Extraktion berufsspezifischer Kompetenzen nach VerBIS aus deutschsprachigen Online-Jobanzeigen

Bertelsmann Stiftung

Version
1
Autoren
Rahkakavee Baskaran, Johannes Müller
Annotation und inhaltliche Beratung
Alketa Hiseni, André Zucker, Angela Pilgrim, Daniel Bensel, Gunvald Herdin, Moritz Güth, Tobias Ortmann, Ines Galla, Jana Fingerhut, Kerstin Ködding, Leonie Holtkamp, Tanja Meyer zur Heyde, Wiebke Lüpkes
Erstellt am
09. Juni 2022
Letzte Änderung am
11. Juli 2024

&effect

Inhalt

Abkürzungsverzeichnis

BA	Bundesagentur für die Arbeit
BERT	Bidirectional-Encoding-Representation from Transformers
ID	Identifikation
KldB	Klassifikation der Berufe 2010 – überarbeitet Fassung 2020
LR	Logistische Regression
ML	Machine Learning
NER	Named Entity Recognition
VerBIS	Vermittlungs-, Beratungs- und Informationssystem der Bundesagentur für Arbeit

Abbildungsverzeichnis

Tabellenverzeichnis

1 Zusammenfassung

Die Extraktions-Pipeline dient der Identifizierung und Kategorisierung berufsspezifischer Kompetenzen aus deutschsprachigen Stellenausschreibungen gemäß der VerBIS (Vermittlungs-, Beratungs- und Informationssystem der Bundesagentur für Arbeit) Taxonomie der Bundesagentur für die Arbeit (BA). Der zweistufige Prozess umfasst die Phasen der Extraktion und Normalisierung der Kompetenzen nach VerBIS.

In der ersten Stage des Prozesses wird unter Verwendung eines Segementationsalgorithmus, sowie eines Klassifikationsalgorithmus basierend auf Bidirectional-Encoding-Representation from Transformers (BERT) und Logistic Regression (LR) berufsspezifische Kompetenzen aus den Stellenausschreibungen extrahiert. In der zweiten Stufe werden die extrahierten Kompetenzen mithilfe eines Matchers, sowie einem BERT Modell den VerBIS Klassen zugeordnet. Die Pipeline wird auf Daten von Textkernel trainiert und evaluiert.

Die Gesamtperformance der Pipeline wird basierend auf gängigen Metriken für die einzelnen Komponenten sowie auf einer grafischen Analyse beurteilt. Abschließend werden verschiedene Optionen zur Filterung der extrahierten Daten empfohlen, um die Qualität der Daten für die Analyse zu verbessern. Dies wird anhand von Evaluationsdatensätzen belegt.

2 Methoden

Die Methoden werden in zwei Punkten dargelegt. Zunächst erfolgt eine Beschreibung der verwendeten Taxonomien und der Jobausschreibungsdaten. Daran schließt sich die Darlegung der Extraktionspipeline mit den einzelnen Komponenten und den entsprechenden Algorithmen an.

2.1 Taxonomie

Die VerBIS Taxonomie umfasst insgesamt 8.256 berufsspezifische Kompetenzen. [1] Die Kompetenzen sind in Gruppen organisiert. Diese Gruppen bilden übergreifende Berufsbereiche wie beispielsweise „Floristik“ oder „Forstwirtschaft, Jagd“ oder thematisch zusammengehörende Kompetenzen wie „Prüflizenzen“ ab. Die Taxonomie verfügt über verschiedene Identifikationen (ID) für die Gruppen und die Kompetenzen selbst. Die für die Extraktionspipeline verwendete ID setzt sich wie folgt zusammen: Sie ist durch einen Bindestrich in zwei Teile gegliedert. Der erste Teil weist auf den Gruppennamen hin, der zweite Teil steht für die berufsspezifische Kompetenz selbst. So kann, beispielsweise, aus der ID „K 0005-004“ entnommen werden, dass die Kompetenz der Gruppe „K 0005“, d.h. „Landwirtschaftliche Tierhaltung, Tierzucht“ angehört. Die „004“ steht für „Bienenzucht, -haltung“. Insgesamt gibt es 276 Gruppen in der Taxonomie.

Neben dem Label und den Gruppennamen findet sich in der Taxonomie für jede berufsspezifische Kompetenz eine Liste an Suchwörtern, die auf die Kompetenzen zurückzuführen sind. Für das oben genannte Beispiel „Bienenzucht, -haltung“ gibt es beispielsweise die Suchwörter „Imkerei“ oder „Honigerzeugung“. Insgesamt gibt es 32.737 eindeutige Suchwörter. Zusätzlich wird für jedes Suchwort eine Prior Probability angegeben. Die Prior Probability ist ein Maßstab für die Anzahl der Kompetenzen, in denen ein Suchwort vorkommt. Berechnet wird die Prior Probability als Inverse der Anzahl der Kompetenzen, in der ein Suchwort vorkommt. Eine Prior Probability von 1 bedeutet demnach, dass ein Suchwort nur für diese Kompetenz vorkommt. Von den 32.737 Suchwörtern sind 29.957 Suchwörter nur zu einer Kompetenz zugeordnet.

Trotz einer Vielzahl an repräsentierten Kompetenzen hat die VerBIS Taxonomie einige Grenzen. Ein wesentlicher Punkt ist die Vergleichbarkeit der Kompetenzgruppen. Diese Gruppen sind thematisch unterschiedlich definiert. Beispielsweise stellt Floristik eine spezifische Berufsgruppe dar, während die Gruppe Prüflizenzen verschiedene Berufsgruppen umfassen kann. Infolgedessen sind manche Kompetenzen nur im Kontext ihrer Gruppe interpretierbar, was Herausforderungen für Machine Learning (ML) Modelle in Bezug auf die Normalisierung mit sich bringt. Eine weitere Problematik betrifft die Definition von berufsspezifischen Kompetenzen in der Taxonomie. Es stellt sich die Frage, welche Kompetenzen darin enthalten sein sollten. Beispielsweise gibt es umstrittene Gruppen wie Arbeitsorte und Sprachen. Letztlich ist noch die Aktualität der Taxonomie zu nennen. Ändern sich die Anforderungen auf dem Arbeitsmarkt, was beispielsweise in technischen Berufen oft vorkommt, muss die Taxonomie und die darauf basierenden ML-Modelle entsprechend angepasst werden. Die Taxonomie wird laufend aktualisiert.

[1] Die VerBIS Taxonomie kann im Download Portal der BA heruntergeladen werden (Bundesagentur für Arbeit, 2024a)

Die Herausforderungen, die mit der Taxonomie einhergehen, müssen bei der Entwicklung und Anwendung der Pipeline berücksichtigt werden. In Bezug auf die Definition von berufsspezifischen Kompetenzen wird folgende Maßnahme ergriffen: Die Gruppen Arbeitsorte, Personengruppen und Sprachkenntnisse, sowie alle darunterfallenden Kompetenzen werden ausgeschlossen. Zusätzlich werden die Kompetenzen Aus- und Fortbildung, Praktikantentätigkeit und Anzeigengeschäft herausgefiltert. Die ursprüngliche Anzahl an Klassen reduziert sich somit von den ursprünglichen 8.256 Kompetenzen auf 7.968 Kompetenzen. Die Anzahl der Gruppe reduziert sich auf 273. Die Reduktion hat auch den Vorteil, dass insgesamt weniger Klassen vorhergesagt werden müssen, was sich positiv auf die Performance auswirken kann.

2.2 Jobausschreibungsdaten

Die Trainingsdaten stammen von Textkernel (Textkernel 2022). Diese bestehen aus den Volltexten der Ausschreibungen und einer Segmentierung des Volltextes. Segmentiert werden die Texte in die Kategorien „job_description“, „candidate_description“, „application_description“, „employer_description“ und „conditions_descriptions“. Das Segment „job_description“, im weiteren Verlauf Jobbeschreibung genannt, enthält dabei die für die Extraktion von berufsspezifischen Kompetenzen am relevantesten Informationen. Dieses Segment soll deshalb für das Training der Modelle der Extraktion verwendet werden. Nicht alle Ausschreibungen enthalten allerdings eine Jobbeschreibung, was die Generalisierbarkeit der Modelle beeinflusst. Entsprechende Strategien zur Bewältigung dieser Herausforderung sowie die Grenzen der Modelle werden in der Methodenbeschreibung und der Vorstellung der Modelle näher erläutert.

2.3 Extraktionspipeline

Die Extraktionspipeline besteht aus einem zweistufigen Prozess: Extraktion und Normalisierung. Jede Stage verfügt wiederrum über verschiedene Komponenten, die nacheinander ausgeführt werden. Abbildung 1 zeigt eine Übersicht über die einzelnen Schritte. Der Input der Pipeline sind die Jobbeschreibungen der Stellenausschreibungen. Ist diese nicht vorhanden, wird der Volltext der Stellenausschreibung verwendet.

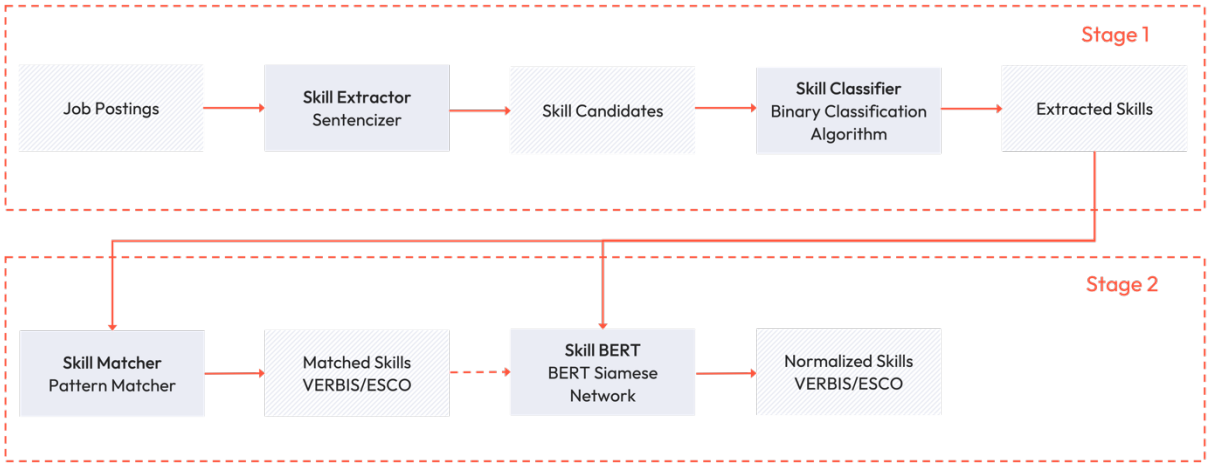


Abbildung 1: Extraktionspipeline

Stage 1. Der Input wird zunächst an die Skill Extractor-Komponente weitergeleitet. Diese Komponente splittet den Input in einzelne Sätze, welche als Kompetenzkandidaten dienen. Im nächsten Schritt werden diese Kandidaten mithilfe eines Klassifizierungsansatzes bewertet. Diese Komponente wird im Folgenden als Skill Classifier bezeichnet. Enthält ein Kandidat eine Kompetenz, wird er zur Liste der extrahierten Kompetenzen hinzugefügt und in Stage 2 weitergegeben. Diese Kandidaten werden im Nachfolgenden als Kompetenzsatz bezeichnet. Enthält der Kandidat keine Kompetenz, wird er entfernt. Die einzelnen Komponenten basieren auf verschiedenen Natural Language Processing (NLP) Techniken, welche im Folgenden erläutert und auf die jeweiligen Komponenten bezogen werden.

Sentence Segmentation
Dependency Parsing

Der Skill Extractor baut auf einer Satzsegmentierung mit Dependency Parsing auf. Bei der Satzsegmentierung werden Satzgrenzen zwischen Wörtern identifiziert, um einen Text in einzelne Sätze zu trennen (Palmer, David D. 2000). Aus theoretischer Perspektive kann Segmentation entweder regelbasiert oder mithilfe komplexerer Methoden, wie Dependency Parsing erreicht werden. Während beim regelbasierten Ansatz nur an zuvor definierten Regeln, wie „am Punkt trennen“, segmentiert werden kann, kann mithilfe von Dependency Parsing, welcher grammatikalische Strukturen zwischen den Wörtern eines Satzes herstellt, auch eine komplexere Segmentierung von Sätzen ermöglicht werden (Yuan, Y, 2019). Ziel der Segmentierung für die Pipeline ist es die Ausschreibung so in einzelne Segmente aufzuteilen, dass für jedes Segment entschieden werden kann, ob eine Kompetenz vorhanden ist oder nicht. Da Ausschreibungen oftmals in unterschiedlichen Formaten vorhanden sind und das Ziel eine gewisse Flexibilität in der Segmentierung mit sich bringen muss, eignet sich ein ausschließlich regelbasierter Ansatz nicht. Für

die Satzsegmentierung wird deshalb ein Dependency Parser verwendet. Allerdings wird, um die Segmentation zu verbessern, zwei Regeln hinzugefügt, die jede Wortsequenz nach einem „+“ oder einem „*“ ebenfalls trennen. Diese Regeln ergeben sich aus der Häufigkeit, mit den Kompetenzen in Ausschreibungen aufgelistet werden, die aber durch den Dependency Parser nicht erkannt werden können.
Aus praktischer Sicht wird für den Skill Extractor der Dependency Parser des vortrainierten „de_core_news_lg“ Modell der deutschen Sprache von Explosion.ai verwendet, der sich mithilfe der Python Spacy Library (Version 3.3.0) implementieren lässt. Spacy verwendet einen „transition-based dependency parser“, welcher auf einem klassischen shift-reduce parsing aufbaut (Spacy o.D., Jurafsky and Martin 2021), nach der Methode von Honnibal et al. (2015).

Feature Extraction
BERT

Für den Skill Classifier müssen zunächst Features aus den segmentierten Sätzen extrahiert werden. Hierfür wird ein BERT-Modell trainiert. BERT verwendet eine „multi-layer bidirectional transformer encoder“ Architektur mit „self-attention“ Mechanismus. Der „self-attention“ Mechanismus verbessert die Repräsentation von Wörtern (Vaswani et al.2017), wodurch BERT im Vergleich zu anderen Vektorisierungen die Ähnlichkeit zwischen Wörtern besser erlernen kann. Für den Skill Classifier wird das trainierte BERT-Modell als Feature Extractor verwendet. Mithilfe eines vortrainierten Modells für die deutsche Sprache, dem „dbmdz/bert-base-german-cased“ (Bayerische Staatsbibliothek, 2019) Modell, sowie einem pooling layer wird das BERT-Modell erst konstruiert und in einem zweiten Schritt ein Fine Tuning angewandt. Das Fine Tuning ist für die erste Stufe nicht von großer Relevanz. Da für die Stage 2 ebenfalls ein BERT-Modell gebraucht wird, wird aus Effizienzgründen das gleiche BERT-Modell verwendet. Der Prozess des Fine Tunings wird entsprechend in der Stage 2 näher erläutert.

Binary Classification
Logistic Regression

Der zweite Teil des Skill Classifier besteht aus einer Logistischen Regression (LR). LR ist ein generalisiertes Modell, welches für die Lösung von Klassifikationsprobleme angewandt wird. Mithilfe der LR können die Kandidaten danach klassifiziert werden, ob sie eine Kompetenz enthalten oder nicht enthalten Kompetenzen. Die hier verwendete LR optimiert das Klassifikationsproblem mit einem „limited memory Broyden-Fletcher-Goldfarb-Shannon“ Algorithmus

und einem „L2 Penalty“ Regularisierung (Fei, Y et al. 2014, Jurafsky and Martin 2021). Der Algorithmus wird unter Verwendung der „sklearn“ Library implementiert (Pedregosa et al., 2011).

Stage 2. Der Output aus Stage 1 wird an zwei Komponenten weitergegeben: Skill Matcher und Skill BERT. Der Skill Matcher prüft für jeden Kompetenzsatz, ob eine VerBIS Kompetenz oder ein Suchwort einer VerBIS Kompetenz wortwörtlich darin vorkommt. Ist das der Fall, wird die entsprechende VerBIS Kompetenz zu der Liste von normalisierten Skills für die Ausschreibung hinzugefügt. Skill BERT prüft für jeden Kompetenzsatz, welche Kompetenz aus der VerBIS Taxonomie am ähnlichsten zu dem Kompetenzsatz ist. Die entsprechende Kompetenz wird zu der Liste der normalisierten Kompetenzen hinzugefügt.

Wie eingangs in diesem Kapitel beschrieben, gibt es nicht für jede Stellenausschreibung eine Jobbeschreibung. Für diese Stellenausschreibungen muss dann der Volltext verwendet werden. Der Skill Classifier ist jedoch ausschließlich auf den Jobbeschreibungen der Stellenausschreibungen trainiert. Es ist deshalb zu erwarten, dass die Performance auf Volltextdaten schlechter ist. Zur Verbesserung der Performance wird deshalb bei Stellenausschreibungen, bei denen der Volltext der Input ist, geprüft, ob der Skill Matcher ein Ergebnis findet. Der Skill Matcher ordnet nur Kompetenzen nach VerBIS zu, wenn diese wortwörtlich vorkommen, während Skill BERT jedem Kompetenzsatz auf jeden Fall eine Kompetenz zuordnet. Um eine bessere Einschätzung zu haben, ob der Satz tatsächlich ein Kompetenzsatz ist, wird der Skill Matcher als eine Art Verifizierung des Ergebnisses des Skill Classifiers verwendet. Findet der Skill Matcher keinen Satz, dann wird der Satz ausgeschlossen. Das Ergebnis des Skill BERT für diesen Kompetenzsatz wird dann nicht berücksichtigt. Das beeinflusst zwar den Recall, weil es Kompetenzsätze gibt, die nicht wortwörtlich VerBIS-Kompetenzen enthalten, aber eventuell indirekt eine Kompetenz, die Skill BERT finden würde. Es wird jedoch davon ausgegangen, dass die Präzision für Volltexte dadurch besser ist, weil somit potenziell falsche Kompetenzsätze ausgeschlossen werden. Wie in Stufe 1 greifen die einzelnen Komponenten auf verschiedene Nature Language Processing Methoden zurück, die folgend erläutert werden.

Named-Entity-Recognition
Rule Based

Der Skill Matcher besteht aus einem regelbasierten Named Entity Recognition (NER) Modell. Mithilfe von regelbasierten NER-Modellen können innerhalb von Texten Wörter oder Sequenzen von Wörtern erkannt und extrahiert werden. Ziel ist es, in den klassifizierten Kompetenzsätzen Kompetenzen aus der VerBIS Taxonomie zu finden. Dies kann mithilfe des PhraseMatchers von Spacy (Spacy o.D.) praktisch umgesetzt werden. Der PhraseMatcher ist dann sinnvoll, wenn man eine lange Liste von Terminologien hat, nach der ein Text oder Satz

durchsucht werden soll. Dabei werden sowohl einzelne Wörter als auch Sequenzen von Wörtern als Patterns akzeptiert. Die hier verwendete Liste von Terminologien besteht aus zwei Gruppen. Erstens werden die VerBIS Kompetenzen selbst als Patterns verwendet. Zweitens werden alle Suchwörter von VerBIS hinzugefügt, die eine Prior Probability von 1 ausweisen. So kann sichergestellt werden, dass der Matcher nur für diese Kompetenz spezifische Suchwörter zuordnet und nicht Kompetenzübergreifende Suchwörter. Die Suchwörter werden im Prozess des Matchings auf die Kompetenzen aufgelöst, sodass der Output des Skill Matcher eine Liste von VerBIS Kompetenzen bleibt.

Text Similarity
Siamese Network Architecture

Ziel des Skill BERT ist es, die Kompetenzsätze aus Stage 1 auf die VerBIS Taxonomie aufzulösen. Das BERT-Modell aus Stufe 1 vergleicht hierfür jeden Kompetenzsatz mit der Liste der VerBIS Kompetenzen. Die Kompetenz, die dem Satz am ähnlichsten ist, wird dann zur Liste der normalisierten Kompetenzen nach VerBIS hinzugefügt. Um die Ähnlichkeit berechnen zu können, ist es notwendig, das vortrainierte BERT-Modell so zu trainieren, dass Kompetenzen, die in der VerBIS Taxonomie ähnlich zueinander sind, durch das Modell erkannt werden. Hierfür wird das vortrainierte BERT-Modell mithilfe von Fine Tuning auf das beschriebene Domain-Wissen trainiert. Das Fine Tuning folgt der Methode von Reimers und Gurevych (2019). Unter Verwendung einer sogenannten Siamese Network Architektur werden Paare von ähnlichen und nicht ähnlichen Kompetenzen aus der VerBIS Taxonomie mit einem Score in das vortrainierte Modell gegeben. Dabei gilt je höher der Score ist, desto ähnlicher ist das Kompetenzpaar. Die Kompetenzpaare werden dann durch das Netzwerk des BERT-Modell geleitet, die die Ähnlichkeit mithilfe der Ähnlichkeitsmetrik „Kosinus-Ähnlichkeit“ berechnet und mit dem Score verglichen. Auf diese Weise können die Gewichte im BERT-Modell entsprechend angepasst werden und so die Ähnlichkeiten der Input Daten erlernt werden. Für die Implementierung des Fine Tunings wird die Sentence-BERT Library verwendet (Reimers und Gurevych, 2019).

3 Annotationen

3.1 Annotation I: Lokalisierung

Die Datengrundlage für die Entwicklung des Skill Classifiers bilden die Jobbeschreibungen der Stellenausschreibungen. Für das Labelling werden die Jobbeschreibungen mit dem Skill Extractor segmentiert und anschließend den Annotator:innen bereitgestellt. Die Annotator:innen labeln dann die Sätze danach, ob ein Satz eine Kompetenz enthält (Label „skill“) oder keine Kompetenz enthält (Label „no skill“). Die Daten für das Labelling stammen dabei aus zwei Textkerneldatensätzen. Während die Stellenausschreibungen für den ersten Datensatz zufällig ausgewählt werden, werden für den zweiten Datensatz aus jedem der 21 Industriebereichen, Stellenausschreibungen gezogen. Betrachtet man berufsspezifische Kompetenzen in Stellenausschreibungen, haben diese unabhängig vom Beruf einen ähnlichen Aufbau und ähnliche Formulierungen. Die meisten Stellenausschreibungen enthalten beispielsweise einen Abschnitt, in dem die Kompetenzen in Stichpunkten aufgelistet sind. Diese Kompetenzen sind oft als Nominalphrasen formuliert und enthalten viele Aufzählungen. Aufgrund dieses Aufbaus, besteht nicht die Notwendigkeit, einen für alle Berufe repräsentativen Datensatz, zu wählen; eine Stichprobe ist ausreichend.

Für eine gute Performance des Skill Classifiers spielt die Qualität des Labellings eine große Rolle. Es muss sichergestellt werden, dass die Annotator:innen konsistent labeln. Für konsistente Daten ist es wichtig, dass die Annotator:innen ein gleiches Verständnis für berufsspezifische Kompetenzen haben. Dies kann durch Annotationsregeln sichergestellt werden. Zudem sollte durch Interrater-Reliabilität Maße sichergestellt werden, dass die annotierten Daten übereinstimmen. Im Folgenden wird der Annotationsprozess im Detail erläutert:

Annotationsregeln. Die Frage nach der Definition berufsspezifischer Kompetenzen lässt sich nicht leicht beantworten. Eine Herausforderung ist die Definition der Grenzen von berufsspezifischen Kompetenzen. Zählen etwa Abschlüsse oder Jobtitel als berufsspezifische Kompetenz, da sich aus den Titeln Kompetenzen ableiten lassen? Eine weitere Frage ist, wie detailliert Kompetenzen sein sollen. Fasst man einzelne Wörter darunter oder ganze Satzteile? Reicht beispielsweise „Software“ oder „Applikation“ als Wort schon aus? Letztlich stellt sich noch die Frage nach dem Grad der Kompetenz. Ist eine Kompetenz einer Stellenausschreibung auch dann als solche zu deklarieren, wenn es während Tätigkeit erst erlernt wird? Beispielsweise dann, wenn die Stellenausschreibung sich auf eine Ausbildung bezieht? Basierend auf diesen Herausforderungen werden die folgenden Regeln für das Labelling festgelegt, um ein inkonsistentes Labelling zu vermeiden:

Alle Kompetenzen, die berufsspezifischen Wörter, wie beispielsweise Maschinen- und Anlagebau oder Prüfservice, enthalten, werden als berufsspezifische Kompetenzen gewertet. Weiterhin werden IT-Kenntnisse, Jobtitel und Abschlüsse hinzugenommen, da sie auf berufsspezifische Kompetenzen schließen. Der Grad der Kompetenz spielt dabei keine Rolle. D.h. Kompetenzen, die noch erlernt werden, als Erfahrung deklarierte Kompetenzen usw. werden ebenfalls als Kompetenz annotiert. Aufgrund der Verwendung ganzer Sätze für das Labelling erübrigt sich die Frage, wie detailliert eine Kompetenz sein muss. Abgegrenzt werden hiervon Soft Skills, bzw. Transversale Kompetenzen, Sprachkenntnisse, sowie lediglich Teilnahme, Mitarbeit, Zunahme ohne Nennung eines berufsspezifischen Wortes. Weiterhin sind Innendienst, sowie Außendienst ohne weitere Spezifikation keine berufsspezifischen Kompetenzen. Unternehmensbeschreibungen, aus denen möglicherweise Kompetenzen abgeleitet werden könnten, werden ebenfalls nicht als Kompetenz gelabelt. Dasselbe gilt für Schulungen, die keine Beschreibung enthalten, welche berufliche Kompetenz erlernt werden soll.

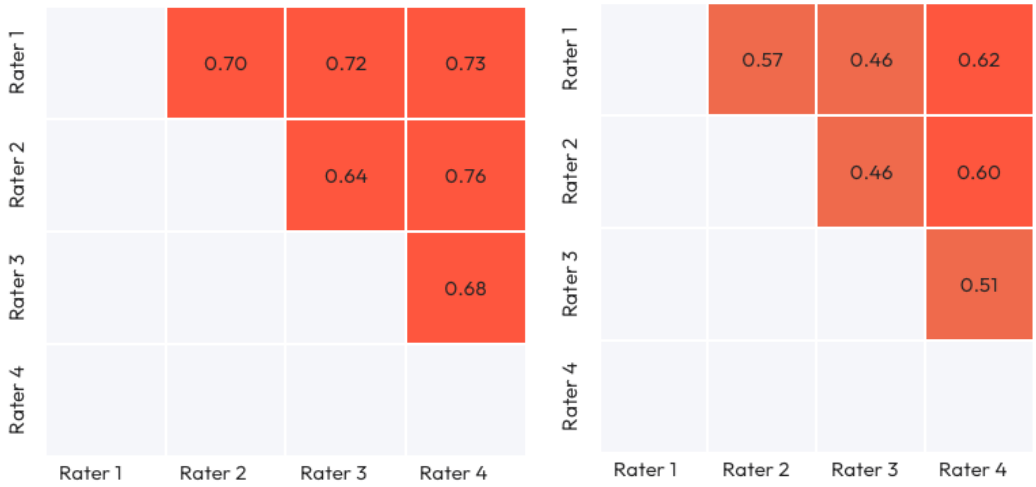
Interrater-Reliabilität. Neben der Findung einer geeigneten Definition wurde zur Sicherstellung der Datenkonsistenz in Runden gelabelt, mit jeweils einem überschneidenden Datenanteil. Der Prozess bestand aus einer Testrunde, sowie zwei Runden, die die Trainingsdaten für den Algorithmus bilden. Nach jeder Runde wurde für den überschneidenden Teil das Agreement zwischen den Annotator:innen berechnet sowie Unklarheiten bei den Daten der einzelnen Annotator:innen besprochen. Als Metrik wird das Cohens Kappa herangezogen. Das Cohens Kappa wird jeweils zwischen zwei Annotator:innen berechnet. Für einen Gesamteindruck über alle Annotator:innen hinweg wird der Durchschnitt der einzelnen Werte berechnet. Formal lässt sich das wie folgt ausdrücken:

$$\sum_{i=1}^{n} \frac{p_{0i} - p_{ci}}{1 - p_{ci}}$$

wobei für jedes Paar von Annotator:innen i der Anteil an Übereinstimmung p_0, sowie der Anteil an erwarteter Übereinstimmung per Zufall p_c berechnet wird. Cohens Kappa kann maximal einen Wert von 1 und minimal einen Wert von -1 annehmen (Cohen, J. 1960)

Annotationsrunden. Für Runde 1 und Runde 2 werden jeweils mit vier Annotator:innen gelabelt. Dabei werden pro Annotator:innen 300 Sätze annotiert, von denen über alle Annotator:innen hinweg 50 Sätze übereinstimmend sind. Abbildung 2 bildet die Übereinstimmungen zwischen den einzelnen Ratern:innen für beide Runde ab. Die Ergebnisse des durchschnittliche Cohens Kappa Wert unterscheiden sich in den Runden (0.71 und 0.54). Nach einer Besprechung der Inkonsistenzen, welche händisch aufgelöst werden, werden alle Annotationen in einen Datensatz überführt.

Abbildung 2: Cohens Kappa Annotation Skill Classifier Trainigsdatensatz

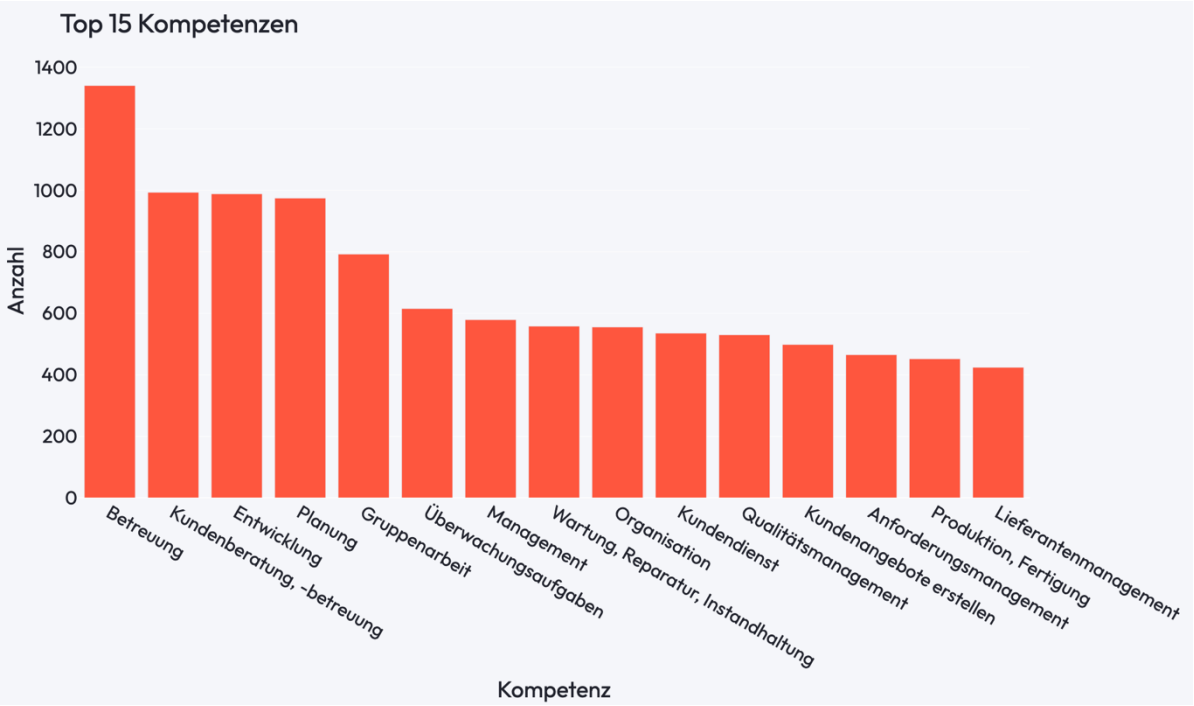


3.2 Annotation II: Normalisierung

Neben den Validierungsdaten für das Training und das Fine-Tuning des Modells, wird für die Evaluation einzelner Komponenten des Modells ein Testdatensatz mithilfe einer zufälligen Stichprobe der Textkernel-Daten erstellt. Dies ist hauptsächlich notwendig, weil keine Validierungsdaten für die Evaluation des Skill BERTs vorhanden sind. Die Annotationen bzw. die Evaluation des Testdatensatzes erfolgt händisch von bis zu sieben Annotator:innen.

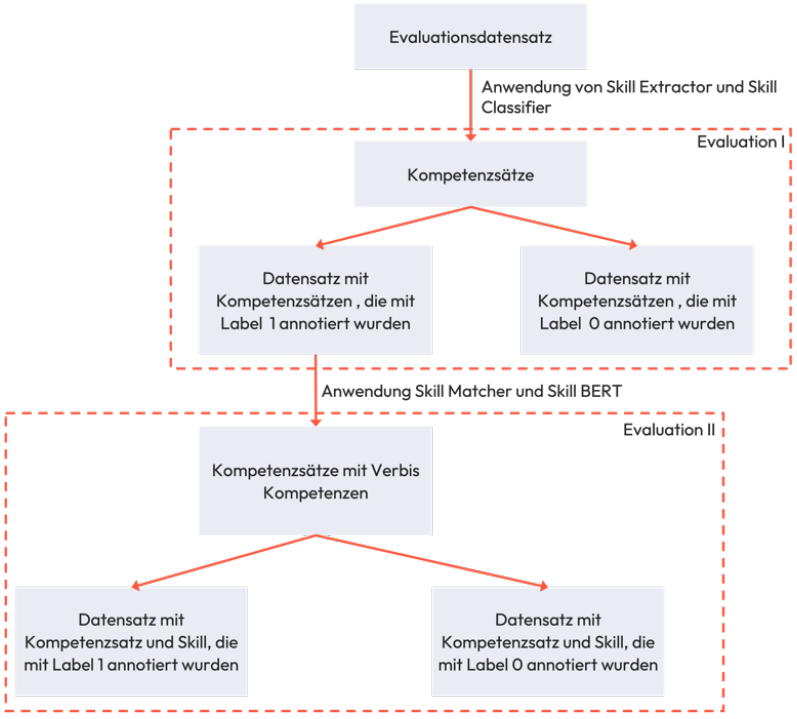
Repräsentation des Datensatzes. Der Datensatz beinhaltet 9.844 Postings. Für die Erstellung wird der Datensatz auf der Extraktionspipeline angewendet. Es werden die Ergebnisse des Skill Classifiers (Kompetenzsätze) und des Skill BERTs im Datensatz erfasst. Insgesamt sind das in dem extrahierten Datensatz über alle Stellenausschreibungen hinweg 68.664 Kompetenzsätze. Von den Kompetenzen sind 4.060 individuelle Fachkompetenzen. Das entspricht knapp 51% aller Kompetenzen in der VerBIS Taxonomie. Von den 273 Gruppen der VerBIS Taxonomie werden 269 Gruppen im Datensatz abgedeckt. Da die VerBIS Kompetenzen zum Teil sehr spezifische Kompetenzen abbilden und nahezu alle Gruppen der VerBIS Taxonomie im Datensatz repräsentativ sind, wird dies als ausreichend angenommen für den Testdatensatz. Die Top 15 am häufigsten vorkommenden Kompetenzen machen 15% des Datensatzes aus. Dies wird in der Interpretation der Evaluationsergebnisse entsprechend berücksichtigt. Abbildung 3 zeigt die Top 15 Kompetenzen und Anzahl der Vorhersagen in dem Datensatz:

Abbildung 3: Verteilung der Kompetenzen Evaluationsdatensatz



Annotationsprozess. Die Erstellung des Testdatensatzes erfolgt in einem händisch aufwendigen Prozess mit zwei Evaluationsrunden. Abbildung 4 zeigt den Prozess der Evaluation auf. In der ersten Evaluation wird der Skill Classifier bewertet, in der zweiten Evaluation die Ergebnisse des Skill Matchers und des Skill BERTs. Im Folgenden wird der Evaluationsprozess für beide Komponenten im Einzelnen beschrieben.

Abbildung 4: Evaluationsprozess Testdatensatz



Annotation II-1. Wie in Kapitel 2.3 Stage 1 beschrieben, wird jede Stellenausschreibung in einzelne Sätze gesplittet und jeder Satz wird mithilfe des Skill Classifiers danach klassifiziert, ob der Satz eine Kompetenz enthält oder nicht. Sätze, die von dem SkillClassifier mit „kein Skill“ bewertet werden, werden für die Evaluation nicht berücksichtigt, um den Annotationsaufwand geringer zu halten. Die Evaluation kann auf zwei Wegen stattfinden. In der ersten Option werden alle Sätze, die zu einer Stellenausschreibung gehören, gemeinsam bewertet und basierend auf einer Regel entschieden, ob die Stellenausschreibung als Ganzes korrekt eingestuft wird. In der zweiten Option wird die Evaluation auf Satzebene durchgeführt und jeder Satz einzeln bewertet. Aufgrund der Möglichkeit, dass die Evaluation auf Satzebene auch aggregiert werden kann, um die erstere Möglichkeit abdecken zu können, wird die zweite Variante bevorzugt. Aus dem Datensatz ergeben sich insgesamt 63.652 Einzelsätze. Ein Ausschnitt aus dem evaluierten Datensatz ist in Tabelle 1 dargestellt. Der Tabelle 1 ist zu entnehmen, dass die Annotator:innen jeden Satz mit 1 und 0 bewerten. Auf Basis dieser Evaluation wird dann die Performance der Komponente bewertet.

Tabelle 1: Ausschnitt Testdatensatz für die Evaluation des Skill Classifiers

job_id	sentence	evaluation_classifier
1	Sie beraten Bestandskunden freundlich und mit Charme zu Produkterneuerungen und Zusatzprodukten.	1
1	Sie bleiben immer auf dem Laufenden durch regelmÃ¤ÃŸige Schulungen und die â¢ kontinuierliche Weiterentwicklung der Produkte unserer Auftraggeber (sic)	0
2	Gabelstaplerfahrer / in Gabelstaplerfahrer/	1
2	Für unseren Kunden in Lorsch suchen wir ab sofort mehrere Staplerfahrer m/w/d Ihre Aufgaben: - Be- und Entladen von LKW - Zusammenstellen von Aufträgen - Allgemeine Lagertätigkeiten	1

Der Annotationsprozess ist ähnlich strukturiert zu der Erstellung des Trainingsdatensatzes. Auch werden zur Sicherstellung der Datenkonsistenz in jeder Runde an mehrere Annotator:innen die gleichen Kompetenzsätze vergeben, sodass die Interrater-Reliabilität gemessen werden kann. Der Prozess besteht aus zwei Testrunden und elf Annotationsrunden. Aufgrund des längeren Zeitraums der Annotationsrunden sind nicht alle Annotator:innen für alle Runden verfügbar, sodass die Anzahl der Annotator:innen innerhalb der Runden sich unterscheidet. Auch muss aufgrund der Ergebnisse der Interrater-Reliabilität teilweise Sätze neu annotiert und für die Auflösung von nicht übereinstimmenden Annotationen eine händische Auflösung am Ende von allen Annotationsrunden durchgeführt werden. Im Appendix A findet sich eine Übersicht mit allen Annotationsrunden, Anzahl der Annotationen und Interrater-Reliabilitäten.

Annotation II-2. Die Kompetenzsätze, die in der Annotation II-1 als korrekt eingestuft werden, bilden die Grundlage für Annotation II-2. Dies lässt sich wie folgt begründen: Wie in Stage 2 des Kapitels 2.3 beschrieben, werden alle Sätze, die nach dem Skill Classifier eine Kompetenz enthalten, in die Skill Matcher und die Skill BERT Komponente übergeben. Während der Skill Matcher für einen Kompetenzsatz nur dann VerBIS Kompetenzen zuordnet, wenn eines der definierten Patterns im Satz zu finden ist, ordnet Skill BERT auf Basis von Semantischer Suche garantiert eine Kompetenz zu. Falls die Einschätzung des Skill Classifier falsch ist, d.h. es ist keine Kompetenz im Satz vorhanden, muss davon ausgegangen werden, dass auch die Zuordnung von Skill BERT falsch ist. Aufgrund dieser Abhängigkeit werden nur Sätze berücksichtigt, die auch korrekt klassifiziert wurden. Bei allen anderen Kompetenzsätzen wird davon ausgegangen, dass diese ohnehin als falsch evaluiert werden würden. Zudem sinkt damit die Anzahl der Sätze, die evaluiert werden müssen, erheblich.

Die Evaluation der Resultate kann auf drei Ebenen durchgeführt werden. Die oberste Ebene wäre auf Stellenausschreibungsebene. Wie bei dem Skill Classifier können alle Resultate von allen Sätzen einer Stellenausschreibung zusammengefasst und dann bewertet werden. Ähnlich könnte man auf Satzebene alle Resultate für einen Satz gebündelt betrachten. Stimmen alle Sätze überein, wird der Satz als korrekt evaluiert. Auf der untersten Ebene müsste man die Resultate nochmals splitten für die einzelnen Sätze und dann auf Einzelkompetenzebene die Sätze bewerten. Wir sehen bei der Evaluierung auf Stellenausschreibungs- und Satzebene die Herausforderung, dass Annotator:innen schnell den Überblick verlieren könnten bei einer hohen Anzahl von Kompetenzen und Sätzen. Wie bei dem Skill Classifier sehen wir auch den Vorteil der Möglichkeit der Hochaggregation der Evaluation, wenn die möglichst unterste Ebene evaluiert wird. Ein Nachteil bei der Evaluation auf Einzelkompetenzebene ist allerdings die Anzahl an Sätzen, die evaluiert werden. Da allerdings durch die erste Evaluationsrunde alle Sätze ausgeschlossen werden können, die mit 0 bewertet werden, bleibt hier eine ähnliche Anzahl an Sätzen übrig. Auf Basis dieser Abwägung wird die Evaluation auf Kompetenzebene durchgeführt. Die Evaluation sieht entsprechend wie folgt aus:

Tabelle 2: Ausschnitt Testdatensatz für die Evaluation des Skill Matchers und Skill BERTs

job_id	sentence	result	evaluation_classifier
1	Zudem lernst du einen angrenzenden Schnittstellenbereich deiner Wahl in der EDEKA-Zentrale kennen und erlebst das praktische Arbeitsleben im Gro- und Einzelhandel direkt vor Ort.	Einzelhandel	1
1	Zudem lernst du einen angrenzenden Schnittstellenbereich deiner Wahl in der EDEKA-Zentrale kennen und erlebst das praktische Arbeitsleben im Gro- und Einzelhandel direkt vor Ort.	Berufsberatung	0
2	Allgemeine Stallarbeit	Stallarbeit	1
2	- Eiersortierung - Tiere füttern	Tiere füttern	1

Der Annotationsprozess ist gleich strukturiert wie in der ersten Evaluation. Der Prozess besteht aus zehn Annotationsrunden. Auch hier sind nicht alle Annotator:innen für alle Runden verfügbar. Der Prozess bestand aus insgesamt 10 Runden. Aufgrund der Komplexität des Tasks sind die Interrater-Reliabilität Werte in manchen Runden nicht ausreichend. Auch sind einige Kompetenzsätze ungültig (falsch annotiert) oder gar nicht annotiert. Das Cohens Kappa ist teils auch negativ, was für eine starke Unstimmigkeit bei den Annotationen spricht (siehe Abbildung 11). Zur Überprüfung wurde in einer 11. Runde ein Sample aus den fehlenden oder nicht-übereinstimmenden Annotationen von 1000 gezogen und nochmals von drei Annotator:innen annotiert. Auch hier zeigen sich unterschiedliche Ergebnisse. Dies muss in der Performance berücksichtigt werden. Entsprechend kann auch die hohe Anzahl an nicht-übereinstimmenden Annotationen nicht händisch aufgelöst werden. Um möglichst viele Annotationen aus den Runden mit berücksichtigen zu können, werden unterschiedliche Strategien zur Auflösung der nicht übereinstimmenden Annotationen anwendet. Der Prozess für die Auflösung der Annotationen ist im Appendix (Abbildung 12) aufgezeigt.

Für die Auflösung wird zunächst der evaluierte Datensatz aufgeteilt nach Evaluationen, die ein korrektes Format, inkorrektes Format oder keine Annotation enthalten. Korrektes Format sind alle Annotationen mit 1 oder 0. Korrekte Annotationen werden direkt in den Goldstandard überführt. Alle korrekt annotierten Evaluationen, die mehreren Annotator:innen zugeordnet sind, werden danach aufgeteilt, ob alle Annotator:innen übereinstimmen. Stimmen alle Annotationen überein, wird die Evaluation in den Goldstandard überführt. Bei nicht übereinstimmenden Annotationen wird versucht die Annotation mithilfe von Mehrheitsbeschluss und Expertenvotum aufzulösen. Bei dem Mehrheitsbeschluss wird als Regel festgelegt, dass der Annotationswert korrekt ist, bei dem es eine einfache Mehrheit gibt. Gibt es keine Mehrheit, wird geschaut, ob ein Experte:in die entsprechende Annotation bewertet hat. Falls ja, wird die Annotation der Expert:in übernommen. Von den sieben Annotator:innen wurden drei Expert:innen bestimmt, die in beiden Evaluationen in allen Runden mit annotiert haben und somit die höchste Erfahrung mit den Annotationen haben. Für Annotationen mit inkorrekten Formaten und NA wird ebenfalls die Auflösung mit Mehrheitsbeschluss und Expertenvotum versucht. Auch hier gilt, eine Auflösung mit beiden Strategien ist nur möglich, wenn es für die entsprechenden Evaluationen auch mehrere Annotator:innen gibt. Diese und Annotationen, die durch beide Strategien nicht aufgelöst werden können, werden verworfen.

4 Modell 1: Lokalisierung

Im Folgenden werden, die für die in Stage 1 der Pipeline verwendeten Modelle, angelehnt an die Modelkarten-Methode von Mitchell et al. 2019, dargestellt und erläutert. Die vorgeschlagene Methode umfasst in Grundzügen Informationen über das Modell, die intendierte Nutzung, die Performance Faktoren, die verwendeten Metriken, die Bewertungs- und Trainingsdaten, sowie Ergebnisse.

4.1 Information zum Modell

Das Extraktionsmodell für berufsspezifische Kompetenzen besteht aus dem Skill Extractor und dem Skill Classifier. Der Skill Extractor besteht aus einer Kombination, einer regelbasierten Segmentation und einem Dependency Parser. Der Skill Classifier ist ein LR-Modell mit BERT als Feature Extraction Methode. Die LR verwendet einen lbsfg Optimizer und eine L2 Penalty Regularisierung.

4.2 Intendierte Nutzung und Performance Faktoren

Der Skill Classifier des Extraktionsmodells ist auf der Jobbeschreibung trainiert. Im Vergleich ergibt sich für die Volltexte somit ein Nachteil in der Performance. Wird die Extraktion auf Volltexte angewandt, ist es wichtig, mithilfe des Skill Matchers aus Stage 2 alle klassifizierten Kompetenzsätze auszuschließen, für die es kein Ergebnis mit dem Matcher gibt. Andernfalls ist die angegebene Precision deutlich geringer. Es ist zu beachten, dass durch das Filtern der Recall für die Volltexte abnimmt.

4.3 Metriken

Der Skill Extractor wird nicht separat evaluiert. Vielmehr ergibt sich durch die Evaluation des Skill Classifiers ein Gesamtbild der Performance für Stage 1. Der Skill Classifier kann mithilfe des Validierungsdatensatzes, wie in Kapitel 3 beschrieben, evaluiert werden. Für einen Gesamteindruck über die Performance kann die Accuracy herangezogen werden. Die Accuracy berechnet sich für den binären Fall aus der Summe der Anzahl an korrekten Vorhersagen gemessen an der Gesamtanzahl N der Vorhersagen. Gegeben, dass „skill“ die positive Klasse ist, bildet die Anzahl der korrekt als „skill“ klassifizierten Vorhersagen die True Positive (TP) und die korrekt als „nicht skill“ vorhergesagten Werte die True Negatives (TN). Formal lässt sich dies wie folgt ausdrücken (Powers, D. M. 2020):

$$Accuracy = \frac{TP + TN}{N}$$

Obwohl die Accuracy ein guter Indikator für die Performance ist, wird sie kritisiert, die häufig vertretenen Klassen zu bevorzugen. Deshalb wird gängig neben der Accuracy auch die Precision, Recall und F1 berechnet. Die Precision beschreibt den Anteil an korrekt vorhersagten Fällen, gemessen an allen tatsächlich positiven Fällen. Der Recall hingegen gibt eine Auskunft darüber, wie positive Fälle unter den tatsächlich positiven Fällen gefunden wurden. Der F1 Score harmonisiert beide Werte. Formal (Powers, D. M. 2020, Fatourechi et al. 2008):

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{Precision \times Recall}{Precision + Recall} \times 2$$

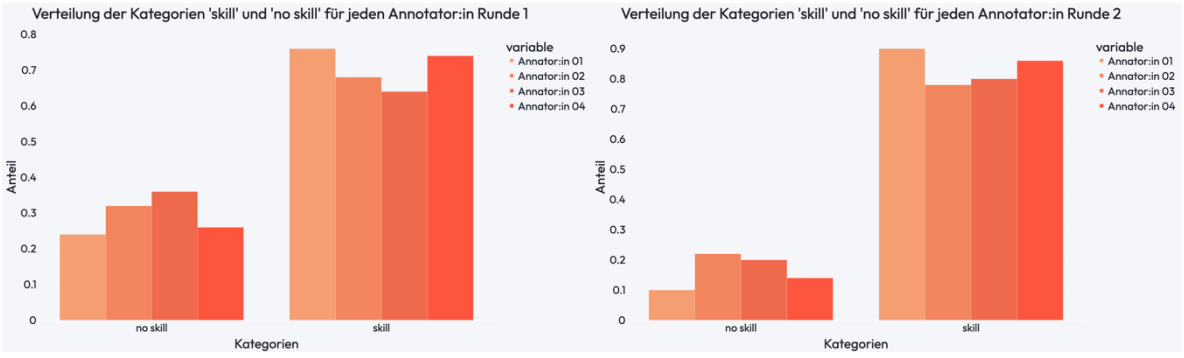
Die als falsch positiv klassifizierten Fälle bilden dabei die Anzahl der False Positive (FP) und entsprechend die als falsch negativ klassifizierten Fälle die False Negative (FN) ab.

4.4 Trainings- und Evaluationsdaten

Für das Training des Skill Classifiers werden die durch Annotationen erstellen Daten (Kapitel 3) verwendet. Insgesamt ergibt sich ein gelabelter Datensatz von 2.100 Examples für den Skill Classifier. Für das Training werden die Daten in Trainings- und Validierungsdaten gesplittet. Der Validierungsdatensatz macht 25% Prozent des Gesamtdatensatzes aus.

Damit der Klassifikationsalgorithmus eine hohe Performance erreicht, ist es wichtig, dass die Daten über alle Klassen hinweg gleichmäßig verteilt sind. Ist das nicht der Fall, könnte das Modell in Richtung der am häufigsten vertretenen Klasse verzerrt sein (Longadge, R., & Dongre, 2013). Betrachtet man in Abbildung 5 die Verteilung der Klassen für beide Runden über alle Annotator:innen hinweg, zeigt sich für beide Runden ein höherer Anteil für die Klasse „skill“. Ein Bias ist somit nicht auszuschließen und muss in der Performanceevaluation berücksichtigt werden. Es werden deshalb neben der Accuracy auch die Precision, der Recall und der F1 Score berechnet.

Abbildung 5: Verteilung Kategorien Skill Classifier Trainingsdatensatz



Wie in Kapitel 3 erläutert, besteht der Datensatz nur aus den Jobbeschreibungen von Stellenausschreibungen. Für die Performance auf den Volltextdaten gibt diese Form der Evaluation somit keinen Aufschluss. Es wird deshalb zu dem Validierungsdatensatz der Testdatensatz (Kapitel 3.2) ebenfalls evaluiert. Dieser enthält Stellenausschreibungen mit Jobausschreibungen und Volltexten. Wie in Kapitel 3.2 beschrieben, werden nur die Kompetenzsätze berücksichtigt. Betrachtet man die Konfusion Matrix für die Vorhersage, wird deutlich, dass die evaluierten Daten nur die True Positives und False Positives abbilden (Abbildung 6). Damit kann nur die Precision für die den Testdatensatz angegeben werden.

Abbildung 6: Konfusion Matrix Evaluationsdatensatz

4.5 Ergebnisse

Im Folgenden werden die Ergebnisse des Validierungsdatensatzes und dem Testdatensatz dargestellt.

Betrachtet man die Ergebnisse des Validierungsdatensatzes, zeigt sich eine hohe Performance des Modells. Mit einer Accuracy von 0.84 ist der Anteil an korrekt klassifizierten Sätzen im Allgemeinen sehr hoch. Zugleich ergeben sich hohe Werte für die Precision mit 0.90, den Recall mit 0.88 und entsprechend dem F1-Score mit 0.89. Das bedeutet, dass die positiven Vorhersagen in den allermeisten Fällen korrekt sind und gleichzeitig das Modell eine hohe Trefferquote unter allen Kompetenzsätzen hat. Ein Bias hinsichtlich der positiven Klasse („Kompetenz“) ist somit unwahrscheinlich. Dies bestätigt sich auch in der in Abbildung 7 dargestellten Konfusionsmatrix, in der sich die Fälle der False Positives und False Negatives nicht erheblich unterscheiden, was ebenfalls gegen das Vorliegen eines Bias spricht.

Abbildung 7: Konfusionsmatrix Skill Classifier Validierungsdatensatz

Tatsächliche Klassen

Vorhergesagte Klasse	Kompetenz	keine Kompetenz
Kompetenz	343	39
keine Kompetenz	45	98

Für den Testdatensatz kann wie in Kapitel 4.4 erläutert, lediglich die Precision berechnet werden. Auf Satzebene beträgt diese 0.67. Aggregiert man den Wert auf Stellenausschreibungsebene ergibt sich eine deutlich geringere Precision von 0.37. Die im Vergleich zum Validierungsdatensatz niedrigen Precision Werte sind auf beiden Ebenen erwartbar, da hier auch Klassifikationen der Volltexte vorhanden sind. Für die Volltexte wird, wie in Kapitel 2.3 Stage 2 näher erläutert, ein zusätzlicher Filter durch den Skill Matcher angewendet, der Falsch Positive Klassifizierungen nochmals reduziert. Obwohl die Performance Scores des Validierungsdatensatzes mit dem Testdatensatz nicht bestätigt, werden können und somit kein exakter Performancescore angegeben werden kann, sollte die zusätzliche Filterung insgesamt zu einer höheren Precision führen. Es ist anzunehmen, dass der tatsächliche Precision-Wert, zwischen den im Testdatensatz beobachteten Werten liegt und als hinreichend betrachtet werden kann.

5 Modell 2: Normalisierung

Im Folgenden werden die für in Stage 2 der Pipeline verwendeten Modelle angelehnt an die Modelkarten-Methode von Mitchell et al. 2019 dargestellt und erläutert. Die vorgeschlagene Methode umfasst in Grundzügen Informationen über das Modell, die intendierte Nutzung, die Performance Faktoren, die verwendeten Metriken, die Bewertungs- und Trainingsdaten, sowie Ergebnisse.

5.1 Informationen zum Modell

Die Klassifikation der Kompetenzsätze nach der VerBIS Taxonomie basiert auf der Skill Matcher und der Skill BERT Komponente. Der Skill Matcher basiert auf einem PhraseMatcher und normalisiert Kompetenzsätze, die wortwörtlich Kompetenzen aus der VerBIS Taxonomie enthalten. Das Skill BERT baut auf einem vortrainierten Modell und einem pooling Layer auf. Es wird unter Verwendung einer Siamese Network Architektur so gefinetuned, dass es ähnliche Kompetenzen aus der VerBIS Taxonomie erkennen kann. Mithilfe von BERT kann dann über die Kosinus-Ähnlichkeit jeder Kompetenzsatz auf die ähnlichste Kompetenz aus der VerBIS Taxonomie aufgelöst werden.

5.2 Intendierte Nutzung und Performance Faktoren

Die Pipeline wird auf einer reduzierten Version der VerBIS Taxonomie getestet und evaluiert. Die Reduzierung ergibt sich durch die hier festgelegene Definition von berufsspezifischen Kompetenzen. Wird die festgelegte Definition von berufsspezifischen Kompetenzen angenommen, sollte das Modell nur auf der reduzierten Variante der Taxonomie verwendet werden. Dennoch kann das Modell, da es auf der gesamten Taxonomie trainiert ist, auch auf der gesamten Taxonomie angewendet werden. Die Performance kann allerdings dadurch beeinflusst sein.

5.3 Metriken

Für die Messung der Performance im Fine Tuning Prozess des BERT Modells wird für das vortrainierte Modell und für das Modell mit Fine Tuning die Kosinus-Ähnlichkeit für die ähnlichen und nicht ähnlichen Paaren berechnet. Anschließend wird der vorgegebene Score von der errechnete Similarity abgezogen. Beide Scores sollten entsprechend niedrig sein. Die Differenzen für die einzelnen Paare werden dann separat für die ähnlichen und nicht ähnlichen Paare gemittelt. Die Mittelwerte werden mit den Ergebnissen des vortrainierten Modells verglichen.

Die Ergebnisse aus dem Skill Matcher und dem Skill BERT werden gemeinsam evaluiert, da es für die Performance keine Rolle spielt, welche Komponente, welche Resultate herausgibt. Die Ergebnisse beider Komponenten kann als Multiklassifikationsproblem betrachtet werden, unter der Voraussetzung, dass die Ergebnisse pro Kompetenzsatz und pro Resultate, wie bei dem Testdatensatz in Kapitel 3.2, beschrieben, aufgeteilt sind. Jeder Kompetenzsatz wird dann einer der Kompetenzklassen der VerBIS Taxonomie zugeordnet. Bei einem Multiklassifikationsprobleme berechnet sich die Accuracy wie folgt:

$$OA = \frac{\sum_{i=1}^{m} tp_i}{N}$$

wobei m die aktuelle Klasse definiert, tp die True Positive Werte für die aktuelle Klasse und N die Gesamtanzahl aller Klassifikationen.

Die Precision, der Recall und der F1 Wert können entweder als Makro oder Mikro Werte berechnet werden. Da der Micro Score die Werte über alle Klassen hinweg aggregiert ist er gleich zur Accuracy und somit besteht keine Notwendigkeit diesen separat zu berechnen. Die Berechnung des Recalls erfolgt wie in Kapitel 4.3 auf Basis von den False Negatives. Da für den Testdatensatz lediglich evaluiert wird, ob die Resultate korrekt sind, nicht aber korrigiert werden, können die False Negatives nicht ermittelt werden. Folglich kann keine Berechnungen von Recall und F1 Score erfolgen. Hingegen kann die Anzahl an False Negatives bestimmt werden und somit auch die Makro Precision. Diese wird hier neben der Accuracy ebenfalls angegeben. Das ist wichtig, weil die Klassen des Testdatensatzes ungleich verteilt sind. Angenommen M ist die Anzahl der allen Klassen, tp die True Positives und fn die False Negatives für die aktuelle Klasse m, dann berechnet sich die Precision wie folgt:

$$precision_{macro} = \frac{1}{M} \sum_{i=1}^{m} \frac{tp_i}{tp_i + fn_i}$$

5.4 Trainingsdaten und Evaluationsdaten

Die Suchwörter, die Labels für jede berufsspezifische Kompetenz, sowie die Label der Gruppennamen der VerBIS Taxonomie bilden die Grundlage für das Fine Tuning. Während für das Tuning die gesamte Taxonomie verwendet wird, wird für die Evaluation die reduzierte Version, wie in Kapitel 2.1 beschrieben, verwendet. Erstere rechtfertigt sich insofern, dass die Verwendung der gesamten Taxonomien für Training es ermöglicht das BERT-Modell auf die gesamte Taxonomie anwenden zu können, wenn eine andere Definition, als die hier festgelegte für berufsspezifischer Kompetenzen angenommen werden möchte.

Für das Erlernen von Ähnlichkeiten benötigt das Siamese Network sowohl Paare mit hohem Ähnlichkeitsscore als auch Paare mit niedrigem Score. Für die ähnlichen Paare werden jeweils eine berufsspezifische Kompetenz und ein dazugehöriges Suchwort kombiniert und mit einem Score von 0.8 als Input verwendet. Insgesamt ergeben sich so 40.535 Paare. Für die unähnlichen Paare werden berufsspezifische Kompetenzen mit ihnen nicht zugehörigen Gruppenlabel kombiniert. Für die unähnlichen Paare wird ein Sample aus allen Kombinationen verwendet. Aufgrund der hohen Anzahl an möglichen Kombinationen, wird hier ein Sample von 1.000.000 Paaren verwendet. Der Score wird auf 0.2 gesetzt. Tabelle 3 zeigt ein Ausschnitt aus dem so erstellen Datensatz:

Tabelle 3: Trainingsdaten Fine Tuning Skill BERT

Kompetenz	Suchwort	Label
Blumenversand	Blumenhandel	0.8
Versiegeln (Parkett)	Parkettversiegeln	0.8
CNC-, NC-Programm Gildemeister	Kommunikationspsychologe	0.2
Produktionsleitung (Film, TV, Bühne)	Brandbekämpfung	0.2

Die Daten werden in Trainings- und Validierungsdaten gesplittet mit einer Validierungsgröße von 20% der Daten. Die Validierungsdaten sollen zur Messung der Performance des Fine Tunings verwendet werden.

Wie in Kapitel 3.2 beschrieben, wird der Testdatensatz für die Berechnung der Metriken in Stage 2 verwendet. Basierend auf der vorliegenden Datenbasis und dem beschriebenen Auflösungsprozess in Kapitel 3.2 umfasst der Datensatz der zweiten Evaluation 7.441 Postings von ursprünglich 9.844. Dies führt zu einer Reduktion der Abdeckung der VerBIS-Kompetenzen und -Gruppen. Insgesamt sind 3.373 VerBIS-Kompetenzen aus 259 Gruppen vertreten, was als ausreichende Abdeckung betrachtet wird. Jedoch ist es wichtig zu beachten, dass der Testdatensatz möglicherweise in zwei Punkten verzerrt ist. Erstens, da nur Kompetenzsätze berücksichtigt werden, die in der Evaluation als "1" klassifiziert wurden, also nur die True Positives, könnte es sein, dass Sätze, die tatsächlich eine Kompetenz enthalten, tendenziell korrekt durch den Skill Matcher und Skill BERT klassifiziert werden, während False Positive-Sätze tendenziell falsche Klassifikationen aufweisen. Zweitens besteht das Risiko, dass Sätze, bei denen die Annotatoren:innen sich uneinig sind und die verworfen werden durch den Auflösungsprozess, mehrheitlich Kompetenzsätze sind, die tatsächlich falsch klassifiziert sind. Beide Verzerrungen müssen bei der Interpretation der Performance Scores berücksichtigt werden.

5.5 Ergebnisse

Im Folgenden werden die Ergebnisse für das Fine Tuning des Skill BERT Modells beschrieben und die Ergebnisse der Kompetenzzuordnungen von Skill BERT und Skill Matcher.

Fine Tuning. Für das Basismodell (ohne Fine Tuning) ergibt sich ein Wert von 0.098 für ähnliche Paare und für die Evaluation der unähnlichen Paare ein Wert von 0.55. Während das Modell mit Fine Tuning für ähnliche Paare sich mit einem Wert von 0.11 leicht verschlechtert hat, kann das Modell durch das Fine Tuning mit einem durchschnittlichen Wert von 0.02 für nicht ähnliche Paare erheblich besser unähnliche Kompetenzen differenzieren. Es lässt sich entsprechend Schlussfolgern, dass das Fine Tuning die Performance des Modells verbessert hat.

Skill BERT und Skill Matcher. Für den Testdatensatz ergibt sich eine Accuracy von 0.73 und eine Makro-Precision von 0.63. Die hohe Accuracy deutet darauf hin, dass in den meisten Fällen Kompetenzsätze korrekt klassifiziert werden. Der niedrigere Makro-Precision-Wert im Vergleich zur Accuracy deutet darauf hin, dass es für manche Klassen eine höhere Anzahl an False Positives gibt. Betrachtet man die Verteilung der Klassen im Testdatensatz, so haben etwa 75% der Daten weniger als zehn Beispiele, und knapp 29% der Labels haben nur ein Beispiel. Berechnet man die Precision ohne Berücksichtigung der Labels mit nur einem Beispiel, ergibt sich eine Precision von 0.66. Wenn man nur Labels berücksichtigt, die mehr als zehn Beispiele haben, liegt die Precision bei 0.70. Die Precision-Werte sind zwar höher, aber nicht erheblich angestiegen. Daraus kann gefolgert werden, dass die Mehrheitsklassen im Durchschnitt eine leicht bessere Precision haben, die Minderheitsklassen jedoch auch gut vorhergesagt werden. In Anbetracht dessen und der Vielzahl an Labels in der VerBIS-Taxonomie ist die Performance als hoch und ausreichend zu bewerten. Es ist jedoch zu beachten, dass der Testdatensatz nur Sätze enthält, die tatsächlich eine Kompetenz enthalten. Die Werte geben daher keinen Aufschluss über die Performance bei Sätzen, die in der ersten Stufe der Extraktionspipeline fälschlicherweise als Kompetenzsätze eingestuft und in der zweiten Stufe klassifiziert werden.

6 Validierung

Die in den Modellkarten dargestellten Ergebnisse zeigen die Performance einzelner Komponenten, liefern jedoch keine umfassenden Performance-Scores für die Gesamtqualität der Extraktion. Zudem sind alle angegeben Metriken nicht auf Stellenausschreibungsebene. Für die bessere Einschätzung der Gesamtqualität, wird im Folgenden neben der Berechnung der Performance-Metriken eine grafische Validierung durchgeführt.

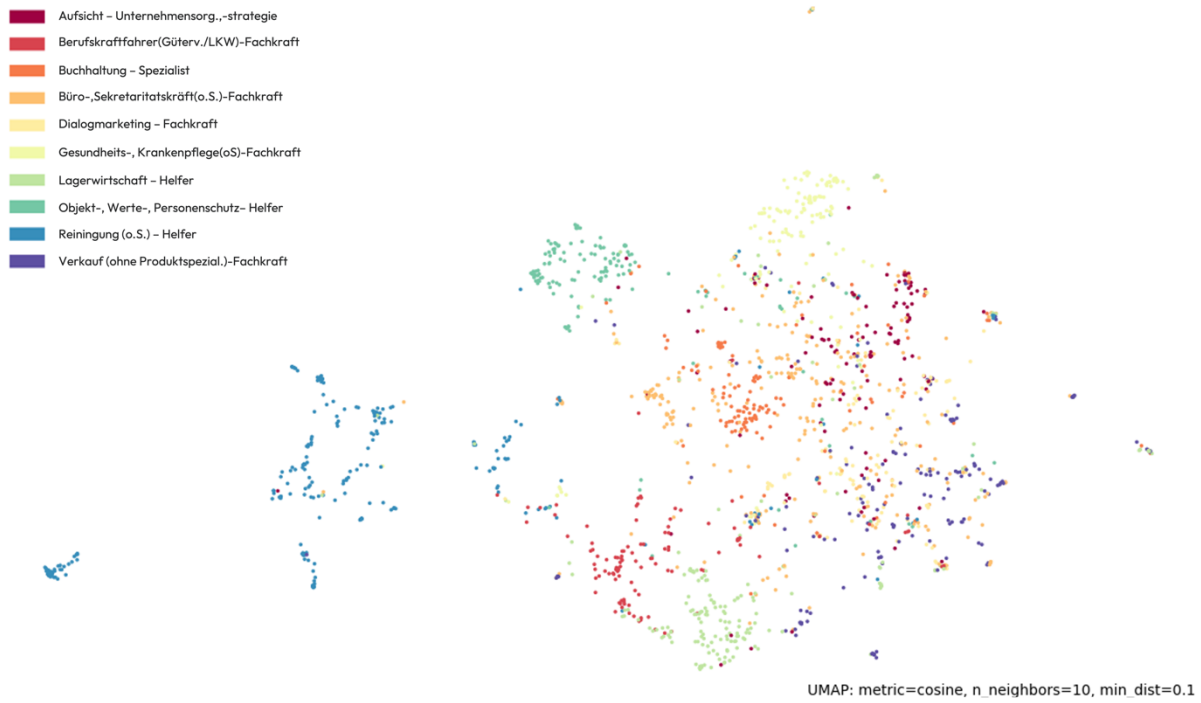
Die Analyse erfolgt auf Basis der Berufe, die in den Stellenausschreibungen gefragt sind. Die Berufe werden durch „Klassifikation der Berufe 2010 – überarbeitet Fassung 2020“ (KldB) Taxonomie abgebildet. Für die Analyse wird angenommen, dass ein Beruf ähnliche Kompetenzen oder ein ähnliches Kompetenzsets aufweiset. Folglich müssen Stellenausschreibungen mit gleichen KldB Beruf ähnlich zueinander sein. Gegeben dieser Annahme sollten sich Cluster mit Ausschreibungen bilden, die demselben Beruf angehören, wenn die Extraktion eine gute Gesamtperformance aufweist.

Für die Durchführung werden zunächst für die Stellenausschreibungen in dem Evaluationsdaten die KldB-Berufe ergänzt.[2] Die Kompetenzen in dem Testdatensatz werden nach Stellenausschreibungen zusammengefasst und in einem String verkettet mit einem Separator. Hat z.B. einen Stellenausschreibung die Kompetenzen „Planung“ und „Werbung“, dann sieht die Kompetenzfolge wie folgt aus: „Planung|Werbung“. Diese Kompetenzfolge repräsentiert dann die Ausschreibung. Zwischen den Ausschreibungen basierend auf den Kompetenzfolgen, wird anschließend die Kosinus-Ähnlichkeit berechnet. Für die Berechnung müssen die Kompetenzfolgen vektorisiert werden. Da die Reihenfolge, wie die Kompetenzen einer Ausschreibung verkettet sind, nicht beachtet werden sollte, werden die Kompetenzfolgen als Bag-of-Words Vektorisierung vektorisiert. Numerisch spiegelt sich das darin wider, dass die Kompetenzen in Häufigkeiten übersetzt werden und somit der Kontext ungeachtet bleibt. Implementiert wird die Vektorisierung mit dem CountVectorizer von sklearn (Pedregosa, F 2011). Damit die Kompetenzen als Tokens behandelt werden, wird ein Token Pattern an den CountVectorizer übergeben, sodass jede Kompetenz als Token repräsentiert wird. Anschließend werden die Kompetenzen in einem 2-dimensionalen Raum abgebildet und farblich nach KldB-Berufen markiert. Aus Darstellungsgründen werden nur die Top 10 Berufe in dem Testdatensatz visualisiert. Für die Visualisierung im zweidimensionalen Raum wird die „Uniform Manifold Approximation and Projection“ (UMAP) Methode verwendet. UMAP dient zur Dimensionsreduktion (McInnes & Healy, 2018). Bei der Anwendung von UMAP müssen verschiedene Parameter festgelegt werden. Der Parameter „n_neighbors“ definiert die Anzahl benachbarter Punkte: Größere Werte führen zu einer globaleren Sichtweise, während kleinere Werte lokale Strukturen betonen. Der Parameter „min_dist“ legt fest, wie dicht die Punkte angeordnet werden: Höhere Werte führen zu einer größeren Streuung der Punkte, während kleinere Werte die Punkte enger zusammenführen.

Abbildung 8 zeigt die Ergebnisse für den Testdatensatz. Die Parameterwerte „min_dist“ von 0.1 und „n_neighbors“ gleich 10 sind durch Experimentieren mit verschiedenen Parametereinstellungen ermittelt. Die Abbildung zeigt für einige Berufe klare Cluster. Deutlich wird das beispielsweise bei „Objekt-,Werte-,Personenschutz-Helfer“ und „Gesundheits-, Krankenpflege(o.S) – Fachkraft“. Auch für „Reinigung (o.S.) – Helfer“ zeigen sich zwei Cluster. Im Gegensatz dazu ist der Beruf „Büro-, Sekretariatskräfte (o.S.) – Fachkraft“ sehr breit gestreut und zeigt kein erkennbares Cluster. Generell gibt es auch, trotz einiger Cluster , Stellenausschreibungen, die keinem Cluster zugeordnet werden können. Bei Berufen wie „Büro-, Sekretariatskräfte (o.S.) – Fachkraft“ oder „Aufsicht-Unternehmensorg.-strategie“ könnte dies jedoch darauf zurückzuführen sein, dass die geforderten Kompetenzen in diesen Berufen sehr unterschiedlich sind. Im Allgemeinen bestätigt die graphische Darstellung aber das Bild, dass die Extraktionspipeline eine gute Differenzierung hat und sich Cluster für die Berufe bilden.

[2] Der Testdatensatz enthält für jede Stellenausschreibung die zusätzliche Information „Jobtitel“. Für die Klassifikation des Jobtitels nach KldB wird das Modell, dass von Baskaran R., Müller J.(2023) beschrieben wurde, verwendet.

Abbildung 8: Verteilung von Kompetenzfolgen nach Berufen mit UMAP



7 Zusammenfassung und Anwendung

Die Ergebnisse aus den Modellkarten und die Graphische Analyse zeigen generell eine gute Performance für die die Extraktion. Zugleich, zeigt aber auch eindeutig die Accuracy von 0.37 auf Stellenausschreibungsebene für den Skill Classifier und die Ausreißer in den Abbildung 8, dass innerhalb von Stellenausschreibungen nicht passende berufsspezifische Kompetenzen vorkommen werden und gegebenenfalls die Qualität von Analysen beeinträchtigen kann. Für die Verbesserung der Qualität der Extraktionsdaten besteht deshalb die Möglichkeit eine weitere Filterung der Daten vorzunehmen und so die Datenqualität zu verbessern.

Die VerBIS Kompetenzen sind in einem Mapping von Berufnet (Berufenet 2024) mit den KldB-Berufen vernetzt. Jedem KldB Beruf sind eine Reihe an möglichen VerBIS Kompetenzen zugeordnet, die für den Beruf passend sind. Abbildung 9 zeigt ein Ausschnitt aus dem Mapping. Die Zuordnung ist von Expert:innen durchgeführt und mit empirischen und theoretischen Methoden unterlegt (Bundesagentur für Arbeit 2021). Das Mapping kann deshalb genutzt werden, um VerBIS Kompetenzen für eine Stellenausschreibung zu entfernen, die nach dem Mapping für einen Beruf nicht vorkommen können. Ist eine Stellenausschreibung beispielsweise der KldB „72302“ zugeordnet, dann ist „Gewerbesteuer“ eine berufsspezifische Kompetenz, die für die Stellenausschreibung „erlaubt“ ist. Kompetenzen, die in der „skill_mappings“ Liste für die KldB „72302“ hingegen nicht vorkommen, aber für die Stellenausschreibung extrahiert sind, würden dann durch die Filterung entfernt werden. Auf diesem Wege kann sichergestellt werden, dass Kompetenzen, die nicht zu dem Beruf passen, nicht in der Extraktion vorkommen, was die Qualität der Extraktion verbessert.

Abbildung 9: VerBIS-KldB Mapping


```
{
  "kldb": "72302",
  "skill_mappings": [
    {
      "label": "Akten, Schriftgut verwalten",
      "type": "Kernkompetenz"
    },
    {
      "label": "Gewerbesteuer",
      "type": "Kernkompetenz"
    },
    {
      "label": "Steuererklärungen anfertigen",
      "type": "Kernkompetenz"
    }, ...
 },
 {
  "kldb": "72213",
  "skill_mappings": [
    {
      "label": "Monatsabschluss",
      "type": "Kernkompetenz"
    },
    {
      "label": "Baulohnbuchhaltung, -abrechnung",
      "type": "Kernkompetenz"
    }, ...
  }, ...
```


Das Mapping kann mithilfe von drei Faktoren erfolgen und entsprechend kann die Filterung restriktiver oder wenig restriktiver sein. Im Folgenden werden die einzelnen Faktoren näher erläutert.

KldB-Level. Die KldB Taxonomie ist hierarchisch aufgebaut mit fünf Leveln. Individuelle Berufe werden auf den einzelnen Levels in Gruppen zusammengefasst, wobei mit aufsteigendem Level die Gruppen spezifischer werden. Aufgrund der Kodierung der KldB kann mit dem Level 5 KldB-Code jedes andere KldB-Level für einen Beruf bestimmt werden.[3] Entsprechend kann das in Abbildung 9 dargestellte VerBIS-KldB Mapping angepasst werden. Fasst man beispielsweise das Mapping auf KldB Level 1 zusammen, würden für alle Stellenausschreibungen deren KldB-Berufe mit „7" beginnt, sowohl die berufsspezifischen Kompetenzen von der KldB „72302" als auch von der KldB „72212" zulässig sein.

Kompetenztyp. Das Mapping unterscheidet zwischen Kernkompetenzen und weiteren Kompetenzen (Abbildung 9, „type"). Kernkompetenzen sind berufsspezifische Kompetenzen, die nur für einen Beruf gelten, während die weiteren Kompetenzen auch in anderen Berufen vorkommen können. Werden nur die Kernkompetenzen berücksichtigt, dann wird die Anzahl an möglichen Kompetenzen für einen Beruf eingeschränkt. Das kann bei zielgerichteteren Analysen von Kompetenzen sinnvoll sein.

Nicht validierbare Kompetenzen. Nicht alle Kompetenzen, die in der VerBIS Taxonomie vorkommen, sind auch in dem Mapping vorhanden. Entsprechen können entweder die nicht prüfbaren Kompetenzen entfernt werden oder sie werden beibehalten. Entfernt man die nicht prüfbaren Kompetenzen nicht, dann können in der Extraktion weiterhin Ausreißer vorhanden sein. Bei Entfernung könnten allerdings auch Kompetenzen entfallen, die eigentlich korrekt zugeordnet sind.

Alle drei Faktoren können miteinander unterschiedlich kombiniert werden. Es muss für den Einzelfall eine Abwägung zwischen der Anzahl an entfallenen Extraktionen und Notwendigkeit der Restriktionen getroffen werden. Es wird vorgeschlagen, die Analyse zunächst ohne Filterung durchzuführen und anschließend basierend auf den Ergebnissen zu entscheiden, ob und in welchem Umfang eine Filterung erforderlich ist.

Mithilfe der Filterung werden nochmals Performance Scores für den Testdatensatz mit der Filterung berechnet. Es wird erwartet, dass durch die Filterung die Performance Scores sich verbessern. Da die Annotation des Testdatensatzes unabhängig von der Filterung geschehen ist, kann so die Filterung validiert werden und zu gleich auch eine hohe Annotationsqualität der Testdaten bestätigt werden. Für die Filterung werden beide Kompetenztypen miteinbezogen, nicht validierbare Kompetenzen werden entfernt. Die Ergebnisse werden für das KldB Level 1,3 und 5 angegeben.

[3] Eine ausführliche Beschreibung der KldB-Taxonomie findet sich in Baskaran R., Müller J.(2023).

Tabelle 4: Filterung der Kompetenzen

Level	Anzahl an Daten	Anzahl an Klassen	Accuracy	Precision
1	26.705	1878	0.79	0.73
3	16.997	1440	0.84	0.81
5	10.419	1119	0.88	0.85
Baseline (ohne Filterung)	46.205	3373	0.73	0.63

Die Accuracy und Precision steigen im Vergleich zur Baseline deutlich kann. Das wird vor allem in der Precision deutlich. Auch ist zu sehen, dass die restriktivste Filterung auf Level 3 zu einer deutlichen Steigerung der Precision und der Accuracy führt im Vergleich zur Baseline, aber auch im Vergleich zu den Filterungen auf den anderen Level. Zugleich sind auf Level 5 nur knapp 23% der vorhanden, während auf Level 1 58% der Daten noch vorhanden sind.

Appendix

A. Testdaten Extraktionspipeline Annotation II-1

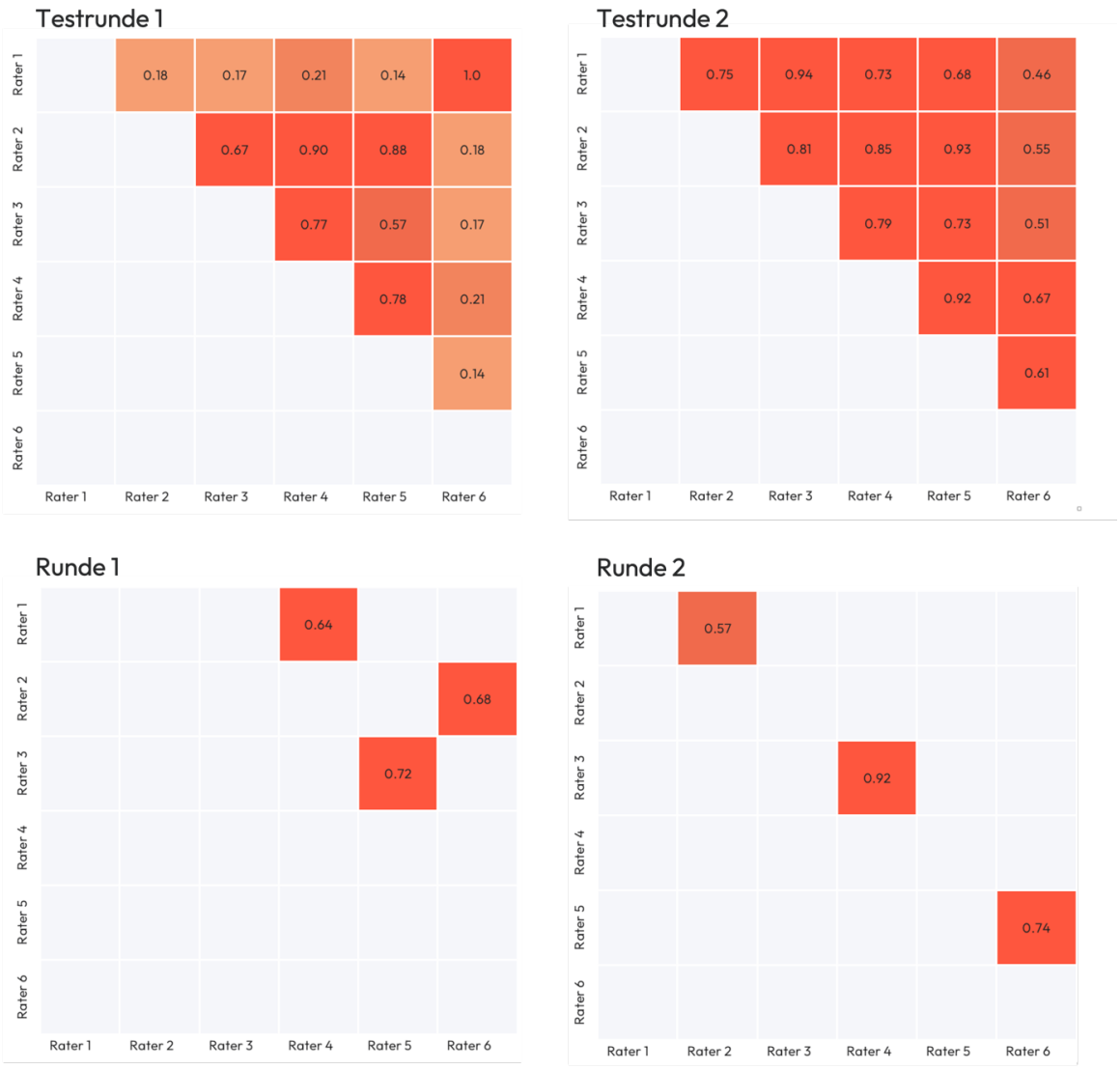
A1. Überblick Anzahl an Annotationen pro Runde

Tabelle 5: Anzahl an Annotationen pro Runde Annotation II-1 (Appendix)

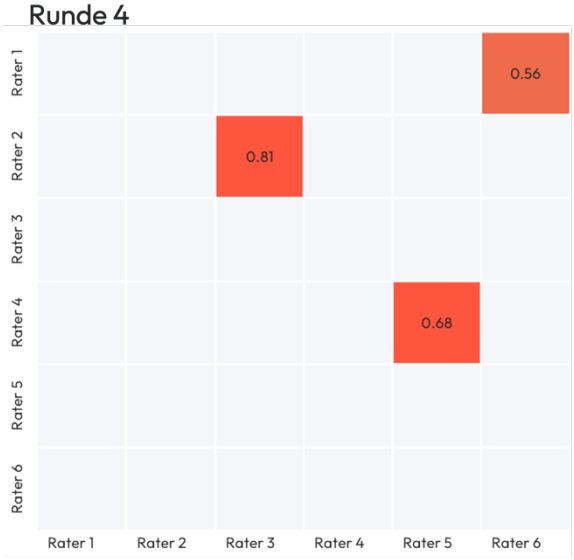
Runde	Anzahl Daten	Anzahl Überschneidung
Testrunde 1	50	50 (zwischen allen Annotator:innen)
Testrunde 2	50	50 (zwischen allen Annotator:innen)
Runde 1	1000	100 (zwischen jeweils 2 Annotator:innen)
Runde 2	1500	100 (zwischen jeweils 2 Annotator:innen)
Runde 3	1500	100 (zwischen jeweils 2 Annotator:innen)
Runde 4	1500	100 (zwischen jeweils 2 Annotator:innen)
Runde 5	1500	100 (zwischen jeweils 2 Annotator:innen)
Runde 6	1500	100 (zwischen jeweils 2 Annotator:innen)
Runde 7	1500	100 (zwischen jeweils 2 Annotator:innen)
Runde 8	1500	100 (zwischen jeweils 2 Annotator:innen)
Runde 9	1500	100 (zwischen jeweils 2 Annotator:innen)
Runde 10	886	100 (zwischen jeweils 2 Annotator:innen)
Runde 11	1500	100 (zwischen jeweils 2 Annotator:innen)

A2. Cohens Kappa für pro Annotationsrunde

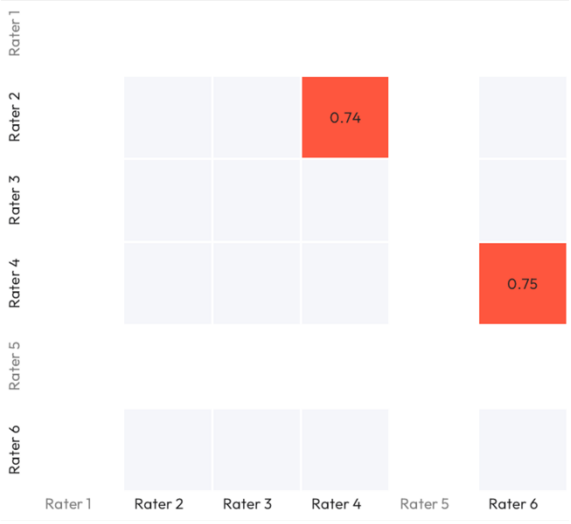
Abbildung 10: Cohens Kappa pro Runde Annotation II-1 (Appendix)



Runde 4

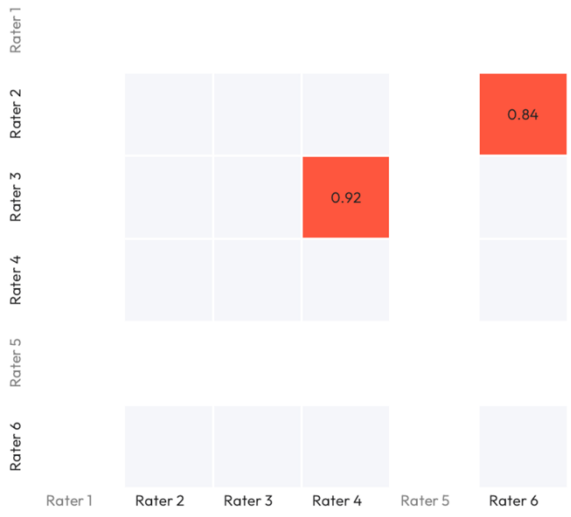


Runde 5



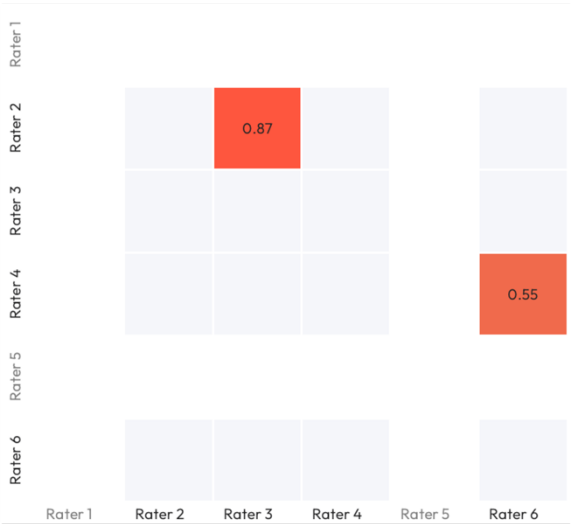
Runde 6

Runde 7



Runde 8

Runde 9



Runde 10

Runde 11

B. Ergebnisse Annotation II-2: Testdaten Extraktionspipeline

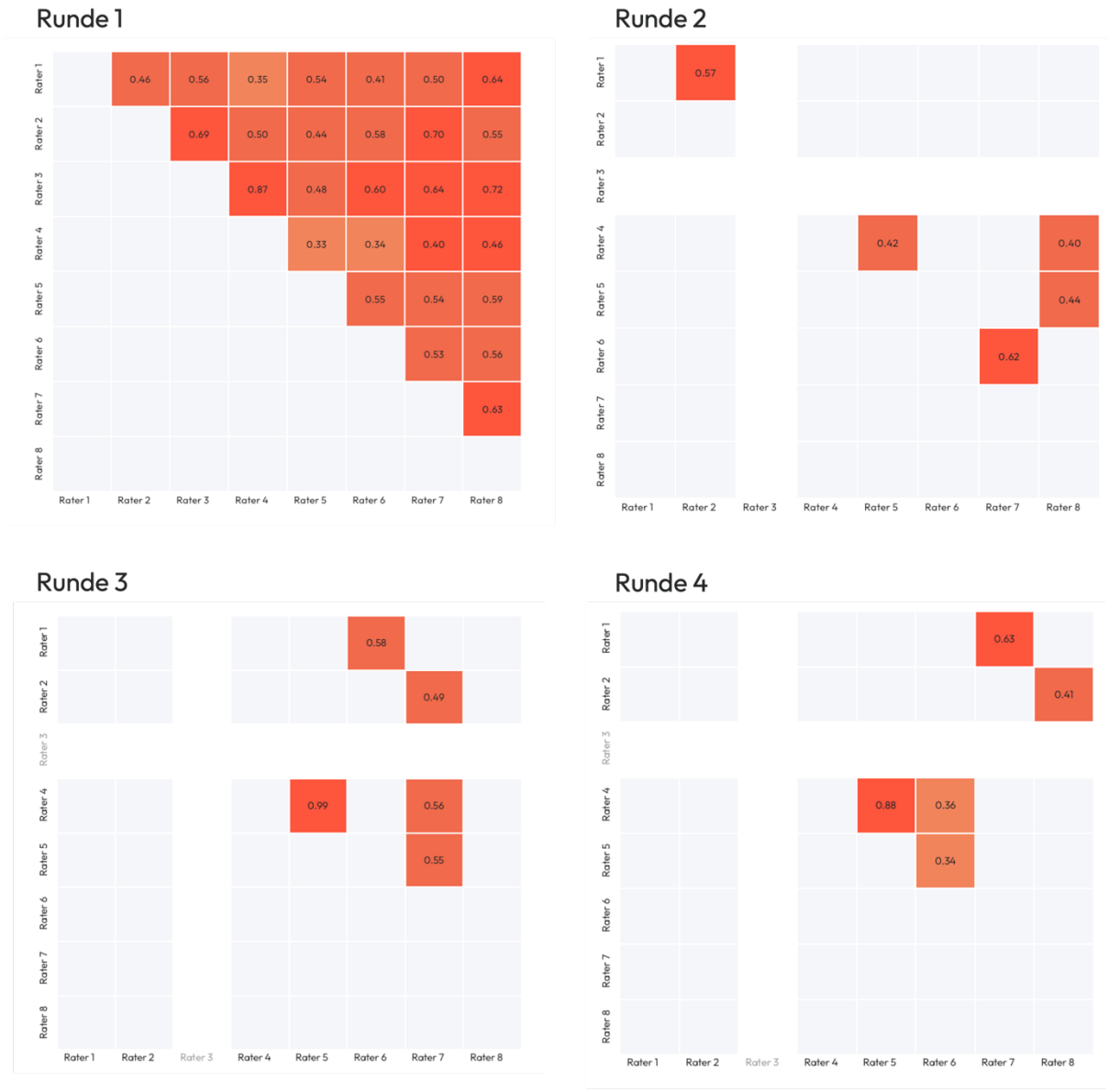
B1. Überblick Anzahl an Annotationen pro Runde

Tabelle 6: Anzahl an Annotationen pro Runde Annotation II-2 (Appendix)

Runde	Anzahl Daten	Anzahl Überschneidung
Runde 1	200	200 (zwischen allen Annotator:innen)
Runde 2	1200	200 (zwischen 2, bzw. 3 Annotator:innen)
Runde 3	1200	200 (zwischen 2, bzw. 3 Annotator:innen)
Runde 4	1200	200 (zwischen 2, bzw. 3 Annotator:innen)
Runde 5	1200	200 (zwischen 2, bzw. 3 Annotator:innen)
Runde 6	1200	200 (zwischen jeweils 2 Annotator:innen)
Runde 7	1200	200 (zwischen jeweils 2 Annotator:innen)
Runde 8	1200	200 (zwischen jeweils 2 Annotator:innen)
Runde 9	1200	200 (zwischen jeweils 2 Annotator:innen)
Runde 10	1200	200 (zwischen jeweils 2 Annotator:innen)
Runde 11	1000	1000 (zwischen allen Annotator:innen)

B2. Cohens Kappa für pro Annotationsrunde

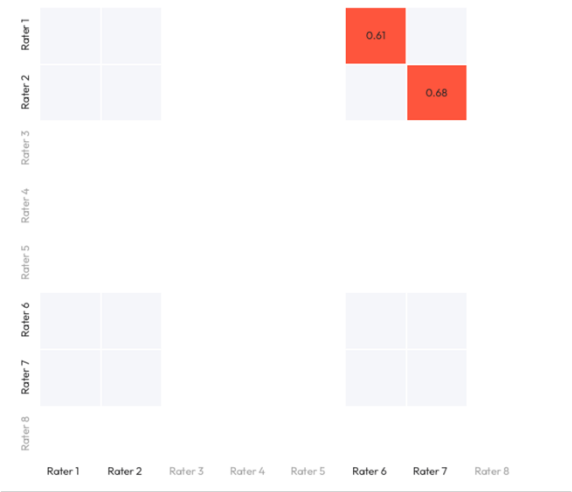
Abbildung 11: Cohens Kappa pro Runde Annotation II-2 (Appendix)



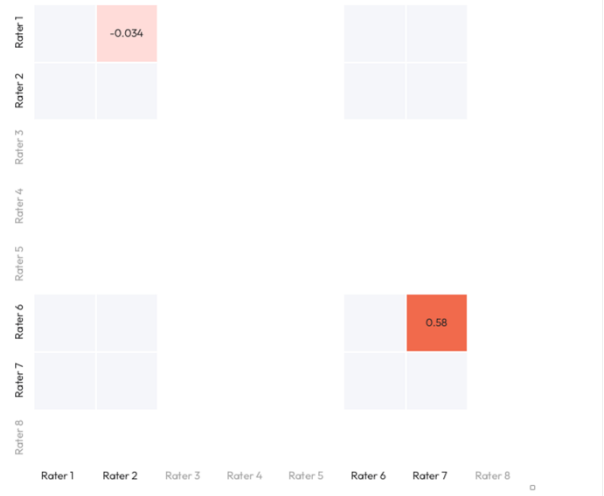
Runde 5

Runde 6

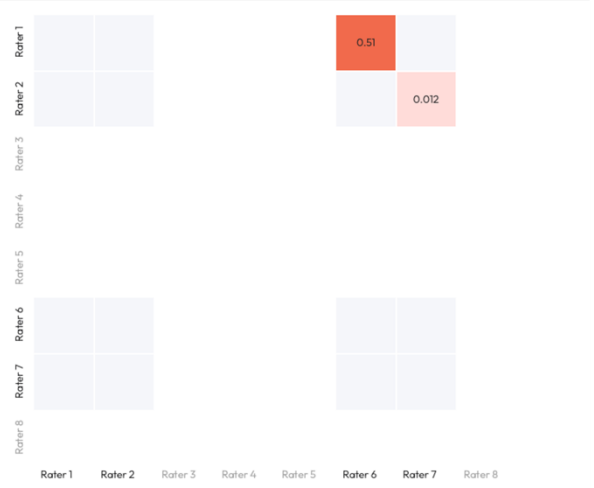
Runde 7



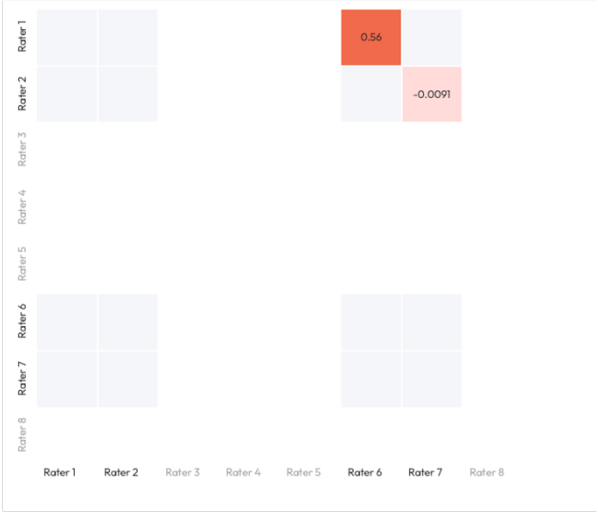
Runde 8



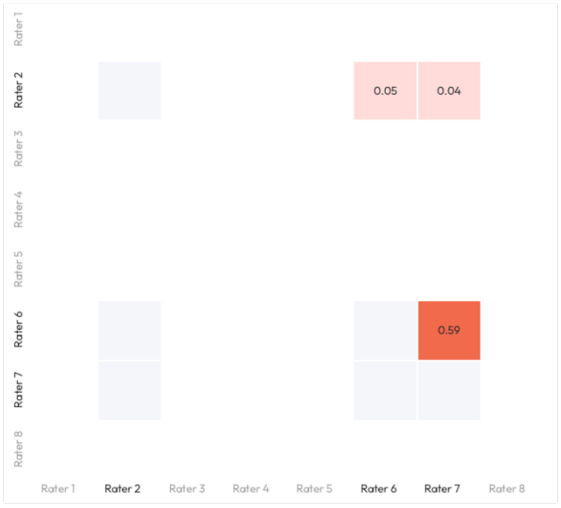
Runde 9



Runde 10

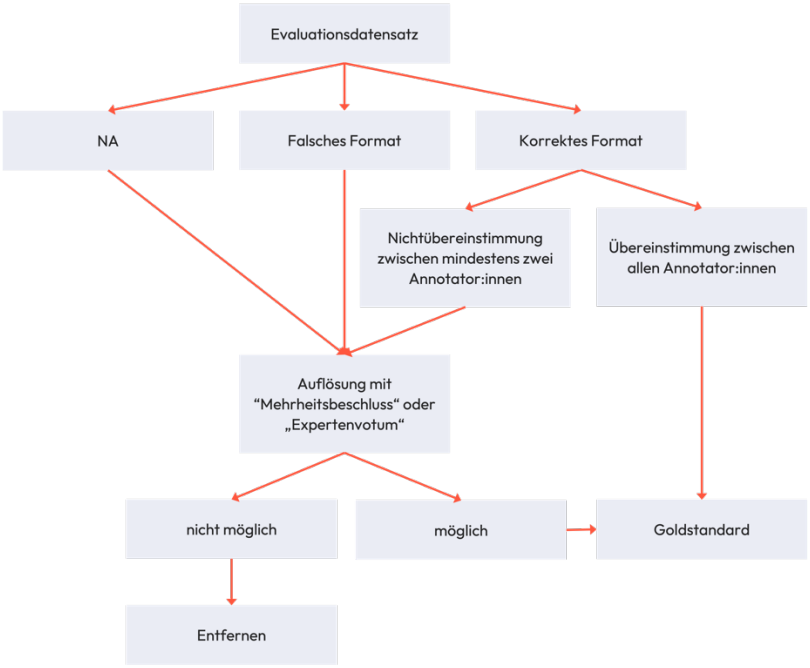


Runde 11



C. Auflösungsprozess Annotation :Normalisierung

Abbildung 12: Prozess zur Auflösung von nicht übereinstimmenden Annotationen Evaluation Normalisierung (Appendix)



Literatur

Baskaran R., Müller J.(2023). Classification of German job titles in online job postings using the KldB 2010 taxonomy <https://www.and-effect.com/publications/2022-11-21_technical_report_kldb.pdf> (letzter Zugriff am 24.06.2024).

Bayerische Staatsbibliothek (2019). Dbmdz German BERT models. < https://huggingface.co/dbmdz/bert-base-german-cased> (letzter Zugriff am 01.07.2024).

Bundesagentur für Arbeit (2021). Klassifikation der Berufe 2010 - überarbeite Fassung 2020 Band 1: Systematischer und alphabetischer Teil mit Erläuterungen.

Bundesagentur für Arbeit (2024a). Download Portal Arbeitsagentur <https://download-portal.arbeitsagentur.de> (letzter Zugriff am 04.07.2024).

Bundesagentur für Arbeit (2024b). Berufenet <https://web.arbeitsagentur.de/berufenet/> (letzter Zugriff am 05.07.2024).

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales., 20(1). In Educational and Psychological Measurement, (pp.37–46).

Fatourechi, M., Ward, R. K., Mason, S. G., Huggins, J., Schlögl, A., & Birch, G. E. (2008). Comparison of evaluation metrics in classification applications with imbalanced datasets. In 2008 seventh international conference on machine learning and applications (pp.777-782).

Fei, Y., Rong, G., Wang, B., and Wang, W. (2014). Parallel L-BFGS-B algorithm on GPU. In Computers & graphics, (pp.1-9),

Honnibal, M., & Johnson, M. (2015). An improved non-monotonic transition system for dependency parsing. In Proceedings of the 2015 conference on empirical methods in natural language processing, (pp. 1373-1378).

Jurafsky, D. and Martin, J. H. (2021). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition <https://web.stanford.edu/~jurafsky/slp3/old_dec20/ed3book_dec302020.pdf> (letzter Zugriff am 01.07.2024).

Longadge, R., & Dongre, S. (2013). Class imbalance problem in data mining review.

McInnes, L, Healy, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.

Palmer, D. D. (2000). Tokenisation and sentence segmentation. In Handbook of natural language processing, (pp.11-35).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research.

Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

Reimers, N. & Matthes, B. (2019). Sentence-Bert - Sentence emebddings using siamese bert networks. arXiv

spaCy. (o.D.). Industrial-strength Natural Language Processing in Python <https://spacy.io/> (letzter Zugriff am 1.07.2024).

spaCy. (o.D.). Dependency Parser. <https://spacy.io/api/dependencyparser> (letzter Zugriff am 01.07.2024)

Textkernel. (2022). Textkernel. <https://www.textkernel.com/de/> (letzter Zugriff am 01.09.2023).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems, 30.

Yuan, Y., Jiang, Y., & Tu, K. (2019). Bidirectional transition-based dependency parsing. In Proceedings of the AAAI Conference on Artificial Intelligence (pp. 7434-7441).