabei spiegelt der Wert 0 eine niedrige und 1 eine hohe Ähnlichkeit wider. Das Ziel ist es die Schrittweise Anpassung der Parameter, den Ähnlichkeitswert näher zu 1 zu bringen.

Word Embedding

Word Embedding sind Vektordarstellungen eines Wortes, die durch Training eines neuronalen Netzes auf einem großen Korpus gewonnen werden [31]. Sie finden häufig Anwendung bei der Klassifikation von Texten anhand semantischer Ähnlichkeit [31]. Word2vec stellt eine der am häufigsten verwendeten Formen von *Word Embedding* dar [31]. Das Word2Vec-Modell nimmt einen Textkorpus als Eingabe und erzeugt

Wortvektoren als Ausgabe, die anschließend für die Klassifizierung eines beliebigen anderen Wortes verwendet werden können [31]. Dazu wird der entsprechende Vektorwert ermittelt [31]. Im Rahmen des Experiments wurde ein vortrainiertes *Word Embedding*-Modell namens *ConceptNet Numberbatch* [32] verwendet, um Vektoren für den Eingabetext zu erstellen.

Cosine-Similarity

Cosine-Similarity misst den Kosinus des Winkels zwischen zwei Vektoren, die auf eine mehrdimensionale Ebene projiziert werden [31]. Es stellt eine weit verbreitete Metrik im Bereich des Information Retrieval dar [28]. Die Metrik modelliert ein Textdokument als einen Vektor von Begriffen [28]. Die Berechnung des Kosinuswerts zwischen den Vektoren zweier Dokumente erlaubt die Ermittlung der Ähnlichkeit zwischen diesen Dokumenten [28]. Der Grad der Ähnlichkeit zwischen den Vektoren ist ein Indikator für die Relevanz zwischen den Texten [28].

Optimierungsmaßnahmen

Im Rahmen der Evaluierung erfolgt eine schrittweise Anpassung der vier Parameter des Systems. Im Anschluss an jede Anpassung erfolgt eine Evaluierung der *Cosine-Similarity* im Hinblick auf eine etwaige Verbesserung oder Verschlechterung im Vergleich zur Erwartungshaltung. Im Anschluss an die Ermittlung des höchsten Scores werden die Ergebnisse dargestellt und Auffälligkeiten diskutiert. Zum Schluss jedes Schrittes erfolgt eine feste Übernahme des betreffenden Parameters in das System. In der Folge wird der darauffolgende Parameter angepasst. Die vier zu evaluierenden Parameter sind wie folgt definiert:

1. Korpusgröße der Schlüsselwortextraktion

In diesem Zusammenhang erfolgt eine Anpassung der Menge an *Bedarfsmeldungen* als Korpus für die Schlüsselwortextraktion. In einem nächsten Schritt werden neben den *offenen* und *eskalierten Bedarfsmeldungen* auch alle bereits *geschlossenen* Bedarfsmeldungen in den Korpus aufgenommen. Dies resultiert in einer signifikanten Erhöhung der Wortmenge innerhalb von *TF-IDF*. Im Rahmen der Evaluierung wird untersucht, ob die Erweiterung des Korpus einen positiven Effekt auf die Identifikation der Schlüsselwörter hat.

2. Scorethreshold

Innerhalb der *TF-IDF* erfolgt eine Anpassung des Scorethresholds zur Ausschließung von Schlüsselwörtern mit geringem Score. Der Score wird zunächst mit einem niedrigen Wert (nahe 0) getestet und anschließend schrittweise erhöht.

3. POS-Tagging Kombinationen erlauben

Im Rahmen dieses Prozesses erfolgt eine schrittweise Freigabe von Wortartenkombinationen, die als untypisch definiert wurden. Im Rahmen dieses Prozesses wird in jedem Schritt eine Kombination aus dem System entfernt, bis schließlich keine Kombinationen mehr enthalten sind.

4. Übersetzung beibehalten

Im Rahmen der Untersuchung werden die einzelnen Übersetzungsschritte entfernt und deren Einfluss auf die Ergebnisqualität evaluiert.

Systemspezifikation

Im Rahmen der Evaluation wurde ein System eingesetzt, dessen Spezifikationen wie folgt definiert sind: Prozessor: 13th Gen Intel(R) Core(TM) i7-1360P, 2200 MHz, 12 Kerne, 16 logische Prozessoren, RAM: 16 GB

6.2. Beschreibung des verwendeten Datensatzes

In diesem Kapitel wird zunächst eine Ausgangslage konstruiert, auf deren Basis anschließend eine Erwartungshaltung definiert wird. Die Berechnung der Kosinus-Ähnlichkeit erfolgt mit dem Ziel, eine Verbesserung der textuellen Ähnlichkeit festzustellen.

Ausgangslage

Die Ausgangslage bildet ein synthetischer Datensatz. Die Grundlage hierfür bildeten authentische *Bedarfsmeldungen*. Bei der Erstellung wurde darauf geachtet, dass der Datensatz einer echten *Bedarfsmeldung* ähnelt. Der Ausgangsdatenbestand liegt als Volltext vor.

Aufgabe ist es, beim Aufbau einer Plattform zu unterstützen, die auf Basis von Azure, OpenAI, Java und ReactJS aufgebaut ist. Aktuell (und perspektivisch) haben wir einen Engpass vor allem im Frontend. Hilfe im Backend (Java) wäre aber auch willkommen.\r\nDas Projekt bietet einiges an Potenzial, d.h. wir erwarten hier noch einige Interessenten für die Plattform. 2 Jahre Erfahrung und Kenntnisse in der Software-Entwicklung mit Java sind erwünscht

Der Volltext weist eine Vielzahl unterschiedlicher Zeichen, Formatierungen (beispielsweise \r \n für Zeilenumbrüche) sowie zeitbezogene Daten auf.

Erwartungshaltung

Die Erwartungshaltung wurde manuell aus der Ausgangslage abgeleitet. Bei der Erstellung wurde darauf geachtet, dass Füllwörter entfernt und wichtige Informationen beibehalten wurden, die in Kapitel 2 erfasst wurden. Das Ziel besteht in der Annäherung an die Erwartungshaltung.

- Aufbau einer Plattform
- Azure, OpenAI, Java und ReactJS
- Engpass im Frontend
- Hilfe im Backend (Java)
- bietet Potenzial
- erwarten Interessenten für Plattform.
- 2 Jahre Erfahrung in Java

6.3. Präsentation und Diskussion der Ergebnisse

Im Folgenden werden die Resultate der einzelnen Anpassungsschritte, wie sie in Kapitel 6.1 beschrieben sind, dargelegt.

1. Korpusgröße der Schlüsselwortextraktion

Die These besagt, dass mit einer Zunahme der im Textkorpus enthaltenen Wörter und Daten für die Schlüsselwortextraktion eine Steigerung der Qualität der Schlüsselwörter einhergeht. Diese Annahme basiert auf der Prämisse, dass eine höhere Anzahl an themenbezogenen Dokumenten, die häufig verwendete fachliche Ausdrücke enthalten, zu einer höheren Gewichtung dieser Ausdrücke bei der Term-Frequency-Berechnung führt. Dies wiederum resultiert in einer höheren Wahrscheinlichkeit, dass sie als potenziell wichtige Schlüsselwörter identifiziert werden.

Die zuvor aufgestellte These soll durch eine schrittweise Erhöhung der Anzahl der im Textkorpus enthaltenen Wörter überprüft werden. Dazu wird die *TF-IDF*-Methode eingesetzt. Die Bedarfsmeldungen mit den Status *offen*, *eskaliert* und *angeboten* wurden gemäß der Vorverarbeitung in Kapitel 5.3.3 zu Textblöcken mit Wörtern zusammengefasst. Die ersten zehn Wörter der *offenen Bedarfsmeldungen* dienen als Grundlage für die weitere Analyse. Im Rahmen der Analyse werden den *Bedarfsmeldungen* sukzessive mehr Wörter hinzugefügt, um die Veränderung des Scores zu beobachten. Bei der Durchführung ist der Scorethreshold bei 0.0, um Einflüsse vor den Anpassungen in Schritt 2 zu vermeiden.

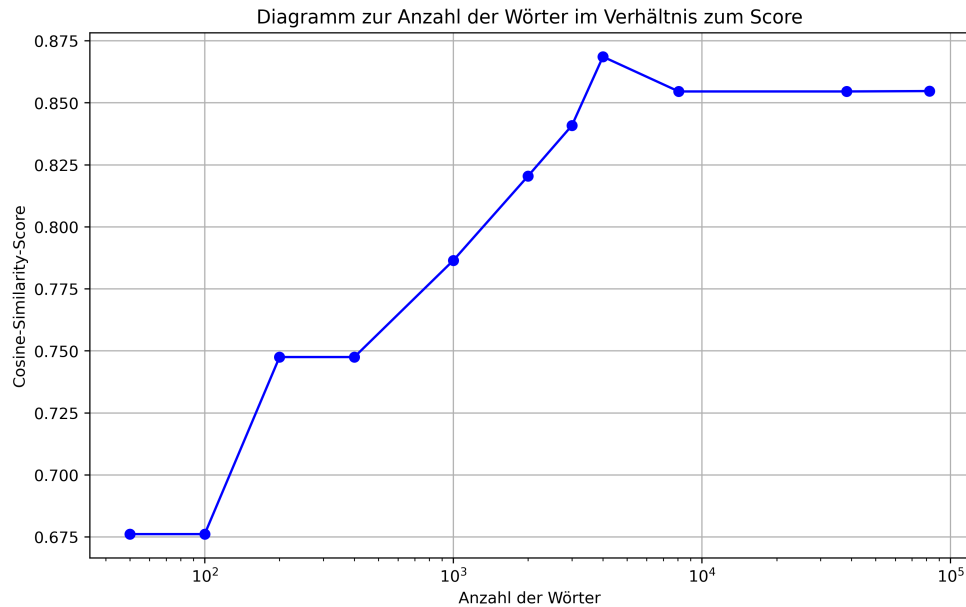


Abbildung 6.1.: Diagramm zur Anzahl der Wörter im Verhältnis zum Score.

In der Abbildung 6.1 wird das Verhältnis der Anzahl an Wörtern zum Score in Form eines Diagramms dargestellt. Im halblogarithmischen Diagramm lässt sich ein annähernd linearer Zusammenhang zwischen dem Score und der Anzahl der Wörter erkennen. Dies impliziert, dass der Score proportional zur exponentiellen Anzahl der Wörter ist. Der beschriebene Effekt ist bis zum Erreichen eines Plateaus bei 80.000 Wörtern erkennbar. In der Konsequenz lässt sich ein Anstieg der Ergebnisqualität bis zu einem gewissen Punkt feststellen, je mehr Wörter im Textkorpus enthalten sind. Eine Überschreitung dieses Punktes führt zu einer Stagnation der Ergebnisqualität. Bei einer zu geringen Anzahl an Wörtern im Textkorpus ($n < 10$) konnten keine Scoring-Werte generiert werden. Die Generierung von Stichpunkten durch das System ist nur möglich, wenn die entsprechenden Schlüsselwörter vorhanden sind, da andernfalls keine Zuordnung erfolgen kann.

Anzahl Wörter: 4000	Anzahl Wörter: 8070
<ul style="list-style-type: none"> • die Aufgabe Unterstützung Entwicklung Plattform basierend auf Azure OpenAI • OpenAI, Java, ReactJS • reactjs Zukunft • zukünftiger Engpass • Engpass vor allem Frontend Hilfe Backend Java wäre willkommen • begrüßen das Projekt viel Potenzial dh • d.h. interessiert • Interessenten Plattform Erfahrung Kenntnisse Software Entwicklung Java erwünscht • 2 Jahre 	<ul style="list-style-type: none"> • die Aufgabe Unterstützung Entwicklung Plattform basierend auf Azure OpenAI • OpenAI, Java, ReactJS • reactjs Zukunft • zukünftiger Engpass • Engpass vor allem Frontend Hilfe Backend Java wäre willkommen • begrüßen das Projekt viel Potenzial dh interessiert • Interessenten Plattform Erfahrung Kenntnisse Software Entwicklung Java erwünscht • 2 Jahre

Tabelle 6.1.: Gegenüberstellung der Ergebnisse mit 4000 und 8070 Wörtern im Textkorpus.

In der Tabelle 6.1 erfolgt eine Gegenüberstellung der Ergebnisse bei einer Wortanzahl von 4.000 bzw. 8.0070 Wörtern. In der linken Spalte findet sich das Resultat mit dem höchsten Score. Das darauffolgende Ergebnis in der rechten Spalte ist Bestandteil des beobachteten Plateaus. Der Unterschied zwischen den beiden Resultaten manifestiert sich in der linken Spalte, Punkt 7, beim Stichwort *d. h. interessiert*. Dieses ist auf der rechten Seite der Tabelle in Punkt 6 zu finden. Der Unterschied ist marginal, sodass der Beginn der Plateaus ab 8070 Wörtern als Anhaltspunkt für die weitere Evaluation verwendet werden kann.

2. Scorethreshold

Die vorliegende These beschreibt, dass die Qualität von Schlüsselwörtern durch einen Threshold-Parameter optimiert werden kann. Wörter, die den Threshold unterschreiten, werden bei der Ermittlung der Schlüsselwörter nicht berücksichtigt, wodurch Wörter mit einem niedrigen Scoring keine Berücksichtigung finden. Ein niedriger Score kann darauf hinweisen, dass ein Wort lediglich einmalig im Kontext vorkommt und folglich kein Schlüsselwort darstellt.

Die zuvor aufgestellte These soll im Folgenden einer Überprüfung unterzogen werden. Zu diesem Zweck werden die Threshold-Werte in Intervallen von 0,0 bis 1,0 in Schritten von 0,01 durchlaufen.

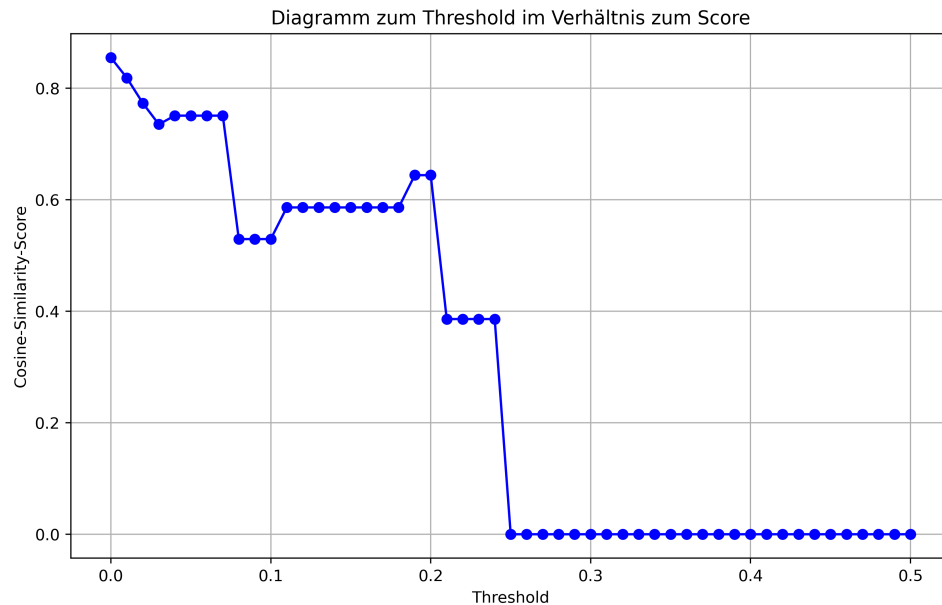


Abbildung 6.2.: Diagramm zum Threshold im Verhältnis zum Score.

In der Abbildung 6.2 ist das Diagramm dargestellt, das die Thresholds im Verhältnis zum Score widerspiegelt. Die dargestellten 0,01-Schritte genügen, um einen Abwärtstrend zu identifizieren. Die aufgestellte These ist als falsch zu erachten, da bei einer Erhöhung des Thresholds der Score annähernd linear sinkt. Dies geschieht bis zum Threshold-Wert 0.24. Ein darüber hinausgehender Wert entfernt zu viele Wörter aus der Schlüsselwortermittlung, wodurch keine Stichpunkte gebildet werden und der Score den Wert 0 annimmt. Der höchste Thresholdwert des Diagramms der Abbildung 6.2 wurde auf den Wert 0.5 gesetzt, da alles darüber hinaus ebenfalls den Similarity-Score 0 annehmen.

Threshold: 0.0	Threshold: 0.05
<ul style="list-style-type: none"> • die Aufgabe Unterstützung Entwicklung Plattform basierend auf Azure OpenAI • OpenAI, Java, ReactJS • reactjs Zukunft • zukünftiger Engpass • Engpass vor allem Frontend Hilfe Backend Java wäre willkommen • begrüßen das Projekt viel Potenzial dh interessiert • Interessenten Plattform Erfahrung Kenntnisse Software Entwicklung Java erwünscht • 2 Jahre 	<ul style="list-style-type: none"> • die Aufgabe • Entwicklungsplattform zur Aufgabenunterstützung • OpenAI, Java, ReactJS • Backend Java würde • begrüßen das Projekt Los • Plattformerfahrung Kenntnisse Softwareentwicklung Java erwünscht • 2 Jahre

Tabelle 6.2.: Gegenüberstellung der Ergebnisse mit dem Threshold 0.0 und 0.05.

Eine manuelle Betrachtung der Ergebnisse hat gezeigt, dass ohne die Anwendung eines Thresholds eine deutliche Zunahme der Stichpunktlänge von 2-5 Wörtern pro Stichpunkt zu 2-9 Wörtern pro Stichpunkt zu verzeichnen ist. Dies ist in Tabelle 6.2 durch die Ergebnisse mit den Thresholds 0.0 und 0.05 dargestellt. Die Inhalte der Stichpunkte sind von hoher Relevanz, was zu einem entsprechend hohen Score führt. Dennoch kann ein geringer Threshold einen positiven Effekt auf die Stichpunkte ausüben auch wenn der Score dadurch leicht abfällt. Der direkte manuelle Vergleich hat ergeben, dass für die ausgewählte Ausgangslage ein Score im Bereich von 0.04 bis 0.07 die Stichpunkte verkürzt werden, ohne die wichtigen Informationen zu beeinträchtigen. Somit kann für die weitere Evaluation der Wert 0.05 aus der Gegenüberstellung weiter verwendet werden.

3. POS-Tagging Kombinationen erlauben

Die hier vorgestellte These besagt, dass die Qualität der Stichpunkte durch die Entfernung von untypischen Wortkombinationen beeinträchtigt wird. Durch fehlende Entfernung untypischer Wortkombinationen könnten Stichpunkte entstehen, die inhaltlich eine geringere Kohärenz beinhalten und dadurch auch einen geringeren Score erhalten.

Die in Kapitel 5.3.3 präsentierte Liste mit POS-Tagging-Kombinationen wird im Folgenden schrittweise modifiziert, um die zugrundeliegende These zu überprüfen.

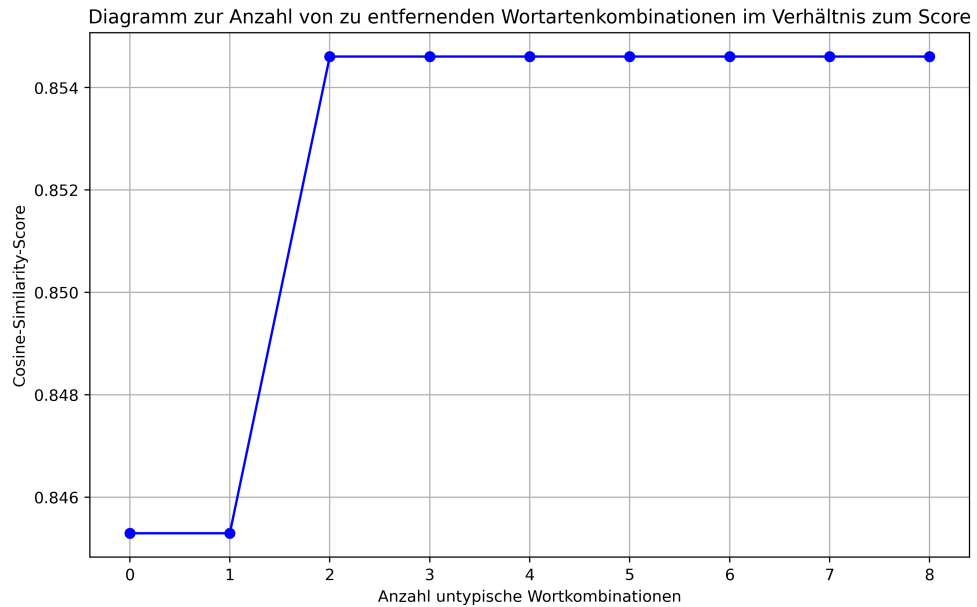


Abbildung 6.3.: Diagramm zur Anzahl von zu entfernenden Wortartenkombinationen im Verhältnis zum Score.

In der Abbildung 6.3 ist das Diagramm zur Anzahl der zu entfernenden Wortartenkombinationen im Verhältnis zum Score dargestellt. Die Werte auf der X-Achse quantifizieren die Anzahl der enthaltenen Wortkombinationen. Der Wert 0 entspricht der Durchführung des Systems ohne den Schritt der Entfernung von untypischen Wortkombinationen. Der Wert 8 stellt demgemäß den Schritt dar, bei dem sämtliche acht Kombinationen aus der Tabelle 5.2 aus dem Kapitel 5.3.3 in die Auswertung einbezogen sind. Die Werte dazwischen repräsentieren die jeweilige Anzahl der zu dem Zeitpunkt enthaltenen Kombinationen.

Anz. untypischer Wortkombinationen: 1	Anz. untypischer Wortkombinationen: 2
<ul style="list-style-type: none"> • die Aufgabe • Entwicklungsplattform zur Aufgabenunterstützung • OpenAI, Java, ReactJS • Backend Java würde begrüßen das Projekt Los • Plattformerfahrung Kenntnisse Softwareentwicklung Java erwünscht • 2 Jahre 	<ul style="list-style-type: none"> •

Tabelle 6.3.: Gegenüberstellung der Ergebnisse mit 1 und 2 untypischen Wortkombinationen.

Es lässt sich erkennen, dass sich der Score von 0,84529346 auf 0,85460645 verbessert hat, nachdem im Schritt 2 die zu entfernende Kombination aus Verb und Adjektiv hinzugefügt wurde. Dies belegt, dass die aufgestellte These korrekt ist und das System durch die Entfernung untypischer Wortkombinationen ein besseres Ergebnis liefert. **ToDo:** nochmal gucken. Kann die Werte nicht mehr reproduzieren

<-hier

4. Übersetzung beibehalten

Die These lautet, dass die implementierten Ansätze ohne eine Übersetzung ins Englische zu schlechteren Ergebnissen führen. Die These basiert auf der Annahme, dass die Modelle mit englischen Datensätzen trainiert wurden und daher bei der Anwendung auf deutsche Wörter möglicherweise nicht die optimalen Ergebnisse liefern.

mit Übersetzung	ohne Übersetzung
Score: 0.75364363	Score: 0.7852247
<ul style="list-style-type: none"> • die Aufgabe • Entwicklungsplattform zur Aufgabenunterstützung • OpenAI, Java, ReactJS • Backend Java würde begrüßen das Projekt Los • Plattformerfahrung Kenntnisse Softwareentwicklung Java erwünscht • 2 Jahre 	<ul style="list-style-type: none"> • openai java und • backend java wäre • mit java sind

Tabelle 6.4.: Gegenüberstellung der Ergebnisse mit und ohne Übersetzungsschritt.

Die nachfolgende Tabelle 6.4 präsentiert eine Gegenüberstellung der Ergebnisse, die mit und ohne Berücksichtigung des Übersetzungsschritts erzielt wurden. In diesem Zusammenhang sind die Möglichkeiten der Kosinusmetrik ausgeschöpft. Obschon bei manueller Betrachtung des Scores ohne die Übersetzung ein höherer Wert registriert wird, lässt sich feststellen, dass die Qualität der Stichpunkte dennoch abnimmt. Eine exakte Erfassung des Grundes ist von außen nur schwer möglich. Es kann angenommen werden, dass bei keiner Übersetzung die Verwendung von Wörtern eine größere Nähe zum Original aufweisen und dadurch zu einer stärkeren Ähnlichkeit führt. Die besseren Ergebnisse durch die Übersetzung kann darauf zurückgeführt werden, dass die Stopwords-Entfernung auf englische Begriffe ausgelegt ist. Dies hat zur Konsequenz, dass diese bei der Entfernung nicht erfasst werden. Des Weiteren ist der Korpus mit englischsprachigen Übersetzungen gefüllt, was eine effiziente Ermittlung der Schlüsselwörter verhindert. Die Stichpunkte im Verfahren ohne die Übersetzung sind kurz und bestehen aus Wörtern, die nach der Übersetzung unverändert bleiben. Zu diesen Wörtern gehören unter anderem Technologien und feststehende Namen. Der aktuelle Aufbau sowie die genutzten Technologien im System benötigen eine Übersetzung, um bessere Ergebnisse zu erzielen. Dennoch könnte die Übersetzung selbst einen Effekt auf die Ergebnisqualität haben. Die Wahl der Übersetzungsschnittstelle basiert auf der öffentlichen Verfügbarkeit des Google Übersetzers. Die Nutzung eines anderen Anbieters, der über die Funktion verfügt, fachlich-softwareentwicklungsspezifische Wörter adäquat zu übersetzen, könnte zu einer präziseren Übersetzung führen. Da der Textkorpus zur Ermittlung der Schlüs-

selwörter mit derselben Methode wie der Volltext vorverarbeitet und übersetzt wurde, sind die Ergebnisse mit dem Google Übersetzer als ausreichend zu betrachten.

Zusätzliche Erkenntnis

Im Rahmen der Evaluierung wurde ersichtlich, dass die eigentlich essenzielle Abfolge des erst Übersetzens und anschließenden Vorverarbeitens eine Problematik generiert. Die ursprüngliche Überlegung zur Reihenfolge des Vorübersetzens basierte auf der Annahme, dass beim Vorverarbeiten Satzzeichen entfernt werden, wodurch die Möglichkeit verloren geht, einzelne Satzkontexte zu identifizieren. Dies würde eine Reduktion der Übersetzungsqualität zur Folge haben. Es hat sich gezeigt, dass Formatierungszeichen wie `\r \n` dazu führen, dass nach der Übersetzung Situationen entstehen, die die ursprüngliche Struktur der Wörter nicht mehr erkennen lassen. So wird aus dem nicht zusammengehörenden Wörtern *Java\r\nSpring* nach der Übersetzung das zusammengesetzte Wort *JavarnSpring*. Diese Problematik ist auf die Funktionsweise der Übersetzungsfunktion zurückzuführen. Um das genannte Problem zu umgehen, wird eine Entfernung der genannten Formatierungen noch vor der Übersetzung durchgeführt, wodurch das Problem gelöst wird.

Des Weiteren wurde festgestellt, dass in einigen Fällen ein Kontext der zeitbezogenen Daten wünschenswert wäre. Die ermittelten Zeiten sind korrekt, allerdings fehlen Angaben dazu, wofür die Zeiten stehen. In dem Datensatz aus der Evaluation werden die 2 Jahre korrekt identifiziert, allerdings fehlt die Information, dass es sich bei den 2 Jahren um Java-Erfahrung handelt.

Es besteht weiterhin Optimierungspotenzial hinsichtlich der Stichpunkte. Auch wenn Anpassungen an den Parametern vorgenommen wurden, weist ein Teil der Ergebnisse unwichtigen Informationsgehalt auf. Auch die Struktur der Stichpunkte weisen untypische und unvollständige Abschnitte auf, wie es im finalen Ergebnis aus der Tabelle 6.4 mit dem Punkt *begrüßen das Projekt Los* zu entnehmen ist. Dennoch werden die wichtigsten Informationen zusammengetragen, sodass der Prototyp einen Schritt in Richtung eines funktionsfähigen Systems darstellt.

7. Zusammenfassung und Ausblick

Die vorliegende Arbeit verfolgt das Ziel, durch die Anwendung von Ansätzen des Information Retrieval semi-strukturierte *Bedarfsmeldungen* in einen überschaubaren und auf die wichtigsten Inhalte reduzierten Informationsgehalt zu überführen. Zu diesem Zweck wurden im Rahmen von Experteninterviews die essenziellen Inhalte einer *Bedarfsmeldung* ermittelt und auf dieser Grundlage eine Vorlage für eine strukturierte Bedarfsmeldung erstellt. Auf Basis der erarbeiteten Ergebnisse wurde ein prototypisches System entwickelt, das die Reduktion von Bedarfsmeldungen sowie die Zusammenfassung unstrukturierter Volltextbereiche auf die wichtigsten Stichpunkte ermöglicht.

Die Evaluierung hat ergeben, dass durch Vorverarbeitungsschritte vor und nach einer Übersetzung der Volltexte eine Ausgangslage geschaffen wird, die beim Abgleich der identifizierten Schlüsselwörter durch *TF-IDF* zur Qualität der Stichpunkte beiträgt. Die Identifikation einer Sammlung von Schlüsselwörtern, deren weitere Reduzierung eine wesentliche Rolle spielt, ist durch einen entsprechend großen Textkorpus möglich, wobei die Größe des Textkorpus ab einem gewissen Punkt keine weitere Rolle spielt. Die Bildung von Wortketten, die auf zusammengehörigen Schlüsselwörtern basieren, ist ein wesentlicher Aspekt des Verfahrens. Die Präzision dieser Wortketten kann durch einen Threshold-Wert für die Schlüsselwörter weiter erhöht werden. Der Schritt der Erkennung untypischer Wortarten konnte bei der Evaluation keinen positiven, aber auch keinen negativen Effekt verzeichnen. Im Rahmen der Tests konnten keine gesuchten Wortkombinationen identifiziert werden, sodass sich das Ergebnis nicht verändert hat. Die Resultate weisen noch Optimierungspotenzial auf. Es zeigt sich, dass nicht jede Information einen relevanten Inhalt widerspiegelt und Stichpunkte teils eine ungewöhnliche Struktur aufweisen.

Das prototypische hybride System stellt ein Proof of Concept dar. Die resultierenden deterministischen Ergebnisse sind vielversprechend und verdienen weitere Untersuchung.

Ausblick

Im Rahmen künftiger Überlegungen könnte die Ermittlung der Schlüsselwörter alternativ mittels eines durch Experten angelegten Katalogs mit aus der Branche ent-

haltenen Schlüsselwörtern erfolgen. Eine Steigerung der Ergebnisqualität ist potenziell zu erwarten, wenn präzisere Schlüsselwörter verwendet werden. Die Evaluation hat ergeben, dass die Kosinusähnlichkeit an ihre Grenzen gestoßen ist, wodurch die Hinzunahme weiterer Metriken zur Evaluierung wichtig ist. Ein weiterer Evaluationsschritt in Bezug auf die NLP-Modelle könnte in einem Austausch der Modellgrößen bestehen. Eine Steigerung der Ergebnisqualität könnte durch den Einsatz von größeren und komplexeren Modellen erzielt werden. Des Weiteren besteht die Möglichkeit, den Übersetzungsschritt durch andere Anbieter durchführen zu lassen, wodurch fachlich korrektere Übersetzungen gewährleistet werden können. Es wäre von großem Interesse, eine direkte Gegenüberstellung mit einem System vorzunehmen, das auf Large Language Models basiert. In dem Aspekt wäre es aufschlussreich, die Ergebnisse zu vergleichen, wie es sich in Bezug auf den Ressourcenverbrauch und die inhaltliche Performance schlägt.

A. Anhang

A.1. Interviewtranskripte

A.1.1. Vorabtest Fragen und Antworten

1. Welche Art von Projekten sind typischerweise in Ihrem Unternehmen an der Tagesordnung? Können Sie uns Beispiele für verschiedene Arten von Projekten geben, die adesso durchführt?

Software-Entwicklungsprojekte, angefangen von Projekten in dem ein adesso in einem Kundenprojekt arbeitet über gemischte Teams aus adesso und Kunde bis hin zur kompletten Lieferung von Projektleitern, Testern, Requirements Engineer und Entwicklern

2. Wie werden Projektbedarfe und -anforderungen innerhalb von adesso typischerweise kommuniziert und dokumentiert?

Initial über den Maitre, der das Staffing übernimmt bzw. auch Vorschläge von Projektleitenden zum Staffing annimmt, teilweise auch über das eigene Netzwerk zwischen Führungskräften, im CC, Bereich oder der LoB. Am Ende über das Staffing Jira

3. Welche Informationen halten Sie in einer Bedarfsmeldung für besonders wichtig oder unverzichtbar?

Senioritätslevel, Tagessatz, Remote/on Site Einsatz, Dauer, Technischer Stack (Muss- und Kann Kriterien), Einarbeitungszeiträume (ist es verrechenbar oder nicht?), Lieferverpflichtung

4. Wie detailliert sollten Bedarfsmeldungen Ihrer Meinung nach sein? Sind bestimmte Schlüsselaspekte oder -informationen in jeder Bedarfsmeldung enthalten?

Es sollte aussagefähig sein zumindest welche technischen Kompetenzen wichtig sind und welche Tagessätze, ob Remote möglich ist und die Dauer mindestens in der Bedarfsmeldung vorhanden sein.

5. Welche Herausforderungen oder Schwierigkeiten sind bei unklaren oder unvollständigen Bedarfsmeldungen aufgetreten?

Der Anforderer muss ggf. Fragen mehrfach beantworten, der Kanal über den kommuniziert wird (Teams Chat, Anruf, im Ticket, ...). Dadurch verliert man ggf. den Überblick

6. Wer sind die typischen Stakeholder bei der Erstellung von Bedarfsmeldungen und welche Rolle spielen sie?

Sales/PL: Anforderer mit den technischen Informationen, Maitre: kümmert sich um das Staffing bzw. die eigentliche Besetzung

7. Wie wird die Qualität von Bedarfsmeldungen bei *adesso* bewertet? Gibt es bestimmte Kriterien oder Standards, anhand derer Bedarfsmeldungen beurteilt werden?

Gar nicht meines Wissens nach.

8. Wie können Sie die Qualität und Klarheit von Bedarfsmeldungen verbessern?

Zukünftig: Durch klarere Vorgaben und weniger Freitext, aktuell: durch Nachfragen und Bitten um nachträgliche Pflege, ggf. durch Reviewprozesse bei eigenen Bedarfsmeldungen

9. Welche Auswirkungen haben unklare oder fehlende Informationen in Projektbeschreibungen auf die Effizienz und den Erfolg von Projekten?

Das Staffing dauert länger und ggf. werden die Stellen durch andere Dienstleister besetzt

10. Wie können Sie sicherstellen, dass die Bedürfnisse und Anforderungen aller relevanten Stakeholder in einer Bedarfsmeldung angemessen berücksichtigt werden?

Gute Abstimmungen bevor die Bedarfsmeldung erstellt wird, ggf. durch ein Quality-Gate (Review).

A.1.2. Interview 2

0:1:13 → 0:1:33

I:

Du bist zum einen CC Leiter. Das heißt du hast auch Menschen unter deiner Leitung, die du auf Projekte zuweist.

0:1:33 → 0:1:29

B:

Genau ja.

0:1:29 → 0:1:37

I:

Du weiß also zu Projekten zu, übernimmst auch die Projektleitung.

0:1:37 → 0:1:38

B:

Ja, genau.

0:1:38 → 0:1:50

I:

Genau da würde ich dann erst mal gerne wissen: Was sind so die typischen Stakeholder bei der Erstellung von Bedarfsmeldungen und welche Rolle hast du dabei grundsätzlich?

0:1:50 → 0:2:41

B:

In den Situationen nehme ich immer die Rolle des Beraters erstmal an. Die Stakeholder sind klassisch die Fachverantwortlichen beim Kunden, aber auch die Entscheider. Sprich also deren Vorgesetzte die quasi fachlich vielleicht das ganze nicht so bewerten können, aber das Budget dafür hergeben müssen und natürlich dann im Zweifelsfall auch CO, CEO oder sogar Geschäftsführer.

0:2:41 → 0:2:48

I:

Kannst du ein paar Beispiele nennen, welche Arten von Projekten adesso so erhält und durchführt.

0:2:48 -> 0:3:50

B:

Ja, also im Prinzip kannst du das in 2 Arten von Projekten teilen. Aus einer sind Time Material Projekte, wo quasi der Kunde mit einer Idee kommt, wo wir gut unterstützen können. Beispielsweise bei Bestandsprojekten. Oder vielleicht weil ihnen selbst die Ressourcen dafür fehlen. Zum anderen hast du halt Festpreis Projekte, wo wir bestimmtes Gewerk für den Kunden abschätzen und das Ganze auch dann gänzlich liefern. In Festpreis Projekten haben wir eher die Staffing Hoheit. Das heißt, wir können entscheiden wen wir in das Projekt einsetzen. Wobei bei einigen Projekten durchaus auch der Kunde sich die Profile mit anschaut und dann auch entscheidet. Anhand von Interviews was macht Sinn, was passt bei mir ins Team vielleicht und denke ich, dass das am besten für mich wäre?

0:3:50 -> 0:4:2

I:

Wie werden diese Bedarfsmeldungen und Anforderungen denn so typischer Weise kommuniziert und dokumentiert? Gibt es da eine Art Ablauf, oder wie wird das gemacht?

0:4:2 -> 0:5:16

B:

Hängt auch immer ehrlicherweise vom jeweiligen Kunden ab. Bei manchen reicht ein Interview, welches du führst und dann schreibst du es in ein Word Dokument als Anforderungsbeschreibung nieder. Dann lässt man das gegen Zeichnen und dann ist gut. Manchmal muss man aber auch ein paar mehr Integration fahren und dann nochmal genau abzustecken, was denn Bestandteil der Beauftragung ist und was nicht. Also was die Bedarfsmeldung. Da muss man auch mal durchaus eins tiefer bohren, weil da teilweise der Gedanke, was der Kunde möchte nicht mit dem übereinstimmt, was eigentlich gebraucht wird. Das ist so, weil du klassisch irgendein Anforderungsdokument dafür fertig machst, wenn das immer ein bisschen höher geht Richtung Management. Beim Management mit dem Kunden kann es auch durchaus mal eine Präsentation sein, wo das Ganze nochmal ein bisschen aufbereitet ist. Ein bisschen klassisch, ein bisschen bunter und mit weniger Infos und mit weniger Tiefe präsentiert.

0:5:16 -> 0:5:38

I:

Welche Informationen sind denn für dich in Bedarfsmeldungen besonders wichtig oder auch unverzichtbar? Also zum Beispiel gibt es ja auch das Senioritätslevel.

0:5:38 -> 0:6:50

B:

Ja genau, also Senioritätslevel ist ehrlicherweise erstmal das zweitrangige. Das müssen wir im Nachgang einmal prüfen. Wichtig ist der Tech Stack und was vom Kunden kommt. Und quasi anhand der Bedarfsmeldung an sich. Wie hoch ist der Aufwand, der dahinter steckt? Sowohl durch den Kunden als auch das, was wir teilweise schätzen. Dann kann durchaus sein, dass der Kunde sagt, was gebraucht wird, er aber eigentlich keine Ahnung hat. Es kann vorkommen, dass der Kunde gerne in 2 Monaten durch ist, aber es keinen Sinn macht und wir mindestens ein halbes Jahr benötigt. Dann trifft man sich irgendwo in der Mitte und muss aber noch abgrenzen was Sinn macht und was nicht. Wenn das alles klar ist, dann kannst du überlegen was du eher brauchst. Machst du z.B. nur was mit Senioren? Oder reicht ein Junior. Das hängt dann immer eher von der Gesamtsituation ab.

0:6:50 -> 0:7:5

I:

Wie detailliert sollte dann eine Bedarfsmeldung sein? Gibt es Aspekte oder Informationen, die eigentlich in jeder Bedarfsmeldung drin sein sollten und müssen?

0:7:5 -> 0:7:24

B:

Ich glaube der Umfang ist immer ganz wichtig. Die Erwartungshaltung sollte immer detailliert sein. Und die Technologien ebenfalls.

0:7:24 -> 0:7:33

I:

Welche Herausforderungen oder oder Schwierigkeiten sind dann bei zum Beispiel unklaren oder unvollständigen Bedarfsmeldungen aufgetreten? Hattest du so etwas schon mal?

0:7:33 -> 0:8:8

B:

Ja durchaus, das sind immer Lernprozesse sowohl beim Kunden als auch bei dem, der die Anforderungen aufnimmt. Hängt immer vom Reifegrad des jeweiligen Kon-

terparts ab. Das heißt auch, dass umso mehr du dich in der Situation meldest und Sachen aufgenommen hast, umso genauer kannst du mal nachfragen und hörst auch die Unsicherheit auf der einen oder anderen Seite heraus.

0:8:8 -> 0:8:17

I:

Gibt es denn irgendwie ein Mechanismus, wie die Qualität von Bedarfs Meldungen bewertet wird, oder gibt es irgendwelche Kriterien oder Standards womit dann beurteilt wird, ob eine Bedarfsmeldung gut oder eher schlecht ist?

0:8:17 -> 0:8:58

B:

Ich bin ein Fan davon auch Sachen mal querlesen zu lassen und dann vielleicht mal die ein oder andere Meinung einzuholen und auch noch mit dem Kunden, der die Bedarfsmeldungen stellt ganz nah dran zu bleiben und dann zu gucken, dass das immer funktioniert. Damit auch zum Schluss das rauskommt, was am Anfang vielleicht schon ne Idee gewesen ist.

0:8:58 -> 0:9:7

I:

Wie würdest du denn die Qualität von Bedarfsmeldungen verbessern?

0:9:7 -> 0:9:41

B:

Auch wirklich mit dem Kunden das Ganze einmal durchexerzieren und festzustellen, ob das Verständnis auf allen Seiten das gleiche ist. Weil nur dann kann es auch gut und produktiv werden. Wenn direkt am Anfang schon irgendwie Unklarheit da ist, dann kannst du auch davon ausgehen, dass da Diskussionsbedarf entsteht.

0:9:41 -> 0:9:49

I:

Welche Auswirkungen haben denn unklare und fehlende Informationen in Bedarfsmeldungen in Bezug auf die Effizienz und den Erfolg des Projektes? Hast du da irgendwie Erfahrungen machen können?

0:9:49 -> 0:10:31

B:

Ja, wenn wir uns unklar sind, wird der Aufwand immer um ein Vielfaches erhöhen, weil das irgendwie im Nachgang immer noch mal gerade gezogen werden muss. Wenn du Pech hast, kannst du alles, was du bisher gemacht hast wegschmeißen und nochmal neu anfangen. Bedeutet natürlich auch immer, dass ein großes Diskussionspotenzial zwischen sowohl den Projektbeteiligten als auch Kunde und Projekt existiert.

0:10:31 -> 0:10:44

I:

Wie könnte man theoretisch sicherstellen, dass die Bedürfnisse und Anforderungen aller Leute, die involviert sind, irgendwie angemessen berücksichtigt werden?

0:10:44 -> 0:14:56

B:

Ich glaube wenn du da mit viel Erfahrung reingehst und dann auch vielleicht genau weißt, wo du drauf zu achten hast. Mir fehlt ein bisschen die Fantasie, aber das ist ja jetzt dann deine Aufgabe da so ein Automatismus zu erkennen. Das wird auf jeden Fall schwierig weil ich auch an der Stelle glaube, dass wenn du so ein System schaffst, die quasi den Match drauf machen können, es trotzdem viel lernen muss. Von daher glaube ich, Erfahrung ist das es ausmacht, um am Ende zu sagen was Sinn macht oder nicht.

A.1.3. Interview 3

0:0:14 -> 0:0:22

I:

Ich habe mir ein paar Infos geholt. Du warst zumindest mal richtig CC-Leiter oder?

0:0:22 -> 0:0:45

B:

Ja ich war mal richtig CC Leiter genau. Ich kann zumindest sagen, dass ich sogar mal 2 Kompetenz Center, 1 in München und 1 in Dortmund geleitet habe und für 48 Menschen zuständig war.

0:0:45 -> 0:0:51

I:

Ok, das bedeutet aber du machst teilweise noch CC Leitung. Oder gar nicht mehr?

0:0:51 -> 0:1:21

B:

Nein, aus in der Tat persönlichen Gründen habe ich vor 2 Jahren die Entscheidung getroffen, dass ich diese Rolle verlassen muss.

0:1:39 -> 0:1:44

I:

Okay, das heißt aber auch du bist aktuell Projektleiter.

0:1:44 -> 0:6:1

B:

Ich bin aktuell von der Laufbahnstufe höher. Programm Manager das ist formal noch eine Führungslaufbahn oder Führungsrolle. Das ist etwas anderes zumindest in der adesso Welt als Projektleitung. Ich kümmere mich in der Tat dann in einer Mischform von Projektleitung und Produktunterschiede um interne Projekte. Ich arbeite mit Studierenden zusammen, ich übernehme Angebotsmanagement für große Angebote. Ich vertrete einzelne Themen, bei dem man jemanden braucht, der entsprechend erfahren ist und dann auf der anderen Seite gewisse intellektuelle Fähigkeiten mit sich bringt. Das sind dann aber eher Sonderthemen. Also ein relativ buntes Sammelsurium. Was ich nicht mehr habe ist Personal Verantwortung. Nicht weil ich nicht mit Menschen umgehen kann, sondern sind beispielsweise die sehr verwalterischen Aspekte nicht so angenehm, die nun mal in dieser Rolle mit drinstecken. Ich übernehme auch noch Delivery Management. Das heißt für große Kunden, gibt es eine spezielle Schnittstellen Rolle, die im Prinzip die adesso Organisation vor dem Kunden kapselt, weil der Kunde gar nicht im Detail wissen soll, wie wir aufgebaut sind. Der Kunde hat genau solche Bedarfsanfragen und sagt, ich brauche einen Senior Java Developer mit diesen Fähigkeiten und ich bin dann als Delivery Manager dafür zuständig.

0:6:1 -> 0:6:12

I:

Dann wäre jetzt meine Frage was die typischen Stakeholder bei einer Erstellung von einer Bedarfsmeldung sind und welche Rolle hast du dabei?

0:6:12 -> 0:10:12

B:

Also Bedarfsmeldungen sind ja ein sehr wichtiges Element in einem der adesso Kerngeschäftsprozesse. Wir sind ein IT-Dienstleister. Das bedeutet, wir entwickeln nicht selber etwas im Sinne von Produkten. Das ist etwas anderes und nennt sich Produktgeschäft. Das heißt also, wenn wir etwas tun, Entwicklungstätigkeiten aufnehmen, beraten, brauchen wir und benötigen wir immer einen Auftrag. Also einen Kunden. Wenn wir Kundenaufträge haben, weil wir ja ein IT Dienstleister sind, heißt das unsere Kerngeschäftsobjekte sind die Mitarbeitenden, die mit unterschiedlichen Qualifizierungen daneben in genau diesen Kundenaufträgen tätig sind. Das können vom Kunden durchgeführte Projekte sein. Ein Kundenauftrag kann aber auch sein, dass adesso ein Projekt für den Kunden durchführt. Wir müssen wie gesagt diese Projektstellen gezielt besetzen. Das heißt, wir haben immer eine Projektorganisation, mal ist es auch eine Programmorganisation, wo diese Stellen mit entsprechenden Rollen oder Stellenanforderungen beschrieben werden. Entweder kommen die vom Kunden direkt, oder wir formulieren die selbst. Diese Beschreibung der Anforderungen, also welche Fähigkeiten und welche Erfahrung und in welchem Umfang Menschen diese mitbringen müssen, damit sie genau diese Projektstelle besetzen können und damit eine ganz bestimmte Rolle in einem Projekt Kontext einnehmen oder in einem Programm Kontext. . . Diese Beschreibung ist es, was wir als Bedarfsmeldung bezeichnen. Die Quellen dafür sind entweder Kundenorganisationen, dass die dann durch große Accounts wie e-on, die Deutsche Bahn, die ihre ganz eigenen Systeme haben und dann in einer, nicht normierten, aber in einer semi strukturierten Form diese Anforderungen dokumentiert werden. Die gehen dann mehr oder weniger 1 zu 1 an uns über und wir müssen damit weiterarbeiten. Bis dahin, dass wir als Delivery Manager, als Projektleiter, als Programm Manager, als Account Manager oder Vertriebler mit Kunden sprechen, in Projekte reinschauen, Projektorganisationen definieren und selbst in diesen Rollen eine Bedarfsmeldung erfassen. Es geht allerdings jedes Mal darum, innerhalb eines Projektes oder Programms eine bestimmte Stelle zu besetzen.

0:10:12 -> 0:10:20

I:

Welche Art von Projekten hat adesso typischerweise? Hast du da vielleicht ein paar Beispiele?

0:10:26 -> 0:18:3

B:

Die erstmal abstrakteste und wichtige oder grundsätzliche Unterscheidung ist: Es

gibt Projekte, die ein Kunde eine Kundenorganisation aufsetzt und durchführt, an denen wir uns dann beteiligen, indem wir beispielsweise in bestimmte Rollen an bestimmte Stellen dort Menschen reinbringen. Das heißt, wir arbeiten dann aber in einem extern definierten und normalerweise auch gesteuerten und kontrollierten Projektkontext. Das andere ist, wenn wir im Kundenauftrag Projekte aufsetzen. Da könnte man dann auch nochmal differenzieren. Einmal Projekte unter unserer Kontrolle im Kundenauftrag und dann gibt es natürlich auch interne Projekte. Da kann man aber relativ schnell sagen, dann ist es halt eine interne Stakeholder Position. Zum Beispiel kann die HR-Abteilung auch sowas beauftragen. Das unterscheidet sich dann nicht wesentlich. Was allerdings da dann besser in unserer Kontrolle liegt, ist in der Regel, wenn wir selbst die Projekte in der Organisation in der Durchführung verantworten. Dann haben wir auch die Ausprägung der Projekt Organisation, das Vorgehensmodell, usw. in der Hand. Da haben wir sehr viel mehr Flexibilität, was eben die Definition von Rollen, Anforderungen usw. angeht. Das ist so der wesentliche Unterschied in Bezug auf externe Projekte und von uns durchgeführte Projekte. Es gibt dann aber nochmal einen wesentlichen Unterschied in Bezug auf die Vergütung. Wir unterscheiden da grundsätzlich zwischen Time und Material Aufträgen oder Beauftragung und Festpreis Beauftragung. Festpreis bedeutet, dass es von unserer Seite eine bestimmte, klar bemessene und abgegrenzte Leistung angeboten wird und die Auftraggeber Seite verhandelt mit uns dafür einen festgesetzten Preis. Damit muss man bei der Durchführung beachten, dass man nicht so lange vor sich hinarbeiten kann, bis dann zum Beispiel die Auftraggeberseite sagt OK, wir sind jetzt zufrieden. Sondern man hat halt ein beschränktes Budget und in der Regel auch beschränkte Zeit und trägt damit auch ein größeres Risiko. Dann gibt es da entsprechend die timen Material oder t und m Kontexte. Da trägt die Auftraggeberseite in der Regel das größere Risiko, weil wir abstrakt bezeichnet, erstmal nur dazu verpflichtet sind, nach durchschnittlicher Qualität und Güte solche einzelnen Leistungen über zum Beispiel Mitarbeitende beizusteuern, zu erbringen. Trotzdem hat auch da die Erfahrung gezeigt: Wir sind immer an unserer Kunden Zufriedenheit oder langfristigen Kundenbindung interessiert, dass es uns auch gar nicht hilft, wenn wir in einem t und m Kontext nur durchschnittlich oder vielleicht auch mal schlechte Arbeit leisten und dann hinterher sagen, das war jetzt aber gar nicht unsere Verantwortung. Das ist euer Risiko gewesen. Damit kommen wir auch nicht durch. Und es ist in den seltensten Fällen auch so, dass ein Kunde unerschöpfliche Geldmittel hat. Selbst wenn dieser Kunde diese unerschöpflichen Geldmittel hätte, sagen sie aus rein wirtschaftlichen Aspekten natürlich auch zu irgendeinem Zeitpunkt das reicht jetzt, wir möchten nicht noch mehr Geld ausgeben. Das ist

doch mal eine grundsätzliche Unterscheidung, was das Bezahlen angeht. Mit hinein spielt auch noch eine Unterscheidung in die Art der Projekte. Das ist nämlich einmal ein Gewerk, wo wir Gewährleistung übernehmen und das auch entsprechend kalkulieren müssen. Ganz häufig werden Gewerke in Kombination mit Festpreisen angeboten und durchgeführt. Interessanterweise müssen sie es aber gar nicht zwingend. Im Umkehrschluss meistens, wenn ich nach t und m arbeite, handelt es sich auch dann von unserer Leistung um Dienstleistung. Das sind aber teilweise, wenn man da wirklich genau draufschaut oder wenn man da juristisch drauf schauen würde feine Unterschiede. Wenn man beispielsweise bei der Beauftragung oder in der Kommunikation ganz bestimmte Begriffe verwendet und es würde irgendwann vor Gericht landen, egal ob man einen Dienstleistungsvertrag abgeschlossen hat und die ganze Zeit meinte auch als Dienstleistungen zu arbeiten, könnte zum Beispiel ein Gericht aufgrund von Formulierungen usw. hinterher feststellen, dass es sich doch um ein Gewerk gehandelt hat. Das sind dann teilweise eher juristische Unterschiede. Am Ende des Tages bedeutet Gewerk natürlich, dass wir an einem Stück Software gearbeitet haben und eben für die konsistente Fehler freie gesamte Funktionalität dann entsprechende Gewährleistung anbieten und übernehmen. Also damit der Kunde für das Geld, dass die Organisation gezahlt hat, eine einsetzbare Lösung bekommt. Wohingegen bei Dienstleistung eine gar nicht funktionale Lösung bei rauskommen muss. Beispiel typischer und klassischer Bereich für Dienstleistungsgeschäft ist ja Consulting. Consulting bedeutet qualifizierte, erfahrene Menschen zu bestimmten Themen kommunizieren, was ausarbeiten, etwas aufschreiben, können etwas spezifizieren können. Aber letztendlich ist da die menschliche Arbeit beziehungsweise der Erkenntnisgewinn im Zentrum und es wird keine Lösung in dem Sinne geschaffen und bereitgestellt. Das sind die für mich zumindest relevanten Unterscheidung.

0:18:3 -> 0:18:16

I:

Wie werden dann Bedarfsmeldungen und die Anforderungen typischerweise bei Adesso kommuniziert und auch dokumentiert. Gibt es einen Ablauf, wie das genau gehandhabt wird?

0:18:16 -> 0:24:4

B:

Für den Staffing-Prozess an sich gibt es einen definierten Ablauf, der aber in Großteilen aus manueller Arbeit und manuellem Arbeitseinsatz besteht. Es gibt keine normierte Form einer Bedarfsmeldung. Es hat sich allerdings zumindest eine grobe

thematische Struktur etabliert. Das bedeutet in der Regel hat man so etwas wie eine Überschrift und Bezeichnung. Dann hat man in der Regel einen Bereich, der den Einsatz Kontext ein wenig allgemeiner beschreibt. Dann hat man einen Bereich, der auf die individuell geforderten Fähigkeiten und Erfahrungen eingeht. Man hat normalerweise eine Gewichtung dieser Skills. Das bedeutet in Bezug auf Expertise, die da erwartet wird oder mal solche Dinge wie Primary in Secondary usw. Unterteilungen. Wir haben in diesen Bedarfsmeldungen dann aber auch wirtschaftlich relevante Informationen damit verknüpft. Das ist typisch für ein Dienstleistungsunternehmen. Das bedeutet uns interessiert dann die vertraglichen Konditionen im Sinne von Tagessatz, was für eine Organisation für adesso, im Sinne von Dienstleistung auch absolut relevant ist. Das sind dann immer diese Parameter. Ab wann ist der Einsatz gewünscht? Und für wie lange? Weil wir immer prüfen müssen, selbst wenn wir beispielsweise sehr gut fachlich passende Mitarbeitende finden, sind sie aber eventuell schon in anderen Projekten eingesetzt und können dementsprechend gar nicht dort leisten. Es mag dann je nach Kundenkontext noch weitere Informationsblöcke geben. Wir arbeiten mittlerweile auch mit einem Leveling-System, weil man sich als Organisation oder eben als Mensch innerhalb einer Organisation ein bisschen besser in Abfragen organisieren kann, wenn an solchen detaillierteren Bedarfsmeldungen gewisse Labels dran stehen wie beispielsweise Azure als Technologie oder oder Java. Dann kann ich eine potenziell größere Menge von Bedarfsmeldungen, die ich manuell durchsuche besser filtern und einschränken. Was in Bezug auf unserem Staffing-Prozess sehr relevant ist, ist beispielsweise dann der Status einer solchen Bedarfsmeldung, weil die Organisationen nicht alle Bedarfsmeldungen interessiert. Manche Bedarfsmeldungen sind schon verarbeitet und erfüllt. Manche sind nur dokumentiert und sollen aber gar nicht weiter beachtet werden und erst wenn sie beispielsweise bei uns eskaliert werden, sollte eigentlich die relevante Organisation darauf schauen. Wir haben eine vertikale Einteilung des Unternehmens in Branchen, bzw. technologisch getriebenen Organisationseinheiten. Das sind unsere sogenannten Line of Business. Mittlerweile sind das sogar Business Areas und es ist an der Stelle durchaus relevant, welche adesso Organisationseinheit für so eine Business Area oder Line of Business zuständig ist. Wenn wir an der Stelle mit einem gewissen Vorkaufsrecht feststellen, dass wir es nicht bedienen können, werden natürlich alle gefragt und es dürfen beispielsweise aus der Line of Business Motiv selbstverständlich auch mitarbeiten in Projekten der Line Cross Industries und umgekehrt getätigt werden. Aber umso mehr ist es wichtig, dass dann die jeweils geforderten Skills und Erfahrungen und wirtschaftlichen Konditionen möglichst präzise beschrieben werden, damit diese Fragen die aufkommen nicht immer wieder gestellt werden

und immer wieder beantwortet werden müssen. Das ist aber, wie gesagt schon Kern Staffing-Prozess.

0:24:4 -> 0:24:22

I:

Du hast auch schon einige Punkte in Richtung Verfügbarkeit genannt, dass das sehr wichtige Aspekte sind. Gibt es besonders wichtiger oder auch unverzichtbare Punkte in einer Bedarfsmeldung, die in jeder drin sein sollte?

0:24:22 -> 0:28:46

B:

Das kommt auf die Perspektive an. Wenn ich jetzt eine rein fachliche Perspektive einnehme, ist es da unverzichtbar, dass aufgelistet oder benannt wird, welche Fähigkeiten oder Skills konkret mitgebracht werden müssen und idealerweise auch in welcher Erfahrung und Güte das ist. Aus der Perspektive zum Beispiel des Dienstleistungsunternehmens adesso ist relevant, dass ein Beginn des Einsatzes, ein voraussichtlicher Einsatzzeitraum, ein Tagessatz dran steht. Dass da dran steht, ob Freelancer, ob Smartphone, smartshore Fähigkeit gegeben ist oder near Shore, oder wie die sprachliche Ausrichtung ist. Also ob das zum Beispiel deutschsprachig ist, ob Englisch. Entsprechend zulässige Kommunikationsmittel sind wichtige Zusatzinformationen, die genau darauf abzielen, dass alle Menschen, die potenzielle Kandidaten/Kandidatinnen auf so eine Bedarfsmeldung gefiltert werden. Wenn es nämlich auf der Ebene, dass die Verfügbarkeit aber auch darüberhinausgehend nicht passt oder wenn es zum Beispiel von einem Tagessatz, aus welchen Gründen auch immer sehr unattraktiv ist, es dazu führen kann, dass dann die Organisation bestimmte Personen nicht anbietet. Das ist aber wie gesagt eine Perspektive, die durch die Natur von adesso als Dienstleister, und wir sind ein auslastungsgetriebenes unternehmen, was halt auch versucht wirtschaftlich zu arbeiten, darüber reinkommen. Idealerweise möchte zum gewünschten Einsatzbeginn auch dann wirklich eine Person nicht nur theoretisch benannt, sondern idealerweise interviewt, geprüft, eingeführt und wie auch immer wurde und dann wirklich los laufen kann.

0:28:46 -> 0:28:51

I:

Gibt es Herausforderungen und Schwierigkeiten bei unklaren oder unvollständigen Bedarfsmeldungen? Hast du da Erfahrungen machen können?

0:28:51 -> 0:39:18

B:

Ja. Bei einem Auftraggeber handelt es sich in der Regel um Branchen, die eben nicht als Kerngeschäft IT Software Entwicklung betreiben. Dementsprechend fällt es ihnen teilweise schwer, präzise Bedarfsmeldungen zu formulieren, die inhaltlich genau das transportieren oder beinhalten oder umfassen, was sie eigentlich an Fähigkeiten und Erfahrung benötigen. Das liegt einfach daran, dass sie in Teilen nicht für jeden Kunden gilt, aber in Teilen den Prinzip Menschen etwas beschreiben lassen, die davon nicht wirklich Ahnung haben. Das wiederum führt dazu oder kann dazu führen, dass der Kunde etwas anfordert, was der Kundenorganisation, dem Projekt oder wie auch immer im schlimmsten Falle sogar gar nicht hilft. Oder aber das ist eine andere Ausprägung, dass dann Kombinationen von Erfahrung und Fähigkeiten gesucht werden, die es in der realen Welt einfach so nicht gibt. Das ist die berühmte eierlegende Wollmilchsau, wo man drauf schaut und sagt, diese Menschen hätten wir auch gerne als Mitarbeitende. Vielleicht gibt es auf diesem Planeten auch eine Handvoll davon, aber das ist unrealistisch. Und das kann wie gesagt darin begründet sein, dass auf der auftraggebenden Seite jetzt Menschen damit beauftragt werden, solche Bedarfsmeldungen zu formulieren, die das nicht wirklich können. Eigentlich haben wir an der Stelle schon immer einen idealerweise von uns moderierten Prozess. Deswegen macht zum Beispiel auch die Rolle Delivery Management Sinn. Das hängt wie gesagt auch sehr mit der insgesamten Qualifikation oder Qualität von der Auftraggeberseite in diesem Kontext zusammen. Bei manchen Kundenorganisationen ist das trotzdem sehr gut eingespielt und etabliert und funktioniert auch so. Bei manchen muss man da früh ansetzen und sagen wir sprechen miteinander, und ich arbeite dann beispielsweise in diesem Gespräch heraus, was der Kunde wirklich benötigt, was abgrenzbar eine sinnvolle Bedarfsmeldung ist, was es für eine Stelle beinhaltet und womit wir dann weiterarbeiten können. Das ist oder kann ein Problem sein, wenn wir nicht entsprechend damit umgehen. Nicht nur die fachliche Qualität oder auch die die Passgenauigkeit, die wir dort anbieten können ist da wichtig, sondern auch vor allem die Schnelligkeit vom Staffing-Prozess. Bedeutet, wenn ein Unternehmen sich so aufgestellt hat, dass eine Bedarfsmeldung, die reinkommt innerhalb von sehr kurzer Zeit bearbeitet wird, sagen wir mal 2 Stunden, dann habe ich einen klaren Vorteil gegenüber zum Beispiel einem Anbieter, der dafür 2 Tage braucht oder eine Woche oder 2 Wochen. Denn für die Auftraggebende Seite ist klar, wenn ich dann Rückmeldungen bekomme und ich schaue da rein und diese Rückmeldungen sind plausibel... Was hält mich davon ab, dann zu sagen ich beauftrage ihn jetzt. Es muss nicht perfekt sein, aber wenn es plausibel ist und die Konditionen sind gut,

dann ist der Auftrag ausgesprochen und die anderen gehen logischerweise leer aus. Wenn wir eben entsprechend in unserem Matching entweder nicht gut arbeiten oder überfordert sind und beispielsweise einen Software Architekt für Java mit bestimmten weiteren Anforderungen gefordert ist und wir bieten da ein Profil drauf an, wo man darauf schaut und feststellt, dass es ein .Net Software Developer ist, ist es bei diesem extremen Beispiel so, dass es glücklicherweise auch sofort auffällt. Aber das würde natürlich dann zu Irritationen auf Kundenseite führen. Bedeutet: Wir sollten nicht nur dafür sorgen, dass die Erwartungen oder Anforderungen möglichst gut an der Realität sind, sondern wenn wir dann wiederum auch da etwas matchen und einreichen, dass das dann auch diesen Anforderungen nach Möglichkeit entspricht und, dass innerhalb von möglichst kurzer Zeit so geschieht. Und die unglücklichste Variante ist, wenn man ein Prozess durchläuft und man bietet da jemanden an und der wird sogar genommen und wird eingearbeitet und dann stellt man irgendwie fest der macht irgendwie Unsinn. Man hat nie eine Garantie. Letztendlich sind es Menschen die dort arbeiten.

0:39:18 -> 0:39:22

I:

Welche Auswirkungen haben unklare oder auch fehlende Informationen in Bedarfsmeldungen jetzt aber konkret in Bezug auf die Effizienz und den Erfolg von Projekten.

0:39:27 -> 0:40:49

B:

Im Idealfall wird, möglichst früh erkannt, dass eine Bedarfsmeldung lückenhaft, unpräzise wie auch immer formuliert ist. Dann muss nachgefragt werden. Ich schaue mir etwas an, versuche zu verstehen, was die andere Seite sucht und wenn das für mich dann nicht konsistent auf die zumindest für mich bekannten Rollenlösung ist, dann muss ich nachfragen. Alles andere ist eine Interpretation. Dann läuft man mit sehr großer Wahrscheinlichkeit in die von mir gerade beschriebenen Probleme rein. Die schlechteste Variante ist, dass ich sage ich nehme die Informationen, die ich jetzt da vorgelegt bekommen habe, interpretieren sie nach besten Wissen und Gewissen und dann ist es aber mehr oder weniger ein Glücksspiel. Das heißt, wenn ich dann Profile finde, könnten sie immer noch zufällig das sein, was der Kunde eigentlich wollte und gesucht hat.

0:40:49 -> 0:41:1

I:

Die letzte Frage hast du im Grunde auch schon mit beantwortet. Wie könnte man sicherstellen, dass Bedürfnisse und Anforderungen aller relevanten Stakeholder in einer Bedarfsmeldung Berücksichtigt werden?

0:41:4 → 0:42:26

B:

Zwei Möglichkeiten. Man könnte, da glaube ich aber nicht dran, natürlich den Prozess standardisieren und stark formalisieren. Also das im Prinzip von einer öffentlichen Stelle aus gesagt wird: Alle Dienstleister und Auftraggeber dieser Welt wenn ihr in dieser Art Geschäft betreiben wollt, müsst ihr so ein Format einreichen. Also Bürokratie pur. Das würde uns nichts verbessern, aber das ist eine Möglichkeit. Die andere Möglichkeit ist meines Erachtens, dass dann für genau solche Prozesse entsprechend versierte Menschen diesen Prozess, das heißt die Anforderungserhebung, die Dokumentation, das erfüllen diese Anforderungen komplett begleiten und moderieren. Das ist Delivery Management. Das ist Business Development. Manchmal auch Account Management.

0:42:26 → 0:42:30

I:

Dann sind wir eigentlich schon durch mit den Fragen. Hast du noch zu irgendeinem Punkt irgendwelche Fragen oder irgendwas, was vielleicht noch für mich in dem Themenbereich interessant sein könnte?

0:42:30 → 0:45:6

B:

Die größte Schwierigkeit liegt darin, dass wir Informationen über eine natürliche Sprache transportieren. Das ist zwar flexibel, weil die Sprache an der Stelle ja eben nicht formalisiert ist. Was aber immer das Risiko mit sich bringt, dass etwas nicht präzise beschrieben, abgegrenzt oder interpretierbar wird und genau bei solchen Prozessen, wo wir eigentlich präzise arbeiten wollen, haben wir genau diese große Herausforderung, dass die bisher benutzte Art, um diese Informationen zu ermitteln und die gerade wieder zu lesen zu interpretieren eben ein Stück weit ungenügende Mittel, nämlich den natürlichen sprachigen Raum verwendet. Da ist zwar mit gesundem Menschenverstand gearbeitet worden. Das heißt, es haben sich Semistrukturen gebildet. Aber es ist wirklich sehr individuell unterschiedlich in was für einer Qualität oder was für einer Realitätsnähe solche Bedarfsmeldungen formuliert werden.

Wenn wir auf unserer Seite jemanden sitzen haben, der oder die eben auch in diesem Umfeld relativ wenig Ahnung und Erfahrung hat, dann haben wir auch nochmal ein Risiko, dass selbst wenn die Bedarfsmeldungen präzise und realitätsnah formuliert ist, bei uns in der Interpretation etwas schiefgeht. Das bedeutet, dass halt die Menschen, die auf unserer Seite Bedarfsmeldungen lesen und versuchen zu bedienen, leider auch hinreichend viel Erfahrung in der Projekt IT haben müssen.

A.1.4. Interview 4

0:0:49 -> 0:1:5

I:

Erstmal vorweg zu deiner Person. Ich weiß, dass du CC Leiter bist und auch Softwarearchitekt bist. Kannst du vielleicht kurz erzählen, wie deinen Werdegang. Was du studiert hast und wie du dann bei adesso gelandet bist?

0:1:5 -> 0:1:38

B:

Studiert habe ich Kerninformatik an der TU Dortmund mit einem Diplomabschluss. Danach bin ich 2003 im Bereich der Software Entwicklung gestartet. Und habe dann seitdem Software Architekten und Führungskraft gemacht.

0:1:38 -> 0:1:39

I:

Was sind denn dann so die typischen Stakeholder bei der Erstellung von Bedarfsmeldungen und was für eine Rolle hast du dabei?

0:1:57 -> 0:2:0

B:

Stakeholder bei der Erstellung von Bedarfsmeldungen?

0:2:0 -> 0:2:4

I:

Wer sind so Person, die da dran grundsätzlich beteiligt sind.

0:2:11 -> 0:3:4

B:

Wenn wir Bedarfsmeldungen erstellen, sind wir über den Akquise Prozess schon

hinaus. Das heißt, der Vertrieb ist raus. Damit ist für die Erstellung der Bedarfsmeldungen der Projektleiter und Maitre eigentlich relevant. Also hängt immer davon ab. Der Projektleiter hat ein Bedarf und stimmt den mit dem internen Maitre ab, wer dann für das Staffing verantwortlich ist. Das heißt, sie sind auch für die Bedarfserstellung verantwortlich. Daneben hast du dann noch Leute, die Input geben für die Bedarfsmeldung, also Input, der jetzt nicht zum Beispiel durch den Vertrag geregelt ist, wäre so etwas wie das Geld Profil, was dann vom Architekten zum Beispiel geliefert wird.

0:3:4 -> 0:3:14

I:

Was sind denn so typische Projekte, die bei Adesso angenommen und bearbeitet werden? Hast du ein paar Beispiele für Projekte, die so durchgeführt werden?

0:3:20 -> 0:3:27

B:

Typische Projekte alles rund um die Software Entwicklung, also sowohl in den Development als auch Consulting rollen.

0:3:34 -> 0:3:47

I:

Und wie werden Projektbedarfe und Anforderungen innerhalb von adesso kommuniziert und dokumentiert? Also gibt es da irgendwie einen Ablauf?

0:3:47 -> 0:3:55

B:

Ja, und die Bedarfsmeldungen werden über das Jira erfasst und entsprechend auch mit allen beteiligten Parteien geteilt.

0:3:55 -> 0:4:19

I:

Was sind Aspekte, die in einer Bedarfsmeldung besonders wichtig sind? Gibt es unverzichtbare Punkte, die immer in einer Bedarfsmeldung drin sein sollten.

0:4:19 -> 0:5:47

B:

Rahmenbedingungen wie Kunde, Einsatzbeginn, Laufzeit oder Ort und Tagessatz.

Und die notwendigen Skills sowie eine Aufgabenbeschreibung damit klar ist, was tatsächlich inhaltlich zu tun ist und was vom Kunden oder in dem Fall für ein Profil gefordert wird. Indem Moment von die Aufgabenbeschreibung oder die geforderten Skills fehlen verstehe ich gar nicht, was gesucht wird und kann dementsprechend auch keine passenden Profile anbieten. Die anderen Rahmenbedingungen brauche ich natürlich, um auch zu prüfen, ob das Personal entsprechend überhaupt zum Beispiel verfügbar ist.

0:5:47 -> 0:5:53

I:

Welche Herausforderungen hat man denn bei unklaren oder unvollständigen Bedarfsmeldungen?

0:5:53 -> 0:6:2

B:

Eine Herausforderungen ist, dass das angebotene Personal gegebenenfalls nicht zu dem Einsatz passt oder bei dem Einsatz nicht verfügbar ist. Das heißt am Ende, dass die Rückmeldungen auf die Bedarfsmeldungen gar nicht zielführend ist.

0:6:17 -> 0:6:26

I:

Gibt es Kriterien oder Standards, um die Qualität der Bedarfsmeldungen sicher zu stellen?

0:6:26 -> 0:7:4

B:

Es gibt einerseits die technische Validierung über die Daten im Jira, die man eingibt. Weitergehende Validierung erfolgt dann, nicht immer aber meistens nochmal im 4 Augen Prinzip. Die Projektleitung meldet sich da zur Abstimmung nochmal.

0:7:4 -> 0:7:36

I:

Wie würdest du denn die Qualität und Klarheit von Bedarfsmeldungen verbessern? Was für Mechanismen würdest du denn anwenden, die eventuell auch noch gar nicht von jedem benutzt werden?

0:7:36 -> 0:8:41

B:

Ich glaube, man könnte einige Daten strukturierter erfassen. Bei der inhaltlichen Beschreibung ist es schwierig, das tatsächlich noch automatisiert weiter zu verbessern. Die Herausforderung ist jetzt nicht 15 Java Entwickler Profil zu finden, sondern die besonderen Profile, die wir selten haben, die dann aus der Struktur raus sein, wo du halt zum Beispiel nicht sagen kannst, du hinterlegst die Skills in der Liste und sagt, ich brauche irgendwie Java. Die Herausforderungen ist natürlich alles, was aus dem Raster herausfällt.

0:8:41 -> 0:8:48

I:

Im Bezug auf den Projekterfolg hast du da schon Erfahrungen gemacht, welche Auswirkungen unklare und fehlende Informationen in Bedarfsmeldungen dann wirklich am Ende im Projekt haben?

0:8:48 -> 0:9:32

B:

Die Herausforderung ist, dass die Projektmitarbeitenden dann den Aufgaben im Zweifelsfall nicht gewachsen sind, weil notwendiges Know How in den Technologien fehlt. Weil andere Skills auch unternommen und auch Softskills eventuell fehlen. Und somit das Projekt gar nicht sicher, oder der Projekterfolg gar nicht sichergestellt werden kann.

0:9:32 -> 0:9:45

I:

Wie kannst du sicherstellen, dass die Bedürfnisse und Anforderungen aller involvierten Personen angemessen berücksichtigt werden?

0:9:56 -> 0:10:34

B:

Je nachdem in welche Rolle und in welchen Schritten im ganzen Prozess durch intensives Lesen und drauf schauen. Oder Rückfragen stellen. Rückfrage heißt immer das eine Informationen gefehlt hat.

0:10:34 -> 0:11:26

I:

Ich hätte noch eine spezielle Frage an dich, weil du auch Software Architekt bist und

mit vielen Technologien zu tun hast. Hättest du eine Idee, wie du grundsätzlich das Problem der Informationsgewinnung von unstrukturierten Bedarfsmeldungen angehen würdest. jetzt auch vielleicht in Bezug auf Technologien oder Ansätzen. Hast du da irgendwelche Erfahrungen irgendwann mal machen können?

0:11:26 -> 0:14:27

B:

Man könnte natürlich gerade in der schwierigen Erfassung von den Freitextinformationen, die nicht stark strukturiert sind im Zweifel weitere Validierungen bauen, die auch dann vielleicht über das hinausgehen, was uns Jira an Validierung anbietet. In den Freitextinformationen könnte ich mir vorstellen, dass man da KI gestützt Prüfungen macht, um eine Vollständigkeit sicherzustellen.

A.1.5. Interview 5

0:4:52 -> 0:4:58

I:

Wer sind denn die typischen Stakeholder bei der Erstellung von Bedarfsmeldungen und was für eine Rolle hast du dabei?

0:5:14 -> 0:5:19

B:

Bei adesso ist es im Normalfall so, dass Sales oder der interne Maitre dafür zuständig ist. Er stellt quasi eine Bedarfsmeldung im Normalfall auf Basis von Anforderungen direkt vom Kunden. Also entweder die kommen aus etwas, was wir angeboten haben, oder bei einer Rahmenvereinbarung kommen sie vielleicht aus einem Wettbewerb. Oder es kommt konkret im Projekt aus mündlich genannten Bedarfsmeldungen. Das kann auch durch ein Scrum Master oder Product Owner entsprechend entstehen. Da ist dann derjenige, der die Bedarfsmeldung erstellt auch zuständig, die Kriterien festzulegen. Was für ein Mitarbeiter oder Mitarbeiterin, mit welchen Fähigkeiten wir brauchen und stellt die dann ein. Ich bin Bereichsleitungsebene. Das heißt Ich bin einerseits in den Call, wo sich alle Bereichsleiter des gesamten Unternehmens treffen, um einmal zu schauen, welche Bedarfe eigentlich gerade eskaliert sind. Das heißt, welche sind gerade, oder müssen in naher Zukunft aufgelöst werden? Vieles wird einfach im direkten Kontakt behandelt. Jemand sieht das Ticket, bietet drauf und alles ist gut. Es gibt manche Bedarfsmeldungen, die schwerer zu bedienen sind. In solchen Fällen bin ich involviert. Und ansonsten bin ich eher in einer Support-Rolle in dem

Prozess, weil ich entweder jemanden die Vertretung übernehme, weil ich im Normalfall nicht interner, sondern externer Maitre bin. Das heißt, ich bin eher für das Kunden Management und für die Gemeinsame Gestaltung, wie man zukünftig die Zusammenarbeit gestaltet zuständig, aber nicht so sehr jetzt für einzelne Bedarfsmeldungen. Ansonsten direkte Berührungspunkte mit konkreten Bedarfsmeldungen hab ich nicht.

0:10:59 -> 0:11:8

I:

Wie werden Bedarfsmeldungen und Anforderungen grundsätzlich kommuniziert und dokumentiert?

0:11:8 -> 0:14:11

B:

Von demjenigen der es einstellt, werden die Kriterien definiert und in einem JIRA-Ticket überführt. Das kann sehr unterschiedliche Formen annehmen. Es gibt ein paar Felder, die strukturiert sind. Z.B zu welchem Tagessatz das ganze angeboten wird, wann das Ganze startet, wie hoch das Volumen also an Tagen ist. Es gibt einige Freitextfelder, bei dem drinsteht, was die Aufgaben usw. sind. Zum Beispiel bei einer Rahmenvereinbarung, die wir gerade machen gibt es dann ein Excel, was ausgefüllt werden muss mit seiner Selbstbeurteilung. Wie gut der Kandidat, der jetzt gefunden wird oder gefunden werden soll, darauf passt. So ein Bewertungsschema wird dann mit beigefügt, das die Führungskraft ausfüllen kann. In einem anderen Fall kann das aber ganz anders aussehen, da wird dann einfach der Export aus unserem Profiler, wo also die Profilbeschreibung drinsteht, einfach als Kommentar in das JIRA hinzugefügt und gesagt: Den könnte man jetzt anbieten und das ist eher ein manueller Prozess. Das ist sehr verschieden. Deswegen nutzen wir doch JIRA, weil es im Normalfall nicht genau den einen Case gibt, wie wir Bedarfsmeldung reinkriegen und wie wir diese auch beantworten müssen. Im Normalfall ist es eben so, dass die Bedarfsanfragen in die Welt rausgeschickt werden. Also erstmal in die eigene Organisationseinheit und wenn man dort niemanden hat, der sehr gut passt, dann eben in die gesamte adesso und die Führungskräfte gucken von der anderen Seite da drauf und schauen, ob der eigene Mitarbeiter darauf passt. Im Normalfall ist es nicht so, dass diese Bedarfsmeldungen eine konkrete Suchanfrage sind, sondern die sind eher eine Bedarfsmeldung, wo drauf sich dann jemand melden kann.

0:14:11 -> 0:14:21

I:

Was sind denn besonders wichtige oder auch unverzichtbare Informationen innerhalb einer Bedarfsmeldung?

0:14:21 -> 0:16:50

B:

Wichtig ist für die Führungskraft, die einen Mitarbeiter anbietet, ab wann die dann frei sind. Bis wann, wie lange wieviel Prozent dieser Mitarbeiter oder diese Mitarbeiterin eingesetzt wird. Für denjenigen, der den Bedarf meldet. Ist es am Interessantes oder am wichtigsten was die muss und soll Kriterien sind? Damit der Mitarbeiter möglichst gut Match. Man kann sagen, dass es für beide Seiten wichtig ist. Für die Führungskraft ist es erstmal interessant ob und für wie lange er Mitarbeiter in einem Projekt kriegt. Bekomme ich die passende Ressource auf meine Anfrage? Nachrangiger ist der Tagessatz. Das ist gesamtwirtschaftlich fürs Unternehmen sehr wichtig. Aber dadurch, dass wir eine sehr offene und transparente bereichsübergreifende Zusammenarbeit pflegen, ist so, dass wir, wenn einmal einen Preis, wie z.B. den Tagessatz, von einer Einheit verhandelt wurde, dass das dann von einer anderen Einheit nicht in Frage gestellt wird oder gesagt wird jetzt können wir hier den Mitarbeiter nicht anbieten, weil der Tagessatz zu niedrig ist. Das ist dann so verhandelt und wir sind dann ein Commitment als Firma eingegangen. Wir haben uns selber als Ziel gesetzt, eben hohe Qualität abzuliefern und Kunden zu Partnern zu machen und deswegen bedienen wir dann diese Anfragen.

0:16:50 -> 0:17:2

I:

Welche Herausforderungen oder Schwierigkeiten sind bei unklaren oder auch unvollständigen Bedarfsmeldungen aufgetreten? Hast du da irgendwie Erfahrungen machen können?

0:17:2 -> 0:19:17

B:

Wenn die Bedarfsmeldungen zu wenig Informationen darüber enthalten, was genau die Mitarbeiter, die da drauf angeboten werden sollen, können sollen, wo die vielleicht auch sitzen sollen, idealerweise, ob das ein Angebot einer einzelnen Arbeitskraft ist oder ob das im Rahmen eines Teams ist. Zu schauen ob der Mitarbeiter oder die Mitarbeiterin auf diese Anfrage passt. Wenn ich jetzt einen Junior Mitarbeiter habe, der ganz frisch bei Adesso angefangen hat, dann sollte ich ihn vielleicht nicht auf ein

Ticket anbieten, wo diese Person dann ganz alleine zum Kunden geschickt wird. Das wäre unfair der Person gegenüber und wahrscheinlich auch nicht von Erfolg gekrönt. Wenn auf irgendeiner Ebene was fehlt, führt es dazu, dass entweder falsche Zuordnung gemacht werden, oder dass es einen Kommunikationsoverhead gibt. Oder, dass Versprechungen gemacht werden, die dann hinterher nochmal korrigiert werden müssen, weil es dann doch nicht gut passt. Das heißt es ist schon essentiell, dass die Bedarfsmeldungen gut sind, weil nur dann auch die erste Wahl auch eine gute Wahl ist. Wir haben und deswegen in beide Richtungen Schwierigkeiten. wir haben aber teilweise, dass Mitarbeiter angeboten werden, die nicht so gut passen, wo dann die Informationslage auf der anderen Seite schwierig ist. Das ist dann unter Umständen ein großer Overhead und weil man dann auf der Seite sehr viel Aufwand und unter Umständen durch Interviews oder ähnliches betreiben muss. Was dann auch sehr zeitaufwendig sein kann.

0:19:17 -> 0:19:21

I:

Wie wird denn die Qualität von Bedarfsmeldungen bewertet? Gibt es irgendwelche Kriterien oder Standards die ihr einhaltet?

0:19:21 -> 0:24:2

B:

Es gibt, wie gesagt ein paar feste Felder, die auch in dem JIRA eben festgelegt sind, auch ein paar feste Zustände und Übergänge, die einen groben Rahmen geben. Dann gibt es bei größeren Rahmenvereinbarungen feste Rahmenbedingungen, in denen gesagt wird, so übersetzen wir die Anforderungen des Kunden in eine Bedarfsmeldung. Aber es ist nicht so, dass es eine übergreifende Qualitätssicherung über die Tickets gibt, sondern das ist eher ein Lernprozess. Leute die Tickets erzeugen machen, das entweder schon eine Weile oder es wird dort irgendjemanden zur Seite gestellt, der das schon länger macht. Die muss Kriterien was sind die? Soll Kriterien, dass man dabei schreibt. Nicht nur, dass jemand ein Skill hat, sondern auch wie viel Erfahrung er oder sie in dem Skill. Und noch ein paar weitere Punkte, die die dann ja dazu führen, dass die Ticket Qualität hoffentlich gut ist. Aber sie ist auch nicht durchweg gut, das muss man auch dazu sagen. Es gibt sehr gute Tickets und es gibt welche, die nicht ganz so gut sind. Wobei man bei manchen auch fairerweise sagen muss, dass die unter Umständen dann auch aus Rahmenvereinbarungen kommen, wo wir eigentlich nur Anfragen durch Reichen und auch nur teilweise ein zwei Tage Zeit ist, um überhaupt eine Anfrage zu machen und uns nicht sehr viel mehr übrigbleibt

als nahezu alles was wir von außen reinbekommen dann 1 zu 1 zu übernehmen. Weil dort nicht genug Raum ist, um einen richtigen Angebotsprozess zu machen. Bei großen Rahmenvereinbarungen gibt es dann auch nochmal auf JIRA Tools, die dabei unterstützen diese Bedarfsmeldungen zu erzeugen, zu tracken bis hin zu einem Angebot zu bringen.

0:24:2 -> 0:24:19

I:

Wie kannst du denn sicherstellen, dass die Bedürfnisse und Anforderungen aller involvierten Leute in Bedarfs Bildung berücksichtigt werden?

0:24:19 -> 0:25:43

B:

Ich glaub, es gibt nicht die eine Person, die das sicherstellen kann. Das ist eine gemeinsame Verantwortung. Damit dann am Ende sichergestellt ist, dass das gut funktioniert, gibt es sicherlich ein paar Punkte, wo man positiv darauf einwirken kann. Wir machen regelmäßig Bereichsmeetings, wo dann auch ein Agenda Punkt im Normalfall ist wie das Staffing läuft. Wir haben auch wöchentlich Jour Fixe, wo wir dann über das Tagesgeschäft reden und was die aktuell laufende Staffing-Prozesse sind? Wie sieht die Auslastung aus usw.? Dort kann man schauen wo denn Probleme sind und dann entsprechend kommunizieren. Viele von den Bedarfen werden gar nicht im JIRA eingetragen, sondern es werden direkt entsprechende Mitarbeiter angesprochen.

A.2. Paraphrasierung, Generalisierung und Reduktion

Fall	Nr.	Paraphrase	Generalisierung	Reduktion
1	1	Bedarfsmeldungen beschreiben Software-Entwicklungsprojekte, angefangen von Projekten in dem ein adessi in einem Kundenprojekt arbeitet über gemischte Teams bis hin zur kompletten Lieferung.	Bedarfsmeldungen Beschreiben Software-Entwicklungsprojekte wie Kundenprojekte, bei dem in gemischten Teams bis zur Auslieferung gearbeitet wird	K1 Arten von Projekten -Kundenprojekte - Softwareentwicklungsprojekte -Time Material -Festpreis K2 Stakeholder
1	2	Kommunikationswege sind Initial über den Maitre, der das Staffing übernimmt, teilweise auch über das eigene Netzwerk zwischen Führungskräften. Am Ende über das Staffing Jira.	Kommunikationswege über Maitre, Führungskräftenetzwerk, am ende Staffing Jira	-Maitre -Führungskräfte-netzwerk -Sales -Projektleiter -Fachverantwortliche -Entscheider -Geschäftsführer -Delivery Manager -Account Manager
1	3	Wichtige Bedarfsmeldungsinformationen sind Senioritätslevel, Tagessatz, Remote/on Site Einsatz, Dauer, Technischer Stack (Muss- und Kann Kriterien), Einarbeitungszeiträume, Lieferverpflichtung.	Wichtige Bedarfsmeldungsinformationen sind Senioritätslevel, Tagessatz, Remote/on Site Einsatz, Dauer, Technischer Stack (Muss- und Kann Kriterien), Einarbeitungszeiträume, Lieferverpflichtung	

Fall	Nr.	Paraphrase	Generalisierung	Reduktion
1	4	Bedarfmeldungen sollten aussagefähig sein, welche technischen Kompetenzen wichtig sind, welche Tagessätze, ob Remote möglich ist und die Dauer.	Aussagefähigkeit von Bedarfmeldungen durch technischen Kompetenzen, Tagessätze, Remote Möglichkeit und Dauer	K3 Wichtige Informationen -Tagessatz -Einsatz -Dauer -Tech Stack -Muss/Kann Kriterien
1	5	Der Anforderer muss gegebenenfalls Fragen mehrfach beantworten. Dadurch verliert man teils den Überblick.	Überblick verlieren durch Mehrfachbeantwortung von Fragen	-Einarbeitungszeiträume -Lieferverpflichtung K4 Bedarfsmeldung -Überschrift
1	6	Typische Stakeholder sind Sales/PL: Anforderer mit den technischen Informationen, Maitre: kümmert sich um das Staffing.	Stakeholder sind Sales/PL: Anforderer mit technischen Informationen, Maitre	-Beschreibung -Einsatzkontext -Datum -Volumen -in Jira gespeichert -gewichtete
1	7	Bewertung der Qualität erfolgt gar nicht.	keine Qualitätsbewertung	Fähigkeiten -keine feste Struktur
1	8	Qualitätsverbesserung durch klarere Vorgaben und weniger Freitext, aktuell: durch Nachfragen und Bitten um nachträgliche Pflege, gegebenenfalls durch Reviewprozesse.	Qualitätsverbesserung durch klare Vorgaben, weniger Freitext, Nachfragen und nachträgliche Pflege (Reviewprozess)	

Fall	Nr.	Paraphrase	Generalisierung	Reduktion
1	9	Auswirkungen fehlender Informationen sind, dass das Staffing länger dauert und die Stellen potenziell durch andere Dienstleister besetzt werden.	Fehlinformationsauswirkungen sind längerer Staffingprozess und Umbesetzung durch anderen Dienstleister	K5 Qualitätsbewertung -keine vorhanden -Erfahrung -über Projektleitung -intensives Lesen -Rückfragen stellen -grober Rahmen
1	10	Gute Abstimmungen bevor die Bedarfsmeldung erstellt wird, gegebenenfalls durch ein Quality-Gate (Review).	Abstimmung vor Bedarfsmeldungserstellung durch Quality Gate (Review)	durch Jira -regelmäßige Meetings K6 Qualitätsverbesserung -klare Vorgaben
2	11	In Bedarfsmeldungen sind die typischen Stakeholder Fachverantwortliche, Entscheider und höhere Management-Ebenen wie CEO oder Geschäftsführer.	Stakeholder sind Fachverantwortliche, Entscheider, CEO oder Geschäftsführer	-weniger Freitext -Reviewprozess -Verständnis- übereinstimmung -Beseitigung Missverständnisse -strukturierte Datenerfassung
2	12	Projekte lassen sich in Time Material Projekte und Festpreis Projekte unterteilen, wobei die Art der Projekte die Ressourcenplanung beeinflusst.	Projektunterscheidung in Time Material und Festpreis Projekte	-KI-gestützte Prüfungen

Fall	Nr.	Paraphrase	Generalisierung	Reduktion
2	13	Die Dokumentation der Bedarfsmeldungen variiert je nach Kundenanforderungen, von einfachen Word-Dokumenten bis zu detaillierten Präsentationen.	Dokumentation von Bedarfsmeldungen variieren (von Word-Dokument bis Präsentation)	K7 Auswirkungen -Überblick verlieren -längerer Staffing-Prozess -Umbesetzung -erhöhter Aufwand - Mehrfachbeantwortung
2	14	Wichtig in Bedarfsmeldungen sind das Tech Stack, der Aufwand und die zeitlichen Erwartungen.	Wichtig sind Tech Stack, Aufwand und zeitliche Erwartungen	-Missverständnis -unpassendes Personal
2	15	Senioritätslevel ist zweitrangig und abhängig von der Projektsituation.	Senioritätslevel zweitrangig	
2	16	Unklare oder unvollständige Bedarfsmeldungen führen zu erhöhtem Aufwand und möglichen Verzögerungen, da Missverständnisse geklärt werden müssen.	Erhöhter Aufwand und Verzögerungen durch unklare/unvollständige Bedarfsmeldungen	
2	17	Die Qualität von Bedarfsmeldungen kann verbessert werden, indem man sicherstellt, dass das Verständnis auf allen Seiten übereinstimmt und Missverständnisse vermieden werden.	Qualitätssteigerung von Bedarfsmeldungen durch Verständnisübereinstimmung und Beseitigung von Missverständnisse	

Fall	Nr.	Paraphrase	Generalisierung	Reduktion
2	18	Unklare Informationen in Bedarfsmeldungen erhöhen den Aufwand und führen zu Nacharbeit und potenziellen Konflikten zwischen den Beteiligten.	Unklare Informationen erhöhen Aufwand und resultieren in Nacharbeiten und Konflikten	
2	19	Erfahrung ist entscheidend, um sicherzustellen, dass die Bedürfnisse und Anforderungen aller Beteiligten angemessen berücksichtigt werden.	Erfahrung ist zur Sicherstellung von Bedürfnissen entscheidend	
3	20	Bedarfsmeldungen können entweder vom Kunden direkt kommen oder von uns selbst formuliert werden.	Bedarfsmeldungen können selbst formuliert oder vom Kunden stammen	
3	21	Bedarfsmeldungen enthalten wirtschaftlich relevante Informationen wie vertragliche Konditionen, Tagessatz, Einsatzbeginn und voraussichtlicher Einsatzzeitraum.	Bedarfsmeldungs- informationen sind vertragliche Konditionen, Tagessatz, Einsatzbeginn und Einsatzzeitraum	

Fall	Nr.	Paraphrase	Generalisierung	Reduktion
3	22	Die typische Struktur einer Bedarfsmeldung beinhaltet eine Überschrift, eine allgemeine Beschreibung des Einsatzkontexts, eine Liste der geforderten Fähigkeiten und Erfahrungen sowie deren Gewichtung.	Bedarfsmeldungen enthalten eine Überschrift, eine Beschreibung des Einsatzkontexts, eine Liste der Fähigkeiten und Erfahrungen und deren Gewichtung	
3	23	Unklare oder unvollständige Bedarfsmeldungen können zu Missverständnissen führen und den Staffing-Prozess verlangsamen.	Unvollständige Bedarfsmeldungen führen zu Missverständnissen und langsamen Staffing-Prozess	
3	24	Projekte können entweder vom Kunden gesteuert werden oder von uns im Kundenauftrag durchgeführt werden.	Bedarfe sind entweder kundengesteuert oder Kundenaufträge	
3	25	Die Art der Projektvergabe (Time and Material oder Festpreis) beeinflusst die Anforderungen und den Ablauf.	Beeinflussung der Anforderungen und Ablauf durch Art der Projektvergabe	
3	26	Der Staffing-Prozess bei adesso ist manuell und nicht normiert, aber es gibt eine grobe thematische Struktur, die sich etabliert hat.	Keine feste, aber grobe Struktur zur Erfassung des Staffing-Prozesses	

Fall	Nr.	Paraphrase	Generalisierung	Reduktion
3	27	Die Schnelligkeit im Staffing-Prozess ist ein Wettbewerbsvorteil. Eine schnelle und präzise Bearbeitung von Bedarfsmeldungen ist entscheidend für den Erfolg.	schnelle und präzise Bearbeitung von Bedarfsmeldungen ist Erfolgsentscheidend	
3	28	Versierte Personen wie Delivery Manager oder Account Manager sollten den Prozess der Anforderungserhebung und Dokumentation begleiten und moderieren, um sicherzustellen, dass alle relevanten Informationen erfasst werden.	Begleitung der Bedarfsmeldungsdokumentierung durch Delivery Manager oder Account Manager	
4	29	Die Erstellung von Bedarfsmeldungen erfolgt durch Projektleiter und interne Maitre. Diese stimmen den Bedarf ab und sind für das Staffing verantwortlich.	Staffingverantwortliche sind Projektleiter und Maitre	

Fall	Nr.	Paraphrase	Generalisierung	Reduktion
4	30	Rahmenbedingungen wie Kunde, Einsatzbeginn, Laufzeit, Ort und Tagessatz sowie notwendige Skills und Aufgabenbeschreibungen sind unverzichtbar in Bedarfsmeldungen.	Unverzichtbare Bedarfsmeldungsinformationen sind Rahmenbedingungen, Einsatzbeginn, Laufzeit, Ort und Tagessatz, Skills und Aufgabenbeschreibungen	
4	31	Bedarfsmeldungen werden über Jira erfasst und mit allen beteiligten Parteien geteilt.	Bedarfsmeldungserfassung in Jira	
4	32	Unklare oder unvollständige Bedarfsmeldungen führen dazu, dass das angebotene Personal möglicherweise nicht passt oder verfügbar ist.	Unvollständige/Unklare Bedarfsmeldungen führen zu unpassenden Personal	
4	33	Technische Validierung der Daten erfolgt über Jira, weitergehende Validierung im Vier-Augen-Prinzip durch die Projektleitung.	Qualitätssicherung erfolgt über Jira und der Projektleitung	
4	34	Verbesserungsvorschlag: Strukturiere Erfassung der Daten und KI-gestützte Prüfungen für Freitextinformationen.	Qualitätssicherungsvorschlag durch strukturierte Datenerfassung und KI-gestützte Prüfungen	

Fall	Nr.	Paraphrase	Generalisierung	Reduktion
4	35	Fehlende Informationen in Bedarfsmeldungen können dazu führen, dass Mitarbeitende den Aufgaben nicht gewachsen sind, was den Projekterfolg gefährdet.	fehlende Informationen führen zu fehlplatzierten Mitarbeitern und gefährdeten Projekterfolg	
4	36	Durch intensives Lesen, Rückfragen und die Berücksichtigung aller involvierten Personen werden Bedürfnisse und Anforderungen sichergestellt.	Sicherstellung der Anforderungen und Bedürfnisse durch intensives Lesen, Rückfragen und Berücksichtigung involvierter Personen	
5	37	Bedarfsmeldungen werden normalerweise von Sales oder dem internen Maitre erstellt, basierend auf Kundenanforderungen.	Bedarfsmeldungserstellung durch Maitre oder Sales basierend auf Kundenanforderungen.	
5	38	Der Ersteller der Bedarfsmeldung legt auch die Kriterien für den benötigten Mitarbeiter fest.	Festlegung der Kriterien durch den Ersteller der Bedarfsmeldung	

Fall	Nr.	Paraphrase	Generalisierung	Reduktion
5	39	Die Bedarfsmeldungen werden in JIRA-Tickets überführt, die verschiedene Felder enthalten, wie Tagessatz, Startdatum, Volumen und Aufgaben. Es gibt auch Freitextfelder und Excel-Dokumente zur Selbstbeurteilung des Kandidaten.	Überführung der Bedarfsmeldung in Jira in die Felder Tagessatz, Startdatum, Volumen, Aufgaben, Freitext und Selbstbeurteilung	
5	40	Wichtige Informationen in einer Bedarfsmeldung sind das Startdatum, die Dauer und der Prozentsatz der Einsatzzeit des Mitarbeiters sowie die Muss- und Soll-Kriterien, um die Passgenauigkeit des Mitarbeiters sicherzustellen.	Wichtige Bedarfsmeldungsinformationen sind Startdatum, Prozentsatz, Einsatzzeit und Muss-/Soll-Kriterien	
5	41	Unklare oder unvollständige Bedarfsmeldungen führen zu falschen Zuordnungen, Kommunikationsaufwand und notwendigen Korrekturen. Dies kann auch zu einem erheblichen Overhead und Zeitaufwand durch Interviews führen.	Unklare/Unvollständige Bedarfsmeldungen führen zu falscher Zuordnung, Kommunikations-/Zeitaufwand und Korrekturen	

Fall	Nr.	Paraphrase	Generalisierung	Reduktion
5	42	Es gibt feste Felder und Zustände in Jira, die einen groben Rahmen bieten, sowie feste Rahmenbedingungen bei größeren Vereinbarungen. Es gibt jedoch keine übergreifende Qualitätssicherung, sondern es handelt sich um einen Lernprozess.	Trotz fehlender Qualitätssicherung groben Rahmen durch feste Felder in Jira und Erfahrung	
5	43	Die Bedürfnisse und Anforderungen aller Beteiligten werden durch regelmäßige Meetings, wöchentliche Jour Fixe und direkte Kommunikation berücksichtigt. Viele Bedarfe werden direkt angesprochen und nicht im JIRA eingetragen.	Bedarfe durch regelmäßigen Meetings, Jour Fixe und Kommunikation besprochen und berücksichtigt	

Tabelle A.1.: Erstellung der Kategorien aus den Experteninterviews.

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig angefertigt und mich keiner fremden Hilfe bedient sowie keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen, die wörtlich oder sinngemäß veröffentlichten oder nicht veröffentlichten Schriften und anderen Quellen entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Erklärung zu eingesetzten Hilfsmitteln

1. Korrekturservice der Fachhochschule bzw. des Fachbereichs genutzt:

☐ Ja
☒ Nein

2. Einsatz eines externen (kommerziellen) Korrekturservice:

☐ Ja
☒ Nein

3. Folgende Personen haben die Arbeit zusätzlich Korrektur gelesen:

- ...

4. Nutzung von Sprachmodellen für die Texterstellung (z.B. ChatGPT), wenn ja, welche und in welchen Abschnitten:

☐ Ja
☒ Nein

5. Sprachübersetzungstools (z.B. Google Übersetzer, DeepL), wenn ja, welche und in welchen Abschnitten:

☒ Ja
☐ Nein

- DeepL, Im Kapitel Literaturüberblick für das bessere Verständnis der Literatur
6. Einsatz von Software zur Sprachkorrektur (z.B. Grammarly), wenn ja, welche und in welchen Abschnitten:
- ☒ Ja
- ☐ Nein
- DeepL Write, Im Kapitel Literaturüberblick
7. Einsatz anderer Hilfsmittel:
-
8. Ich stimme dem möglichen Einsatz von Software zur Plagiatserkennung zu:
- ☒ Ja
- ☐ Nein

Ich bestätige, dass obige Aussagen vollständig und nach bestem Wissen ausgefüllt wurden.

Dortmund, den 25. Juli 2024

Ricardo Valente de Matos

transformation der volltextfelder in die struktur

die semantisch Eineindeutigen felder fasst mein system nicht an weil diese eins zu eins übernommen werden. deswegen konzentrieren wir uns auf die transformation der volltexter auf die stichpunkte

graph -> nicht tabellarisch

wir gucken uns den einfluss der verschiedenen variablen an.

was möchte ich zeigen -> wie kann ich die aussage zeigen. Aussagen definieren.

Literatur

- [1] S. A. Alasadi und W. S. Bhaya, “Review of data preprocessing techniques in data mining,” *Journal of Engineering and Applied Sciences*, Jg. 12, Nr. 16, S. 4102–4107, 2017.
- [2] P. Bafna, D. Pramod und A. Vaidya, “Document clustering: TF-IDF approach,” in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, IEEE, 2016, S. 61–66.
- [3] N. J. Belkin und W. B. Croft, “Information filtering and information retrieval: Two sides of the same coin?” *Communications of the ACM*, Jg. 35, Nr. 12, S. 29–38, 1992.
- [4] R. Burke, A. Felfernig und M. H. Göker, “Recommender systems: An overview,” *Ai Magazine*, Jg. 32, Nr. 3, S. 13–18, 2011.
- [5] G. Chartron und G. Kembellec, “General introduction to recommender systems,” *Recommender Systems*, S. 1–23, 2014.
- [6] M. Chiny, M. Chihab, O. Bencharef und Y. Chihab, “LSTM, VADER and TF-IDF based hybrid sentiment analysis model,” *International Journal of Advanced Computer Science and Applications*, Jg. 12, Nr. 7, 2021.
- [7] W. B. Croft, “Combining approaches to information retrieval,” in *Advances in Information Retrieval: Recent Research from the center for intelligent information retrieval*, Springer, 2000, S. 1–36.
- [8] R. Darmawan und R. S. Wahono, “Hybrid Keyword Extraction Algorithm and Cosine Similarity for Improving Sentences Cohesion in Text Summarization,” *Journal of Intelligent Systems*, Jg. 1, Nr. 2, S. 109–114, 2015.
- [9] Z. Dong, Z. Wang, J. Xu, R. Tang und J. Wen, “A brief history of recommender systems,” *arXiv preprint arXiv:2209.01860*, 2022.
- [10] T. Dresing und T. Pehl, *Praxisbuch Interview, Transkription & Analyse: Anleitungen und Regelsysteme für qualitativ Forschende*. dr dresing & pehl GmbH, 2015.
- [11] R. Horesh, K. R. Varshney und J. Yi, “Information retrieval, fusion, completion, and clustering for employee expertise estimation,” in *2016 IEEE International Conference on Big Data (Big Data)*, IEEE, 2016, S. 1385–1393.

-
- [12] N. Jain und V. Srivastava, "Data mining techniques: a survey paper," *IJRET: International Journal of Research in Engineering and Technology*, Jg. 2, Nr. 11, S. 2319–1163, 2013.
 - [13] Y. Kang, Z. Cai, C.-W. Tan, Q. Huang und H. Liu, "Natural language processing (NLP) in management research: A literature review," *Journal of Management Analytics*, Jg. 7, Nr. 2, S. 139–172, 2020.
 - [14] P. Kroha, "Preprocessing of requirements specification," in *Database and Expert Systems Applications: 11th International Conference, DEXA 2000 London, UK, September 4–8, 2000 Proceedings 11*, Springer, 2000, S. 675–684.
 - [15] D. Kumawat und V. Jain, "POS tagging approaches: A comparison," *International Journal of Computer Applications*, Jg. 118, Nr. 6, 2015.
 - [16] M. Lavin, "Analyzing documents with TF-IDF," 2019.
 - [17] J. Lu, Q. Zhang und G. Zhang, *Recommender systems: advanced developments*. World Scientific, 2020.
 - [18] P. Majumder, M. Mitra und B. Chaudhuri, "N-gram: a language independent approach to IR and NLP," in *International conference on universal knowledge and language*, Bd. 2, 2002.
 - [19] A. Mansouri, L. S. Affendey und A. Mamat, "Named entity recognition approaches," *International Journal of Computer Science and Network Security*, Jg. 8, Nr. 2, S. 339–344, 2008.
 - [20] P. Mayring, *Qualitative inhaltsanalyse*. UVK Univ.-Verl. Konstanz, 1994, Bd. 14.
 - [21] P. Mayring, "Qualitative Inhaltsanalyse—ein Beispiel für Mixed Methods," *Mixed Methods in der empirischen Bildungsforschung*, Jg. 1, S. 27–36, 2012.
 - [22] P. Mayring und T. Fenzl, *Qualitative inhaltsanalyse*. Springer, 2019.
 - [23] W. McKinney, *Python for data analysis*. Ö'Reilly Media, Inc.", 2013.
 - [24] C. K. Ogden, "Basic English: A general introduction with rules and grammar," 1930.
 - [25] E. Partalidou, E. Spyromitros-Xioufis, S. Doropoulos, S. Vologiannidis und K. Diamantaras, "Design and implementation of an open source Greek POS Tagger and Entity Recognizer using spaCy," in *IEEE/WIC/ACM International Conference on Web Intelligence*, 2019, S. 337–341.

-
- [26] M. Pfadenhauer, *At eye level: the expert interview—a talk between expert and quasi-expert*. Springer, 2009, S. 81–97.
- [27] S. Pirk, “Implementierung und Visualisierung N-Gramm-basierter Word-Clouds,” B.S. thesis, 2019.
- [28] F. Rahutomo, T. Kitasuka, M. Aritsugi u. a., “Semantic cosine similarity,” in *The 7th international student conference on advanced science and technology ICAST*, University of Seoul South Korea, Bd. 4, 2012, S. 1.
- [29] J. Ramos u. a., “Using tf-idf to determine word relevance in document queries,” in *Proceedings of the first instructional conference on machine learning*, Citeseer, Bd. 242, 2003, S. 29–48.
- [30] F. Ricci, *Recommender Systems: Models and Techniques*. 2014.
- [31] P. Sitikhu, K. Pahi, P. Thapa und S. Shakya, “A comparison of semantic similarity methods for maximum human interpretability,” in *2019 artificial intelligence for transforming business and society (AITB)*, IEEE, Bd. 1, 2019, S. 1–4.
- [32] R. Speer, J. Chin und C. Havasi, “ConceptNet 5.5: An Open Multilingual Graph of General Knowledge,” 2017, S. 4444–4451. Adresse: <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972>.
- [33] V Suhasini und N Vimala, “A Hybrid TF-IDF and N-Grams Based Feature Extraction Approach for Accurate Detection of Fake News on Twitter Data,” *Turkish Journal of Computer and Mathematics Education*, Jg. 12, Nr. 6, S. 5710–5723, 2021.
- [34] M. A. Tayal, M. Raghuwanshi und L. Malik, “Syntax parsing: Implementation using grammar-rules for English language,” in *2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies*, IEEE, 2014, S. 376–381.