Universität zu Köln

Philosophische Fakultät

Institut für Digital Humanities

Themensteller: Dr. Jürgen Hermes

Evaluation von Extraktionsmustern im Kontext von Text Mining auf Stellenausschreibungen

vorgelegt von: Christine Schäfer

Matrikelnummer: 7340118

E-Mail: cschae27@smail.uni-koeln.de

Datum der Abgabe: 12.03.2021

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

Der Computer als ein alltägliches Mittel zur Verarbeitung von Daten ist aus unserer Gegenwart nicht mehr wegzudenken. Durch die fortschreitende Digitalisierung ist seine Notwendigkeit eindeutig, denn mit ihm ist es möglich, die schiere Masse an Daten, die heutzutage produziert wird, zu verarbeiten und in eine für Menschen lesbare und verständliche Form zu bringen. Es ist unumstritten, dass der Mensch es mit seinen verfügbaren Ressourcen nicht schaffen kann, die großen Datenmengen, im informatischen Bereich auch *Big Data* genannt, die heutzutage täglich produziert werden, sei es im öffentlichen, beruflichen oder privaten Bereich, in ihrem Umfang zu verarbeiten. Fast jeder Bereich ist gleichermaßen davon betroffen. Mittels unterschiedlicher Methoden wird in den interdisziplinären Wissenschaften der Informatik, z.B. der Wirtschaftsinformatik, Medieninformatik oder Computerlinguistik, versucht, fachspezifische Ressourcen anhand themenspezifischer Aufgabenstellungen sinnvoll anzuwenden.

Der Bereich der Computerlinguistik befasst sich hierbei mit der Verarbeitung natürlicher Sprache. Dazu zählt sowohl gesprochene als auch geschriebene Sprache, die in verschiedenen Formaten, z.B. in Form von Texten oder Audiodateien, vorliegen kann. Die Anzahl der Daten, die als natürlichsprachliche Texte vorliegen, ist enorm und ist in sämtlichen Bereichen aufzufinden. In diesem Zusammenhang scheint die Entwicklung von Methoden naheliegend, die sich mit der Gewinnung von Informationen aus natürlichsprachlichen Texten beschäftigen. Denn was für den Menschen in natürlichsprachlichen Texten als Informationen eindeutig erscheint, ist für eine Maschine deutlich schwieriger zu identifizieren. Natürliche Sprache und die Darstellung von Informationen innerhalb dieser unterliegt unterschiedlichsten Repräsentationsformen. Für einen Menschen sind diese leicht zu durchschauen, ist er doch ein alltäglicher Produzent natürlicher Sprache. Für eine Maschine dagegen kann es als eine Herausforderung angesehen werden, diese verschiedenen Formen zu erkennen und dementsprechend korrekt anzuwenden. Anwendungen bzw. Teilbereiche innerhalb der Computerlinguistik spezifizieren linguistische Problemstellungen und beschäftigen sich mit der Entwicklung computergestützter Verfahren zur Lösung dieser.

Maschinelle Sprachverarbeitung (*Natural Language Processing*, NLP) als Teilbereich der Computerlinguistik befasst sich mit der Gewinnung von Informationen aus natürlichsprachlichen Texten. Zugrundeliegende Anwendungen hierfür werden beispielsweise als *Text Mining*, *Information Retrieval*, Informationsextraktion oder Textklassifikation bezeichnet, jedoch ist zu vermerken, dass sich einzelne Benennungen in ihrer Funktionalität

miteinander überschneiden können. So wird in dieser Arbeit der Begriff des Text Minings als ein Oberbegriff für Anwendungen innerhalb der Informationsextraktion verwendet (für eine genaue Begriffserklärung und Differenzierung siehe Kapitel 2.1). Zudem ist für die Anwendung von Extraktionsalgorithmen eine Vorverarbeitung der vorhandenen Daten notwendig, die mittels Information Retrieval-Methoden durchgeführt werden kann. Das allgemein gefasste Ziel der Anwendung von Text Mining-Methoden ist die Überführung von unstrukturierten Daten, wie sie in natürlichsprachlichen Texten aufzufinden sind, in einheitlich strukturierte Datenstrukturen, wie sie beispielsweise in Datenbankeinträgen verwendet werden. Inwiefern sich Daten in ihrer unterschiedlichen Form auszeichnen, wird in Kapitel 2.1 kurz skizziert.

Text Mining findet in verschiedenen Bereichen Gebrauch. Als ein Beispiel für eine Anwendungsmöglichkeit ist die Domäne der Stellenausschreibungen zu benennen. Stellenausschreibungen können sowohl in Fließtextform als auch in stichwortartigen Elementen formuliert sein und unterliegen meist einer kanonischen Struktur, die sich in verschiedene Kategorien unterteilen kann: Unternehmensbeschreibung, Jobbeschreibung, Bewerber:innenprofil und Formalia. Als Begründung der Anwendung von Text Mining-Methoden kann diese Domäne wertvolle Informationen über die Entwicklung des Arbeitsmarktes liefern. Zu extrahierende und nutzende Informationen aus den Stellenausschreibungen können beispielsweise die Anforderungen an die Bewerber:innen oder das Tätigkeitsprofil der zu besetzenden Stelle sein, die dann Aufschluss darüber liefern, welcher Fokus in den verschiedenen Branchen gelegt wird.

Das Projekt Qualifikationsentwicklungsforschung[1] des Instituts für Digital Humanities der Universität zu Köln (IDH), das in Kooperation mit dem Bundesinstitut für Berufsbildung (BIBB)[2] steht, beschäftigt sich seit Oktober 2015 mit der Aufbereitung, Annotation und Auswertung eines großen Korpus von mehr als hunderttausenden Stellenanzeigen, das vom BIBB zur Verfügung gestellt wird, mithilfe von Methoden aus dem Bereich des Text Minings. Dieses Korpus soll Aufschluss über das Arbeitsmarktgeschehen liefern. Für die Aufbereitung der Daten wurde das IDH beauftragt, die Stellenanzeigen, die in unstrukturierter Form vorliegen, mittels NLP-basierter Verfahren in strukturierte Daten umzuwandeln. Hierfür wurde ein automatisiertes Verfahren zur Informationsgewinnung durch Anwendung eines *Bootstrapping*-Verfahrens, das mit Extraktionsmustern arbeitet, in mehreren Etappen entwickelt (eine detaillierte Beschreibung der verschiedenen Etappen und

[1] https://dh.phil-fak.uni-koeln.de/forschung/qualifikationsentwicklungsforschung
[2] https://www.bibb.de

Entwicklungszustände wird in Kapitel 3.2 aufgezeigt). Die aus den Stellenanzeigen extrahierten Daten sollen daraufhin vom BIBB inhaltlich ausgewertet werden, um so Trendanalysen vorzunehmen, also Entwicklungen innerhalb des Arbeitsmarktgeschehens (z.B. betriebliche Qualifikationsanforderungen, Fachkräftebedarf) sichtbar zu machen.

Ziel dieser Arbeit ist die Weiterentwicklung des bereits bestehenden Formalismus zur Extraktion von den in Stellenanzeigen aufgeführten Bewerber:innenkompetenzen und geforderten Arbeitsmitteln, der durch einen Bootstrapping-Ansatz realisiert wurde. Durch die Bootstrapping-Methodik ist es möglich, auf Basis eines kleinen Korpus annotierter Trainingsdaten und weniger manuell erstellter Extraktionsmuster Vorkommen von Bewerber:innenkompetenzen und Arbeitsmitteln in klassifizierten Stellenausschreibungen automatisiert aufzufinden und zu extrahieren. Die Weiterentwicklung dieses Formalismus wird durch die Evaluation der Extraktionsmuster sowie daraus resultierenden Extraktionen realisiert, die aufgrund ihrer Funktionalität innerhalb des Bootstrapping-Prozesses bewertet werden. Orientierung findet dieser Ansatz am *Snowball*-System (Vgl. Agichtein & Gravano 2000), das bereits bei der Entwicklung des verwendeten Bootstrapping-Algorithmus als Grundlage diente. Kapitel 4.3 beschreibt die Implementierung eines Formalismus zur Bewertung der verwendeten Muster und Extraktionen sowie darauf aufbauend die Evaluierung des Bootstrapping-Systems. Die Auswirkungen der Bewertung der Extraktionsmuster und resultierenden Extraktionen auf die Arbeitsweise und Funktionalität des Algorithmus werden in Kapitel 4.4 aufgezeigt.

Um einen theoretischen Rahmen zu formulieren, wird in Kapitel 2 auf das Gebiet der Maschinellen Sprachverarbeitung allgemein und dessen Teilbereiche, speziell den Bereich des Text Minings, eingegangen. Die Idee des Bootstrapping-Formalismus wird hier anhand aktueller Forschungsstände beispielhaft skizziert und in Bezug zu der beschriebenen Zielsetzung betrachtet. Weiter wird in Kapitel 3 die Domäne der Stellenanzeigen näher beleuchtet und gefragt, inwieweit diese in vorangegangenen Vorverarbeitungsschritten der Klassifikation und der nachfolgenden Verarbeitung des Concept Minings genutzt wird. Zum Schluss wird ein Ausblick auf mögliche Verbesserungen und Erweiterungen zur Evaluierung von Extraktionsmustern gegeben.

2. Text Mining als Bereich der Maschinellen Sprachverarbeitung

Maschinelle Sprachverarbeitung (*Natural Language Processing*, NLP) als Teilbereich der Computerlinguistik befasst sich mit der maschinellen Verarbeitung natürlicher Sprache. Natürliche Sprache kann sowohl gesprochene als auch geschriebene sein, in dieser Arbeit wird der Fokus auf die geschriebene Sprache bzw. auf natürlichsprachliche Texte gelegt. Bei der Bearbeitung der NLP-Aufgabe geht es darum, wie ein Algorithmus so zu programmieren ist, dass er große Mengen natürlichsprachlicher Daten verarbeiten und analysieren kann. So sollen Informationen und Erkenntnisse innerhalb von Dokumenten extrahiert, klassifiziert und organisiert werden. NLP lässt sich in unterschiedliche Anwendungen unterteilen, die alle gemeinsam haben, dass sie über Wissen der jeweils verarbeitenden Sprache verfügen müssen. Wissen über Sprache lässt sich in verschiedene Kategorien einteilen, die sich an den Beschreibungsebenen natürlicher Sprache orientieren. Dazu zählen:

1) Phonetik und Phonologie: Wissen über artikulatorische Merkmale und die Lautstruktur gesprochener Wörter
2) Morphologie: Wissen über die bedeutungsvollen Einheiten[3] von Wörtern
3) Syntax: Wissen über die Strukturbildung von Sätzen, also die strukturellen Relationen zwischen Wörtern
4) Semantik: Wissen über die Bedeutung sprachlicher Einheiten
5) Pragmatik: Wissen über die Relation der Bedeutung einer Äußerung zu der Intention des Sprechers
6) Diskurs: Wissen über linguistische Einheiten, die größer als einfache Äußerungen sind[4]

Mittels linguistischer Methoden können Formalismen, die mithilfe der benannten Wissenskategorien natürlichsprachliche Texte analysieren können, zur Informationsgewinnung eingesetzt werden. Die eingesetzte Methodik hängt mit der spezifischen Problemstellung zusammen und kann sich an Komplexität unterscheiden. Beispielhaft aufzuzeigen sind *Finite-State*-Maschinen[5] und regelbasierte Systeme, die mittels regulärer, kontextfreier oder merkmalsbasierter Grammatiken für phonologische, morphologische oder syntaktische Anwendungen eingesetzt werden können, oder auch probabilistische Modelle, die häufig in

[3] Unter bedeutungsvollen Einheiten von Wörtern sind Wortbestandteile gemeint, die dem Wort bspw. einem Numerus zuordnen: z.B. *doors* (engl.) wird als Plural erkannt, da es durch ein Plural-*s* gekennzeichnet ist.
[4] Beispielhaft aufzuführen ist hier das Wissen über folgende Frage: „Wie viele Tage unter 0 Grad Celsius wurden dieses Jahr auf der Welt vermerkt?“ Das diskursive Wissen ermöglicht die Interpretation des Satzteils „dieses Jahr“ in Bezug auf den restlichen Diskurs und kann dementsprechend eine Antwort ermöglichen.
[5] Finite-State-Maschinen bezeichnen deterministische und nicht-deterministische endliche Automaten, die einen Eingabestring anhand einer zugrundeliegenden Sprache überprüfen und ihn als Wort dieser Sprache erkennen oder nicht (Vgl. Carstensen et al. 2010: 70).

Zusammenhang mit Disambiguierungsaufgaben genutzt werden (Vgl. Jurafsky & Martin 2009: 5f.). Bereiche des NLP, in denen diese Methoden eingesetzt werden, beschäftigen sich zwar mit spezifischen Aufgaben, haben aber viele Schnittstellen, innerhalb derer sie sich die angewandte Methodik teilen. Zum Beispiel sind die Teilgebiete *Text Mining* und *Information Retrieval* (IR) definitorisch getrennt, Anwendungen innerhalb des Text Minings setzen aber Vorverarbeitungsschritte, die im Information Retrieval zu verorten sind, voraus. Information Retrieval kann folgendermaßen definiert werden: „Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)“ (Manning et al. 2008: 1). Das Ziel von IR ist also das Auffinden relevanter Dokumente mit natürlichsprachlichem Inhalt innerhalb eines großen Korpus. Zu den Aufgaben des IR gehören demnach auch Textklassifikation und *Textclustering*, mit denen die Auswahl der Texte auf Relevanz reduziert werden können (Vgl. Neumann 2010: 9). Betrachtet man nun das Aufgabengebiet des Text Minings - der Fokus liegt hier auf der Informationsextraktion als Teilbereich des Text Minings -, handelt es sich hierbei um eine Variante des *Data Minings*[6], bei der es um die Analyse großer linguistisch vorverarbeiteter Textdaten zum Aufbau lexikalischer Ressourcen, wie z.B. domänenspezifischer Ontologien oder sprachbezogener Wörterbücher, geht (Vgl. Biemann & Mehler 2014: 5). Bevor die Analyse jedoch stattfinden kann, müssen domänenrelevante Dokumente zu einem Korpus zusammengestellt werden. Dafür wird auf die Methodik des IR zurückgegriffen. IR wie auch Text Mining können demnach als Bestandteile des NLP angesehen werden, die aber meist in Verbindung zueinander eingesetzt werden.

2.1 Informationsextraktion

Die Informationsextraktion (IE) als Bereich des Text Minings befasst sich mit der maschinellen Verarbeitung natürlicher Sprache. In diesem Fall geht es darum, aus natürlichsprachlichen Texten, die unstrukturierte Daten enthalten, die enthaltenen Daten durch die Extraktion semantischer Relationen in eine strukturierte Form zu bringen.[7] Vergleichbar ist sie mit dem IR, also der Anfrage eines Users und der Suche der gewünschten Information in einem großen

[6] Data Mining bezeichnet allgemein den Prozess der Extraktion von Wissen aus sehr großen Datensätzen (Vgl. Bidgoli 2002: 477).

[7] Unter unstrukturierten Daten werden im informatischen Bereich verkettete Entitäten wie Sätze, Einträge oder Dokumente gemeint. Auch zählen maschinengenerierte Seiten, wie HTML-Dokumente, die einen Datenbank-Bezug haben (sog. *Wrapper*), dazu. Die Daten innerhalb dieser werden als ein Set aus strukturierten Feldern angesehen. Bei unstrukturierten Daten ist die Herausforderung, bedeutungsvolle Einheiten zur Weiterverarbeitung herauszufiltern. Im Vergleich werden mit strukturierten Daten Entitäten, Relationen und Events gemeint, denen eine domänenabhängige Relevanz zugeordnet werden kann (Vgl. Sarawagi 2007: 269ff.).

Korpus, sie unterscheiden sich aber in der Darstellung des Outputs, da bei der IE das Füllen einer Datenbank oder eines *Templates* (dt. Vorlagen) im Vordergrund steht. Die Anwendung einer Informationsextraktion geht aber in den meisten Fällen einher mit der Anwendung einer IR-Anfrage, bei der relevante Texte für die Extraktion herausgefiltert werden (Vgl. Appelt & Israel 1999: 4). IE wurde erstmals 1994 als eine neue NLP-Technologie innerhalb der *Message Understanding Conference* (MUC) eingeführt und als zukünftig schwierige Aufgabe bezeichnet (Vgl. Grishman 2019: 678). Folgende Ankündigung gibt eine erste Definition über die Beschaffenheit der IE-Aufgabe:

> The Information explosion of the last decade has placed increasing demands in processing and analyzing large volumes of online data. In response, the Advanced Research Projects Agency (ARPA) has been supporting research to develop a new technology called IE. IE is a type of document processing which captures and outputs factual information contained within a document. Similar to an information retrieval system, an IE system responds to a user's information need. Whereas an Information Retrieval (IR) system identifies a subset of documents in a large text database or in a library scenario a subset of resources in a library, an IE system identifies a subset of information within a document. (Okurowski 1993)

Der Ablauf der IE soll sich durch eine automatische Identifikation und Klassifikation der Textinstanzen, die die gesuchten relevanten domänenspezifischen Informationen enthalten, realisiert werden. Die extrahierten Informationen sollen daraufhin allgemein-gültige, universelle Templates füllen und anhand ihrer Genauigkeit bewertet werden. Der Output der Extraktion, also die strukturierten Daten, die beispielsweise in Form von Datenbankeinträgen realisiert werden können, können daraufhin von anderen Anwendungen verwendet werden. Bei der IE sind die genutzten und extrahierten Daten meist auf eine Domäne beschränkt, um die Aufgabe überschaubar zu halten und die Interpretation der Ergebnisse zu erleichtern. Additiv dazu steht die Open IE. Aufgrund der vermehrten Onlinezugänglichkeit wurde der Zugang zu unstrukturierten und strukturierten Daten erweitert, so dass die Open IE auch webbasierte domänenoffene Texte miteinbezieht, die nicht notwendigerweise eine kanonische Struktur, wie sie bei strukturierten Daten vorzufinden ist, aufweisen und komplexeres linguistisches Wissen benötigen (Vgl. Grishman 2019: 677; Sarawagi 2007: 261).

IE gliedert sich in verschiedene Teilaufgaben, die für die Extraktion unterschiedlicher Informationen in Form strukturierter Daten zuständig sind. Typischerweise gliedert sich IE in folgende Teilaufgaben (nach Jurafsky & Martin 2009):

- Eigennamenerkennung (*Named Entity Recognition*, NER)
- Relationsextraktion (*Relation Extraction*)
- Templatefüllung (*Template Filling*)
- Ereignisextraktion (*Event Extraction*)
- Extraktion zeitlicher Ausdrücke (*Temporal Expression Extraction & Normalization*)

Die jeweiligen Teilaufgaben haben gemeinsam, dass sie sich mit dem Aufspüren bekannter Entitäten bzw. Eigennamen in unstrukturierten Daten und mit dem Versehen dieser Entitäten mit Labeln beschäftigen. Bei Entitäten werden typischerweise Nominalphrasen betrachtet, die (klassischerweise) Eigennamen wie die von Personen, Orten, Organisationen etc. darstellen.[8] Innerhalb der MUC-6 wurden drei Subentitäten definiert, die es zu erkennen gab:

1) *ENAMEX* bezeichnen Namen von Personen, Orten und Organisationen. Diese können sich wiederum in verschiedene Subtypen gliedern. Beispielsweise kann Stadt, Staat oder Land als Subtyp für Ort verwendet werden.
2) *TIMEX* bezeichnen zeitliche Ausdrücke wie z.B. Datum und Zeit.
3) *NUMEX* bezeichnen numerische Entitäten wie Geld oder Prozent.
   (Vgl. Nadeau & Sekine 2007: 3)

Heutzutage werden aber noch weitere Entitäten bei der IE-Aufgabe betrachtet. Das Feld der *Generics*, also Entitäten, die z.B. Krankheiten, Proteine, Buchtitel u.a. bezeichnen, hat seit Beginn an großen Aufschwung erlebt. Die Funktionalität des verwendeten Formalismus hängt von den annotierten Erkennungs- und Klassifikationsregeln ab, mit denen Entitäten im Text aufgefunden werden sollen. Solche Regeln werden in Form von distinktiven Merkmalen, also Positiv- und Negativ-Beispielen, erstellt. Bei der Relationsextraktion werden semantische (meist binäre) Relationen zwischen Entitäten, die während einer NER aufgefunden werden, in Form von Tupeln (Entitäten-Relationen-Entitäten) aufgespürt und klassifiziert. Beschränkt werden die Relationen (zumeist) auf das Auftreten der Entitäten innerhalb eines Satzes. Die Bestandteile der Zielrelation werden im Vorhinein aufgabenspezifisch festgelegt. Ein klassisches Beispiel dafür ist die Relation <ORGANIZATION, LOCATION>. So könnte beispielhaft aus dem Satz „Das Institut für Digital Humanities, das an der Universität zu Köln angesiedelt ist, heißt eine neue Kollegin willkommen." die Relation <Institut für Digital Humanities, Köln> extrahiert werden. Auch ist es möglich, komplexe Relationen[9] anhand von Ereignisextraktion oder Template-Filling zu extrahieren. Bei der Ereignisextraktion werden Vorkommen mehrerer, sich im Text wiederholender Events aufgesucht. Das Template-Filling beschäftigt sich mit der Darstellung des Outputs. Dabei werden existierende Templates mit den extrahierten Informationen gefüllt. Templates bestehen aus einer festen Menge von zu

[8] Eigennamen sind sprachliche Ausdrücke, die auf Individuen von Klassen oder Typen bestimmter Entitäten referenzieren. Die maschinelle Extraktion von Eigennamen birgt die Herausforderung der Festlegung, wodurch eine Entität ausgezeichnet wird und der Ambiguität von Eigennamen, die durch *Polysemie* (derselbe Name verweist auf mehrere Entitäten) oder *Synonymie* (dieselbe Entität kann mehrere Namen haben) erzeugt wird. Mittels Disambiguierungsabläufen soll diese Herausforderung gemeistert werden (Vgl. Neumann 2010: 23).
[9] Komplexe Relationen bezeichnen beliebige mehrstellige Relationen mit mehreren unspezifizierten Argumenten (Vgl. Neumann 2010: 28).

füllenden Feldern (*Slots*), die mit Extraktionen des Textes eines bestimmten Datentyps gefüllt werden müssen (Vgl. Jurafsky & Martin 2009: 766). Im Laufe der Weiterentwicklung der IE-Aufgabe, die eine Ausweitung der domänenspezifischen Anwendungen mit sich zog, wurden auch die verwendeten Templates auf sprachlicher und thematischer Ebene immer komplexer. Während der MUC-6 wurde erstmals ein vollständiges System vorgestellt, das Event-Templates füllen konnte (Vgl. Grishman 2019: 680). Abbildung 1 veranschaulicht die Anwendung eines einfachen Template-Fillings auf ein Textbeispiel. Im Text sind jeweils die extrahierten Entitäten hervorgehoben.

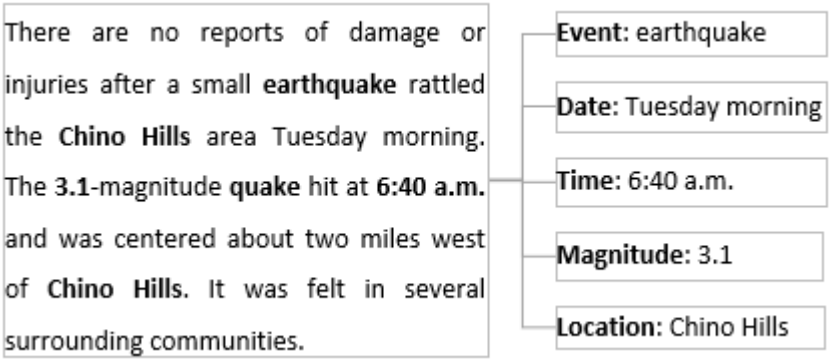


Abbildung 1: Beispiel eines einfachen Template-Fillings (Quelle: Jean-Louis et al. 2011)

Um die Anwendbarkeit der verschiedenen Teilbereiche der IE zu gewährleisten, müssen zuvor einige Vorverarbeitungsschritte durchgeführt werden, die das zugrundeliegende Korpus in eine weiterverarbeitende Form bringt. Der erste Schritt ist die Zerlegung des Textes in thematische Sinneinheiten (*Zoning*) wie Paragrafen und linguistische Einheiten wie Sätze und einzelne Tokens.[10] Darauf aufbauend wird eine morphologische und lexikalische Verarbeitung der einzelnen Teile vorgenommen. Mittels *Part of Speech Tagging* (POS-Tagging) werden die grammatischen Kategorien von Wörtern ausgezeichnet, und anhand von *Word-Sense Tagging* wird versucht, Disambiguierungsprobleme zu lösen, die bei der Verarbeitung von natürlicher Sprache immer vorkommen können. Zudem werden die Grundformen flektierter Wörter (*Lemmata*) und ggf. andere wortgebundene Merkmale wie *Tempus* oder *Modus* von Verben bestimmt. Die nun vorhandenen Daten werden daraufhin einer syntaktischen Analyse unterzogen, d.h. dass mithilfe eines *Parsers* natürlichsprachliche Ausdrücke, die innerhalb des Textes annotiert wurden, auf ihre Grammatikalität überprüft werden und mit einer Interpretation in Form einer syntaktischen Strukturbeschreibung, wie z.B. einem Baumdiagramm, versehen werden. Auf die bislang domänenunabhängigen Verarbeitungsschritte können danach domänenspezifische Analyseschritte aufgebaut werden.

[10] Unter Tokens versteht man üblicherweise ein Wort, eine Stelle oder ein Satzzeichen. Je nach Kontext kann sich die Definition eines Tokens unterscheiden.

So werden in diesem Arbeitsschritt die vorhandenen Informationen genutzt, um die relevanten Entitäten zu extrahieren, die innerhalb der Informationsabfrage benannt wurden. Anschließend werden die Entitäten zum Füllen der Templates genutzt. Die domänenspezifische Extraktion kann durch verschiedene NLP-Techniken wie *Pattern Matching*[11] oder probabilistische Methoden durchgeführt werden. Die Methodik hängt von der spezifischen Aufgabenstellung ab (Vgl. Appelt & Israel 1999: 13ff.). Abbildung 2 veranschaulicht die wesentlichen Komponenten der Vorverarbeitung eines IE-Systems.

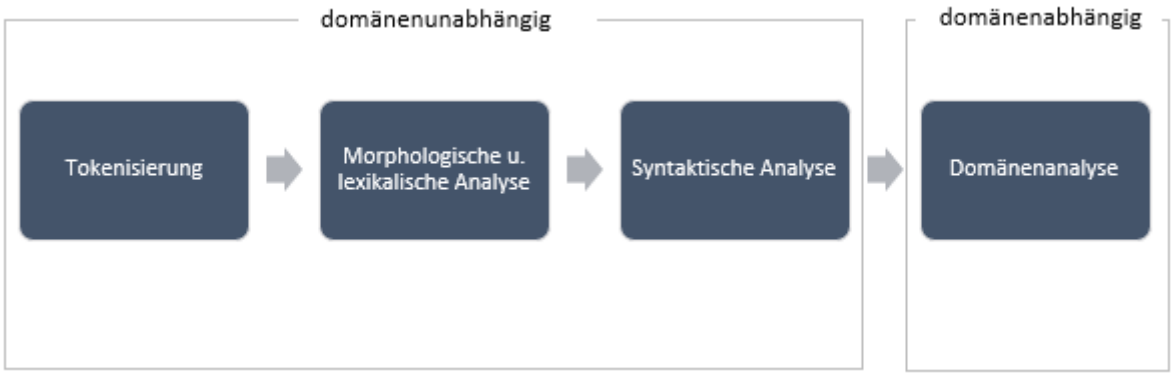


Abbildung 2*:* Grundlegende Komponenten der Vorverarbeitung eines IE-Systems

Die Ausführung der Informationsextraktion kann sich durch verschiedene Varianten von Algorithmen gestalten, die jeweils unterschiedlich starke Supervision benötigen. IE kann sowohl durch manuelle als auch durch maschinelle Lernverfahren angewendet werden. Die Nutzung manuell erstellter Extraktionsmuster ist die wohl älteste und gebräuchlichste Anwendung der Informationsextraktion, aber auch die zeit- und kostenintensivste. Dabei ist es wichtig, eine optimale Bestimmung der Merkmale vorzunehmen, die im weiteren Verlauf extrahiert werden sollen. Da diese Arbeit domänenspezifische Experten benötigt, wird heutzutage das maschinelle Lernverfahren, das Extraktionsregeln automatisch generieren kann, eingesetzt. Sowohl manuelle wie auch maschinelle Lernverfahren können auf Basis regelbasierter oder statistischer Methoden ausgeführt werden. Regelbasierte Methoden erlernen bspw. anhand von manuell annotierten oder gelabelten Beispielen ein Set aus Regeln, das sie im weiteren Verlauf für die Extraktion einsetzen, statistische Modelle basieren wiederum auf der statistischen Auswertung von Trainingsbeispielen (Vgl. Sarawagi 2007: 278f.). Zum Beispiel kann ein Lernalgorithmus die Klassifikation von positiven oder negativen Filmkritiken anhand von Beispielen erlernen, indem er die Wortfrequenzen der Texte ermittelt sowie die

[11] Unter dem Terminus Pattern Matching versteht man das Auffinden von Entitäten in einem von Extraktionsmustern vordefinierten Kontext. Die Nutzung von Pattern Matching innerhalb eines Bootstrapping-Formalismus ermöglicht zudem das Auffinden neuer unbekannter Entitäten.

Wahrscheinlichkeit, mit der ein Wort für eine positive oder negative Kritik steht. Neben den manuellen Lernverfahren nehmen die maschinellen Lernverfahren eine große Rolle bei der Informationsextraktion ein. Bei der jeweiligen Bezeichnung der verschiedenen Arten von maschinellen Lernverfahren wird der Grad der notwendigen Betreuung der genutzten Formalismen unterschieden.

Überwachte maschinelle Lernverfahren sind durch folgende Merkmale gekennzeichnet: „Texts are annotated with relations chosen from a small fixed set by human analysts. These annotated texts are then used to train systems to reproduce similar annotations on unseen texts“ (Jurafsky & Martin 2009: 749). Bei der Anwendung überwachter Lernverfahren ist es also notwendig, dass eine manuell annotierte Basis zur Verfügung steht, die für das Training eines Klassifikators eingesetzt wird, damit dieser innerhalb unbekannter, noch nicht annotierter Texte Entitäten extrahieren und klassifizieren kann. Standardisierte Klassifikationstechniken sind beispielsweise *Support Vector Machines* (SVM), *Hidden Markov Modelle* (HMM) oder *Conditional Random Fields* (CRF). Der Vorteil an überwachten Lernverfahren ist die hohe Genauigkeit der Extraktion bei ausreichend großen Mengen an manuell annotierten Daten. Gleichzeitig ist die Anwendung mit sehr hohen Kosten verbunden, da große annotierte Daten meist nicht zugänglich sind und dementsprechend erstellt werden müssen. In Anbetracht der fortschreitenden Entwicklung von IE-Systemen ist dies wohl der größte Nachteil, da semi- und unüberwachte Lernverfahren mit weniger Aufwand vergleichbare Ergebnisse liefern können.

Die Anwendung von semiüberwachten Lernverfahren wie bspw. Bootstrapping-Formalismen (Bootstrapping wird näher in Kapitel 2.2 beleuchtet) benötigt nur einen kleinen Teil an Supervision: ein kleines Set an manuell annotierten Seeds (dt. Saatgut), um den Lernprozess zu starten. Wenn z.B. ein System darauf spezialisiert werden soll, innerhalb natürlichsprachlicher Texte Eigennamen von Krankheiten aufzusuchen, und vom User eine kurze Liste mit beispielhaften Entitäten zur Verfügung gestellt bekommt, kann es mithilfe dieser die zugrundeliegenden Texte auf Vorkommen dieser Entitäten untersuchen und dessen Kontexte identifizieren, mit denen es wiederum Vorkommen neuer Entitäten suchen kann (Vgl. Nadeau & Sekine 2007: 5f.).

Neben überwachten und semiüberwachten Lernverfahren lassen sich noch die unüberwachten Lernverfahren nennen. Diese werden größtenteils innerhalb der Open IE angewendet, da hierfür Entitäten und Relationen von Entitäten extrahiert werden, ohne gelabelte Trainingsdaten oder Relationslisten zu nutzen. Das bedeutet, dass unüberwachte Lernverfahren nicht auf domänenspezifische Entitäten beschränkt sind und sich dementsprechend für die Open IE

besonders gut eignen, da diese besonders domänenoffene Texte in den Blick nimmt. Der Vorteil an diesen Verfahren ist außerdem, dass mit einer großen Anzahl an Relationen gearbeitet werden kann, ohne diese spezifizieren zu müssen, und deshalb, wie auch bei den semiüberwachten Lernverfahren, kein Fachexperte benötigt wird (Vgl. Jurafsky & Martin 2009: 756). Ein typischer Ansatz für unüberwachte Lernverfahren ist das sog. *Clustering*, bei dem mithilfe der Ähnlichkeit von Kontexten Eigennamen ermittelt werden. Es gibt auch noch weitere Methoden, die aber alle gemeinsam haben, dass sie auf verfügbaren lexikalischen Ressourcen basieren und lexikalische Muster oder Statistiken nutzen, die über große unannotierte Korpora erstellt wurden. (Vgl. Nadeau & Sekine 2007: 6f.).

2.2 Semiüberwachte Lernverfahren: Bootstrapping-Formalismus

Bei der Entwicklung von IE-Systemen ist es notwendig, die verfügbaren Ressourcen zu betrachten. Unter Betrachtung dieses Aspekts können zwei verschiedene grundlegende Ansätze angewendet werden, die die Funktionalität der bereits beschriebenen Lernverfahren bündeln. Zu unterscheiden sind einerseits der *Knowledge Engineering*-Ansatz, dem die überwachten Lernverfahren zuzuordnen sind, denn bei diesem Ansatz entwickelt ein „knowledge engineer" (Appelt & Israel 1999: 7), also eine Person, die sich sowohl mit IE-Systemen als auch mit der spezifischen Domäne auskennt, mithilfe eines Zugangs zu einem ausreichend großen Korpus domänenrelevanter Texte Grammatiken für den einzusetzenden Formalismus. Bei der Entwicklung dieser Grammatiken handelt es sich um einen iterativen Prozess, da die erstellten Regeln der Grammatik immer wieder hinsichtlich ihrer Funktionalität (Über- oder Untergenerierung) aktualisiert werden müssen. Mithilfe präziser Regeln können Knowledge Engineering-Formalismen eine höhere Performance als automatisches Training erreichen. Dem gegenüber steht der Ansatz des *Automatischen Trainings*, dem sowohl die semi- als auch unüberwachten Lernverfahren zugehörig sind. Es wird kein Experte für IE-Systeme wie auch die Domäne benötigt, der Formalismus wird anhand eines kleinen annotierten Trainingskorpus für seine entsprechende Aufgabe trainiert (Vgl. Appelt & Israel 1999: 7ff.).

Bootstrapping als eine Methode zur Anwendung eines semiüberwachten Lernverfahrens hat sich in den vergangenen Jahren als effektiver Ansatz und als Alternative in der automatischen Texterkennung etabliert. Beim Bootstrapping handelt es sich um eine iterative Methode, die zur automatischen Extraktion von Entitätsrelationen eingesetzt wird, indem der Output des Systems für die Generierung des Trainingsinputs der nächsten Iteration genutzt wird. Das Verfahren wurde entwickelt, um das Problem limitierter Trainingsdaten zu lösen, das bereits bei der

Beschreibung der verschiedenen Lernverfahren angesprochen wurde, denn in diesem Fall wird als Ausgangsbasis kein großes Korpus annotierter Trainingsdaten benötigt, sondern nur ein kleines Set von bereits bekannten, domänenspezifischen Eigennamen (Seeds)[12] und ein unannotiertes Korpus (Vgl. Sarawagi 2007; Sun 2009). Mithilfe dieses Sets als Ausgangspunkt lässt sich ein iterativer Ablauf skizzieren:

1) Das Korpus wird auf Vorkommen des übergebenen Sets von Seeds untersucht.
2) Extraktionsmuster werden durch die Bestimmung der Merkmalskombinationen der Kontexte, in denen Seeds vorkommen, automatisch generiert.
3) Das Set von Seeds wird durch die Anwendung der automatisch generierten Muster erweitert.

Die Anwendung des Bootstrapping-Verfahren ist vielfältig. Eine bekannte Anwendung, die auch als Orientierung bei der Entwicklung vieler verwandter Formalismen diente, ist das IE-System *Snowball*, das zum Aufsuchen binärer Relationen der Art <ORGANIZATION, LOCATION> eingesetzt werden kann. Es erfolgt nach dem oben skizzierten Ablauf und beinhaltet zusätzlich eine automatische Qualitätskontrolle der verwendeten und generierten Muster und Tupel bei jeder Iteration, um dem Problem des *Semantic Drifting*[13] entgegenzusetzen. Dabei werden Muster und Tupel jeweils nach ihrer Selektivität bewertet. Das Verfahren wird solange angewendet, bis keine neuen Muster oder Entitäten mehr entdeckt werden (Vgl. Agichtein & Gravano 2000). Abbildung 3 veranschaulicht den Ablauf des *Snowball*-Verfahrens und zeigt auf, an welchen Stellen die Evaluation der Muster und Seeds einsetzt.

[12] Die als Ausgangsbasis verwendeten Entitäten können auch als *Goldstandard* bezeichnet werden. Die Komplexität des Goldstandards kann unterschiedliche Ausmaße haben. Er bezeichnet eine Menge an handannotierten Entitäten, denen korrekte Kategorien zugeordnet wurden und die zur Überprüfung des Systems genutzt werden können (Vgl. Manning et al. 2008: 152ff.).

[13] Semantic Drifting bezeichnet die iterative Wiederholung von Fehlern. Durch fehlerhafte Muster, die beim Bootstrapping automatisch generiert werden und keiner manuellen Überprüfung unterliegen, können fehlerhafte Tupel extrahiert werden, die wiederum zum Bilden problematischer Muster führen (Vgl. Jurafsky & Martin 2009: 755). Deshalb muss darauf geachtet werden, dass die Muster bei der automatischen Generierung den Grad zwischen Spezifikation und Allgemeinheit ausbalancieren.

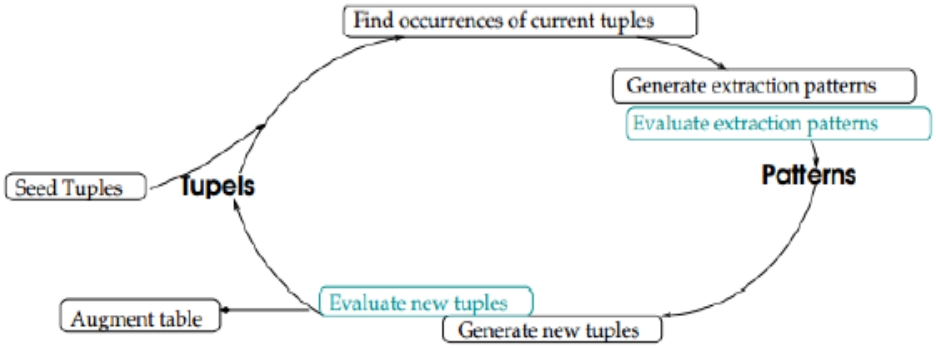


Abbildung 3: Schematische Darstellung des *Snowball*-Systems nach Agichtein & Gravano 2000 (Quelle: Geduldig 2017)

Das *Snowball*-System ist aber nicht der erste Versuch, ein Bootstrapping-Verfahren zur Extraktion von Entitätsrelationen einzusetzen. Als Grundlage bei der Entwicklung dieses Systems diente das System *DIPRE* (Brin 1998), das zur Extraktion von Relationspaaren in Form <AUTHOR, TITLE> aus dem *World Wide Web* (WWW) eingesetzt wurde. Gemeinsam haben sie die Grundlage eines kleinen Sets von bekannten Entitäten, mit denen ihre Vorkommen in einem großen Korpus aufgesucht werden. Die besondere Herausforderung beim *DIPRE*-System ist die Verwendung des WWW als Datenbasis, da die Informationen in dieser riesigen Ressource durch vielfältige Typen, sog. „chunks of information" (Brin 1998: 1), dargestellt werden, die auf viele verschiedene Quellen verteilt sind. Im Vergleich zu vorhandenen Ansätzen, die durch den Einsatz von Wrappern[14] sehr zeitaufwendig sind, ist der Gebrauch eines Bootstrapping-Formalismus mit deutlich weniger Arbeit gekennzeichnet.

Als Vorläufer des Einsatzes von Bootstrapping-Formalismen zur Extraktion von Entitätspaaren kann der korpusbasierte Versuch von Riloff & Sheperd (1997) verzeichnet werden, bei dem auf Basis eines kleinen Sets an Entitäten, das jeweils eine spezifische Kategorie repräsentiert, und einem repräsentativen Korpus ein semantisches Lexikon[15] für die ausgewählten Kategorien gebildet wird. Die Idee für diesen Ansatz boten semantische Wissensgrundlagen wie *WordNet*[16] oder *Cyc*[17], die jedoch nur für eine Handvoll von Anwendungen sinnvoll einzusetzen sind, da

[14] Wrapper finden in der Informationstechnik unterschiedliche Anwendung. In diesem Fall beziehen sie sich auf einen Formalismus, mit dem relevante Dokumente im Web anhand einer Reihe von vordefinierten Begriffen aufgesucht werden können (Vgl. Gudivada & Tolety 1997: 305). Wrapper haben häufig den Nachteil, dass sie auf die Generierung von speziellen *Extractors* für Webseiteninhalte spezifiziert sind und dementsprechend bei Veränderung dieser zeit- und kostenintensiv angepasst werden müssen (Vgl. Chang et al. 2003: 130).
[15] Ein semantisches Lexikon bezeichnet eine Sammlung von Wörtern, die mit einer semantischen Klasse gelabelt wurden (Vgl. Thelen & Riloff 2002: 214). Das Label ist domänenspezifisch und kann beispielsweise die Zuordnung zu der Klasse „Krankheiten" bezeichnen.
[16] https://wordnet.princeton.edu/
[17] https://www.cyc.com/

die vorhandenen Daten weder die den domänenspezifische Subsprachen eigenen Terme noch Jargons beachten, die für die Aufgabe und Genauigkeit der Informationsextraktion eine wichtige Rolle spielen. Der hier eingesetzte Algorithmus nutzt sowohl einfache Statistiken als auch einen Bootstrapping-Algorithmus, die zusammengenommen eine gerankte Liste mit Wörtern, die der jeweiligen Kategorie zuzuordnen sind, erstellen. Eine manuelle Überarbeitung dieser Liste beendet den Ablauf (Vgl. Riloff & Sheperd 1997).

Eine Erweiterung des Ansatzes zur Erstellung eines semantischen Lexikons mithilfe des Bootstrapping-Verfahrens ist bei Riloff (1999), Thelen & Riloff (2002) und Chang et al. (2003) vorzufinden. Die Ansätze nutzen alle Extraktionsmuster, um die Kontexte der gesuchten Entitäten zu untersuchen. Riloff (1999) integriert einen Ansatz, den sie *Multi-Level Boostrapping* nennt. Er erweitert die bereits existierenden *Single-Stage-Bootstrapping*-Verfahren (Vgl. Sun 2009: 76) um eine zweite Ebene: Der entwickelte Algorithmus erstellt anhand eines unannotierten Korpus und eines Sets von Seeds für eine Kategorie sowohl ein semantisches Lexikon als auch eine Sammlung von Extraktionsmustern für die jeweils spezifische Domäne. Dafür wird das Verfahren in ein Mutual bzw. Inneres und ein Multi-Level bzw. Meta Bootstrapping aufgeteilt. Das Mutual Bootstrapping ist für das Aussuchen der besten Extraktionsmuster einer Kategorie und das Füllen des semantischen Lexikons zuständig. Dieser Ablauf wird in ähnlicher Weise in den bereits vorgestellten Ansätzen durchgeführt. Laut Riloff sei aber dieser Ablauf allein anfällig für fehlerhafte Extraktionen, weshalb sie eine zweite Ebene, das Meta Bootstrapping, hinzufügt, um so die fünf zuverlässigsten Lexikoneinträge zu ermitteln, die innerhalb der ersten Ebene extrahiert wurden. Diese werden dann für die nächste Iteration den Seeds hinzugefügt. Vergleichbar ist diese Methodik mit dem *Snowball*-System (Agichtein & Gravano 2000), wobei hier keine zweite Ebene und eine andere Selektionsgrenze definiert wird.

Basilik, der Algorithmus der bei Thelen & Riloff (2002) trainiert wird, stellt Hypothesen über semantische Klassen eines Wortes auf, indem er kollektive Erkenntnisse über semantische Verbindungen anhand der Kontexte, die über die Extraktionsmuster definiert werden, sammelt. Mithilfe des Algorithmus *AutoSlog*[18] (Riloff 1996) werden automatisiert Extraktionsmuster, die jede Nominalphrase (NP) des übergebenen, unannotierten Textkorpus, die die manuell

[18] *AutoSlog* ist ein System, das automatisch Extraktionsmuster mithilfe heuristischer Regeln erstellt. Dafür benötigt es Informationen über die zu tätigenden Extraktionen in Form von annotierten NPs mit domänenspezifischen Labeln. Übergibt man *AutoSlog* eine gelabelte NP und einen Text aus dem Korpus, so identifiziert das System die Sätze des Textes, die die übergebene NP enthalten. Falls mehr als ein Satz die NP enthält, wird nur der erste Satz weiterverwendet. Zur Identifikation von Satzgrenzen und syntaktischen Konstituenten hat es einen Satzanalysator (*CIRCUS*, Lehnert 1991) integriert. Daraufhin werden heuristische Regeln angewendet, die als Ergebnis neue Extraktionsmuster hervorbringen (Vgl. Riloff 1996).

ausgewählten Seedwörter enthält, erstellt. Durch die Durchführung einer Bewertung basierend auf der Funktionalität der Extraktionsmuster als auch der daraus resultierenden Extraktionen werden die besten extrahierten Entitäten dem semantischen Lexikon hinzugefügt (Vgl. Thelen & Riloff 2002).

In Abgrenzung zu diesem Ansatz findet sich die Methode von Chang et al. (2003) wieder, die mithilfe automatischer Extraktionsmustererstellung Informationen aus semistrukturierten Webseiten extrahiert. Dieser Ansatz nutzt im Gegensatz zu den bereits vorgestellten keine handannotierten Trainingsdaten. Ziel dieser Methode ist die Erstellung eines Mustererkennungsalgorithmus, der über jede Webseite, unabhängig ihres einzigartigen Layouts, laufen und spezifische Extraktoren automatisch generieren kann. Auch wenn diese Methoden keine manuell gelabelten Trainingsbeispiele nutzt, wird die Zugehörigkeit zu den semiüberwachten Lernverfahren durch eine manuelle Überarbeitung der automatisch generierten Muster gewährleistet, bei der nur die Muster ausgewählt werden, die dem Nutzer für die Aufgabe sinnvoll erscheinen (Vgl. Chang et al. 2003).

Klassische Anwendung von Bootstrapping-Formalismen zur Extraktion und Klassifikation von NEs, wie bereits zu Beginn des Kapitels beschrieben, finden sich sowohl bei Collins & Singer (1999) als auch bei Sun (2009). Erstere nutzen sieben einfache Rechtschreib- und Kontextregeln zur Klassifikation von NEs, die anhand eines Inputstrings (in diesem Fall ist der Input ein NE) den Typ dieses ermitteln können. Ihr Modell basiert auf zwei Algorithmen:

1) *Word-Sense Disambiguierung*[19] (WSD) mit heuristischen Modifikationen
2) Kombination von zwei einfachen Klassifikatoren

Mithilfe der Einführung eines Kategorien-*Scores* werden die fünf besten Extraktionen ermittelt und als Input für die nächste Iteration genutzt. Dieses Verfahren ermöglicht die Vermeidung einer manuellen Überprüfung der Extraktionen, wobei die Genauigkeit der Selektion großen Einfluss auf die Qualität des Outputs einnimmt (Vgl. Collins & Singer 1999). Sun (2009) wiederum greift den Ansatz von Riloff (1999) mit einer zweiten Ebene innerhalb des Bootstrapping-Ablaufs auf, nutzt diese aber nicht zur Evaluation der Extraktionen. Innerhalb der ersten Ebene werden bereits die genutzten Extraktionsmuster und daraus resultierenden Extraktionspaare der Art <EMPLOYMENT, ORGANIZATION> im Anklang an das *Snowball*-System evaluiert. In der zweiten Ebene werden dann die nominalen Extraktionsmuster mit einer guten Leistung anhand heuristischer Methoden ausgewählt, um

[19] Auf das Problem von ambigen Entitäten in der Informationsextraktion wird in Kapitel 3.2 in Anbetracht des aktuellen Standes des Projekts „Qualifikationsentwicklungsforschung“ eingegangen.

damit Nominalrelationsabfragen zu konstruieren, die wiederum neue Nominale innerhalb des Korpus aufsuchen. Diese werden mithilfe der Bewertungsmechanismen der ersten Ebene evaluiert und daraufhin für die Suche nach neuen Entitätsrelationen eingesetzt (Vgl. Sun 2009).

All diese Anwendungen von Bootstrapping-Verfahren zeigen, dass mittels weniger Daten (in den meisten Fällen) in Form eines kleinen Sets an Seeds und eines unannotierten Korpus als Ausgangssituation die IE-Aufgabe mehr als hinreichend erfüllt werden kann. Anhand dieser Daten werden über einen iterativen Ablauf sowohl die Kontexte der gesuchten Entitäten aufgedeckt wie auch aus diesen Extraktionsmuster erstellt, mit denen neue Entitätspaare gefunden werden können. Zur Überwindung der Integration einer Überprüfung der Funktionalität der Formalismen durch einen Experten werden Selektions- bzw. Bewertungssequenzen in den Ablauf eingefügt, die jeweils die besten bzw. effektivsten Extraktionsmuster und daraus resultierenden Extraktionen für die nächste Iteration filtern. In den meisten Fällen bedarf es trotzdem einer manuellen Überprüfung der Ergebnisse, um zum einen die Funktionalität des Algorithmus verbessern zu können, indem fehlerproduzierende Stellen erkannt und verändert werden, und um zum anderen die extrahierten Entitäten auf ihre Richtigkeit zu überprüfen. Zusammenfassend ist der Einsatz eines Bootstrapping-Formalismus bei der Erstellung semantischer Lexika und der Extraktion von Entitäten bzw. Entitätsrelationen eine zeitsparendere Methode als überwachte Verfahren, da er bei der Bewertung der Ergebnisse (theoretisch) keinerlei Expertise benötigt und sich somit von überwachten Lernverfahren unterscheidet.

2.3 Evaluierungsmaße

Die Evaluation von IE-Systemen zeigt auf, wie gut ein entsprechendes System in Anbetracht seiner spezifischen Aufgabe funktioniert. Sie ist einerseits abhängig von der Genauigkeit der Entitätsextraktion, andererseits von der Schwierigkeit der zugrundeliegenden Aufgabe (Vgl. Grishman 2019; Neumann 2010). Innerhalb der MUC-3 (1991) wurden erstmalig die Maße *Precision* (Relevanz) und *Recall* (Vollständigkeit) aus dem Bereich der IR zur Evaluation von IE-Systemen adaptiert. Bei den Metriken werden sowohl die richtig extrahierten wie auch falsch extrahierten Entitäten betrachtet. Im Kontext einer NER-Aufgabe bezeichnet Precision das Verhältnis zwischen der Anzahl an korrekt gelabelten Entitäten [auch genannt *true positives* (TPs)] und der Gesamtzahl der gelabelten Entitäten [darunter fallen unter anderem die TPs aber auch die sog. *false positves* (FPs), also die Entitäten, die fälschlicherweise als der gesuchten Kategorie zugehörig gelabelt wurden]. Recall beschreibt stattdessen das Verhältnis der Anzahl

der korrekt gelabelten Entitäten (TPs) und der Gesamtzahl aller gesuchten Entitäten, auch derer, die nicht als der gesuchten Kategorie zugehörig gelabelt wurden. Diese werden als *false negatives* (FNs) beschrieben (Vgl. Jurafsky & Martin 2009: 756f.).[20] Formel 2.1 veranschaulicht die Ermittlung von Precision und Recall:

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

$$recall = \frac{true\ positives}{true\ positves + false\ negatives} \tag{2.1}$$

Bei der Evaluation eines IE-Systems ist es demnach wichtig, sowohl eine möglichst hohe Vollständigkeit wie auch Genauigkeit der Extraktionen hervorzubringen. Betrachtet man die beiden Kriterien einzeln, ist es schwierig, beide gleichermaßen zu optimieren. Beide Maße stehen in einem wechselseitigen Verhältnis: So kann etwa eine hohe Precision nur schwer ohne ein Einbüßen des Recalls erreicht werden. Gleiches gilt umgekehrt (Vgl. Chinchor 1991: 22f.). Da die Performance eines IE-Systems aber abhängig davon ist, dass beide Maße gleichermaßen gut arbeiten, wurde im Kontext der MUC-4 (1992) das *F-Maß* als gewichtetes harmonisches Mittel zwischen Precision (P) und Recall (R) entwickelt. Das F-Maß wird wie folgt beschrieben:

$$F = \frac{(ß^2 + 1)\ \times\ P\ \times\ R}{(ß^2\ \times\ P) + R} \tag{2.2}$$

Mithilfe des Parameters ß können Precision und Recall unterschiedlich stark gewichtet werden. Werden beide Maße gleich stark gewichtet, ist ß = 1. Man bezeichnet diesen Zustand auch als *F1-Maß*. Wird wiederum Recall höher gewichtet als Precision, ist ß > 1, andersrum ist ß < 1 (Vgl. Chinchor 1992). Die Adaption der Evaluationsmaße aus dem Bereich der IR in den der IE ist nicht ohne Anpassungen vorzunehmen. So kann es beispielsweise vorkommen, dass eine Dichotomie zwischen korrekt und inkorrekt, wie sie bei der Anwendung der Maße gefordert wird, bei einer IE-Aufgabe nicht unmittelbar gegeben ist. Betrachtet man beispielsweise eine extrahierte Phrase, die in Teilen mit der annotierten Vorgabe übereinstimmt oder sie enthält, kann nur schwer entschieden werden, ob diese Phrase nun als korrekt oder inkorrekt bewertet

[20] In Bezug auf die Anwendung eines Bootstrapping-Formalismus ist anzumerken, dass die Berechnung des Recalls den Zusammenhang zwischen den richtig extrahierten und den im Goldstandard annotierten Entitäten beschreibt. Precision wiederum bezieht sich auf die Anzahl der richtig vorgenommenen Extraktionen in Zusammenhang mit allen vorgenommenen Extraktionen (Vgl. Geduldig 2017: 27f.).

wird. Die in diesen Fällen getroffenen Entscheidungen sollten stets aus Gründen der Nachvollziehbarkeit halber dokumentiert werden (Vgl. Neumann 2010; Geduldig 2017).

3. Forschungsstand Qualifikationsentwicklungsforschung

Das Projekt Qualifikationsentwicklungsforschung beschäftigt sich seit Oktober 2015 mit der Aufbereitung, Annotation und Auswertung eines großen Korpus von mehr als hunderttausenden Stellenanzeigen. Das BIBB, das dieses Korpus zur Verfügung stellt, erhält einmal jährlich von der Bundesagentur für Arbeit (BA)[21] einen Auszug des Jobportals, der aus allen zum 15. Oktober des jeweiligen Jahres aktiven Stellenanzeigen besteht. Dieses Korpus von Stellenanzeigen ist aufgrund des Urheberrechts und Datenschutzstandards nur von einem Stand-Alone-PC ohne Internetzugang im BIBB aufrufbar. Im Kontext der Entwicklung dieser Arbeit war eine Nutzung dieses PCs nicht möglich. Stattdessen wurde ein vergleichbares Korpus mit Extraktionen aus Stellenanzeigen des Dienstleisters Textkernel[22] vom BIBB zur Verfügung gestellt, das von der Notwendigkeit der Nutzung innerhalb des BIBBs nicht betroffen war. Das Ziel des Projekts ist die Entwicklung eines Frameworks[23], das automatisiert relevante Informationen in Stellenanzeigen entdeckt und extrahiert. Relevante Informationen bei der Informationsextraktion sind abhängig von ihrer spezifischen Definition. Das bedeutet, dass es notwendig ist, eindeutig einzugrenzen, welche Informationen gesucht werden. In diesem Fall werden darunter die in Stellenanzeigen aufgeführten zu erfüllenden Bewerber:innenkompetenzen und im Beruf auszuführende Tätigkeiten in Zusammenhang mit den zu verwendeten Arbeitsmitteln verstanden. Durch die Durchführung dieser Extraktion ist es für das BIBB möglich, Analysen über die Entwicklung des Arbeitsmarktes anzustellen, bspw. wann welche Kompetenzen von Bewerber:innen gefordert wurden und in welchem Kontext dies zu beschreiben ist.

Bevor nun auf die einzelnen Verarbeitungsschritte, die im Zuge des Projektes entwickelt wurden und als Basis für die Durchführung des Bootstrapping-Ansatzes dienten, eingegangen wird, wird die Domäne der Stellenanzeigen zunächst näher betrachtet, um so die spezifische Struktur, die die Stellenanzeigen aufweisen, zu beschreiben und zu fragen, inwieweit diese bei der Extraktion der genannten relevanten Informationen eingesetzt werden kann.

[21] https://www.arbeitsagentur.de
[22] https://www.textkernel.com
[23] https://github.com/spinfo/quenfo

3.1 Struktur von Stellenanzeigen

Die Verarbeitung von Stellenanzeigen im Kontext der Informationsextraktion ist ein typisches Anwendungsgebiet und kann als relativ einfache Aufgabe angesehen werden. Die Verfügbarkeit dieser als nutzbares Korpus für die IE ist in der heutigen digitalen Welt stets in einer Fülle gewährleistet, da regelmäßig neue Anzeigen auf Webseiten von Unternehmen oder speziellen Jobangebotsplattformen wie Textkernel veröffentlicht werden. Unter dem Aspekt der Aktualität von Stellenanzeigen ist auch ersichtlich, warum sie sich besonders gut dazu eignen, mit ihnen Trendanalysen über das Arbeitsmarktgeschehen durchzuführen, wie sie nach Abschluss der Entwicklung des Frameworks geplant sind. Obwohl es sich bei Stellenanzeigen um un- bzw. semistrukturierte Daten handelt, die Informationen innerhalb einer Stellenanzeige also keiner einheitlicher Struktur folgen, wie sie bspw. in Datenbanken aufzufinden sind, unterliegt ihnen eine interne wiederkehrende Struktur, in der sich ähnliche, für eine Stellenanzeige stereotype Informationen vorfinden lassen, die sich für eine Extraktion und Überführung in eine strukturierte Form eignen. So lassen sich in (fast) jeder Stellenanzeige Informationen über den Arbeitgeber, die von den Bewerber:innen geforderten Kompetenzen, den Job sowie Formalia oder sonstige Informationen vorfinden. Diese verschiedenen inhaltlichen Klassen werden durch die kanonische Struktur untermalt: Fast jede Klasse ist als eigener Abschnitt innerhalb der Stellenanzeige getrennt. Die Abschnitte folgen einer (scheinbar) festgelegten Reihenfolge. Diese Struktur grenzt das Suchfeld der spezifischen Informationen auf die relevanten Abschnitte ein. Stellenanzeigen können sowohl in reiner Fließtextform als auch in Kombination mit Auflistungen verfasst werden. Die Abwechslung dieser beiden Formen kann demnach auch die Abgrenzung der einzelnen inhaltlichen Klassen verdeutlichen. In Abbildung 4 ist eine beispielhafte Stellenanzeige mit der Einteilung der jeweiligen Klassen aufgeführt. Deutlich wird hier, dass auch durch visuelle Mittel wie z.B. das Fettdrucken bestimmter Phrasen einzelne Abschnitte markiert werden können.

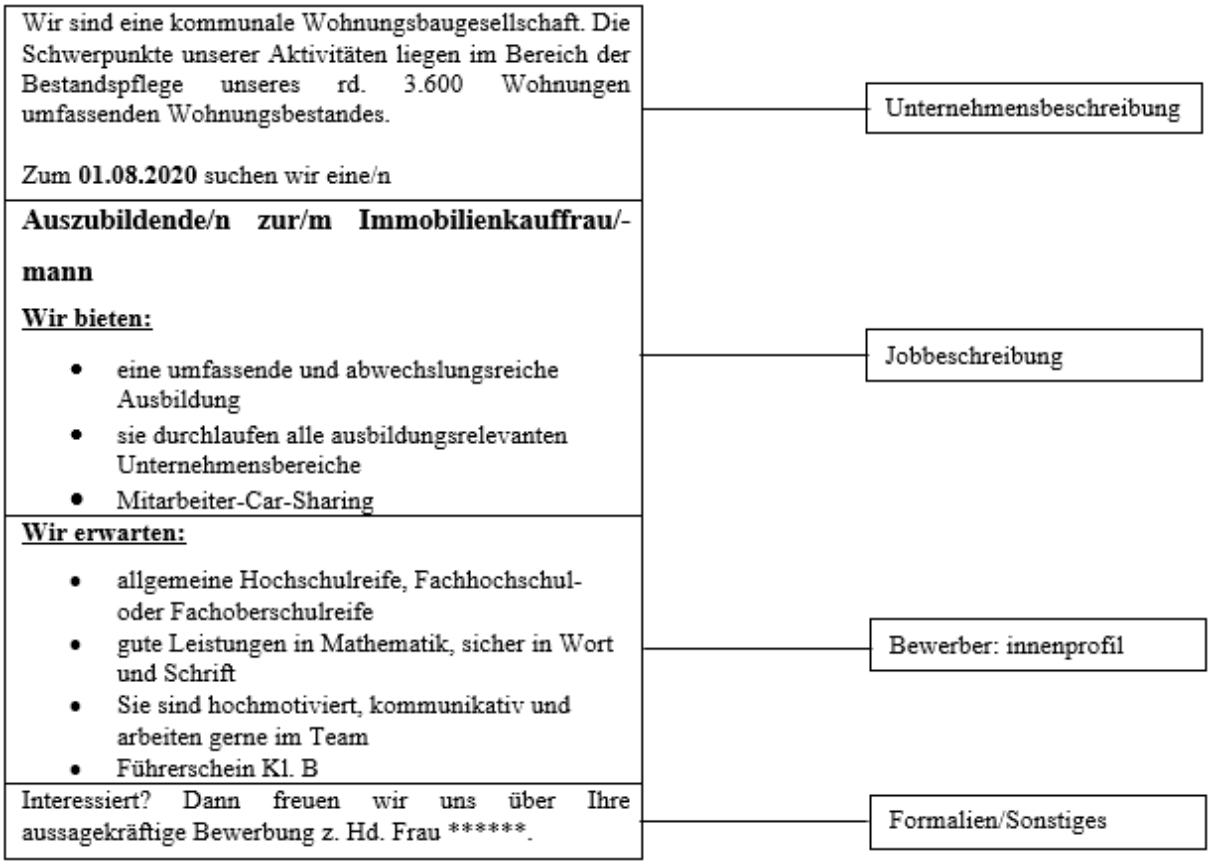


Abbildung 4: Strukturierung einer Stellenanzeige in inhaltliche Kategorien

Bezieht man nun die Aufgabe die Extraktion von relevanten Informationen aus Stellenanzeigen auf die Einteilung der Teilaufgaben innerhalb der IE wird ersichtlich, dass es sich hierbei lediglich um die Erkennung einzelner domänenspezifischer Entitäten handelt. Die Extraktion von Relationen oder Ereignissen wird gänzlich ausgesetzt. Dafür ist aber eine Vorverarbeitung der vorhandenen Daten in Sinne der IR notwendig. Zur Einteilung der Stellenanzeige wird die in der IR angewendete Textklassifikation genutzt. Die Umsetzung der Vorverarbeitung und Informationsextraktion innerhalb des Frameworks des Projekts Qualifikationsentwicklungsforschung wird im Folgenden aufgeführt.

3.2 Das Framework

Das zugrundeliegende Korpus, das vom BIBB zur Verfügung gestellt wurde, weist vielfältige Stellenanzeigen nach dem oben aufgeführten Schema auf. Für das Training der jeweiligen Formalismen des entwickelten Frameworks wurden diese in anonymisierter Form bereitgestellt. Diese Stellenanzeigen galt es in einem ersten Schritt in die verschiedenen inhaltlichen Kategorien zu unterteilen, um so eine Fokussierung auf die relevanten Passagen zu ermöglichen. Mithilfe eines Formalismus, der dem Bereich der Abschnittsklassifikation (*Zone*

Analysis) zuzuordnen ist, wurde eine entsprechende Klassifikation entwickelt, die mittels regulärer Ausdrücke die Texte in einzelnen Paragrafen unterteilt und diesen eine spezifische Kategorie zuordnet.[24] Die Zuordnung zu nur einer Klasse ist nicht immer möglich, da es Paragrafen gibt, die Informationen von mehr als einer inhaltlichen Klasse enthalten. Für die Extraktion ist die Mehrfachklassifikation eine zu meisternde Hürde, deshalb wurden bislang im weiteren Verlauf nur die Paragrafen betrachtet, die eindeutig einer Klasse zuzuordnen sind (Vgl. Hermes & Schandock 2016). Ausgehend von dem fertiggestellten Formalismus zur Klassifikation der Stellenanzeige wurde eine Vorstudie zur Entwicklung eines Extraktionsalgorithmus angesetzt, die untersucht, welche Methodik sich zur Extraktion von Bewerber:innenkompetenzen eignet. Verglichen wurden sowohl der Einsatz regulärer Ausdrücke in Form von manuell erstellten Extraktionsmustern, die mittels *Stringmatching*[25] Kompetenzen aufspüren, als auch ein Dependenzparser zur Mustererkennung im Text und in der syntaktischen Struktur. Verglichen wurde außerdem ein maschinelles Lernverfahren für einen Klassifikationsalgorithmus des *Naive Bayes*-Klassifikators. Den der Studie entnommenen Ergebnissen zufolge erzielt der Einsatz regulärer Ausdrücke die besten Ergebnisse (Vgl. Neumann 2015). Infolgedessen wurde auf Grundlage dieser Ergebnisse ein regelbasierter Formalismus zur Extraktion von Kompetenzen entwickelt, der die manuelle Erstellung von Extraktionsmustern beschreibt und auf Basis dieser die automatische Generierung neuer Muster in Umsetzung eines Bootstrapping-Ansatzes beinhaltet.[26] Für diesen Formalismus wurde ein implementierter *ClassifyUnitSplitter* genutzt, um die Stellenanzeigen in ihre einzelnen Paragrafen zu zerlegen und zu klassifizieren. Dieser berücksichtigte ergänzende Merkmale der Texte, wie den Zusammenhang einiger Abschnitte trotz räumlicher Distanz. Daraufhin wurden die klassifizierten Paragrafen selektiert, und die Paragrafen, die der Klasse 3 (Kompetenzen) zugeordnet wurden, in kleinere Einheiten (Sätze bzw. einzelne Listenelemente) zerlegt. Diese *ExtractionUnits* wurden wiederum nacheinander mit linguistischen Informationen wie Lemmata oder POS-Tag und einer Markierung der Satzgrenzen angereichert. In Abbildung 5 sind die einzelnen Vorverarbeitungsschritte aufgeführt.

[24] Andere Systeme nutzen auch statt regulärer Ausdrücke das sog. *Triggering*, um bestimmte Signalwörter, die ein Hinweis auf die unmittelbare Nähe der gewünschten Information liefern, aufzuspüren (Vgl. Jackson & Moulinier 2002: 75). Für die Extraktion aus Stellenanzeigen bietet sich diese Methode weniger an.

[25] Stringmatching beschreibt das Auffinden von Textsegmenten anhand eines vorgegebenen Suchmusters. In diesem Fall umfasst das Muster eine Anreihung regulärer Ausdrücke, die den Kontext vor und hinter der gesuchten Entität beschreiben (Vgl. Neumann 2015).

[26] Die detaillierte Beschreibung dieses Formalismus ist notwendig, weil diese Arbeit das Ziel hat, die Evaluation der in diesem Teil entwickelten Extraktionsmuster zu dokumentieren. Deshalb wird auf einzelne Prozessbestandteile des Formalismus genauer eingegangen.

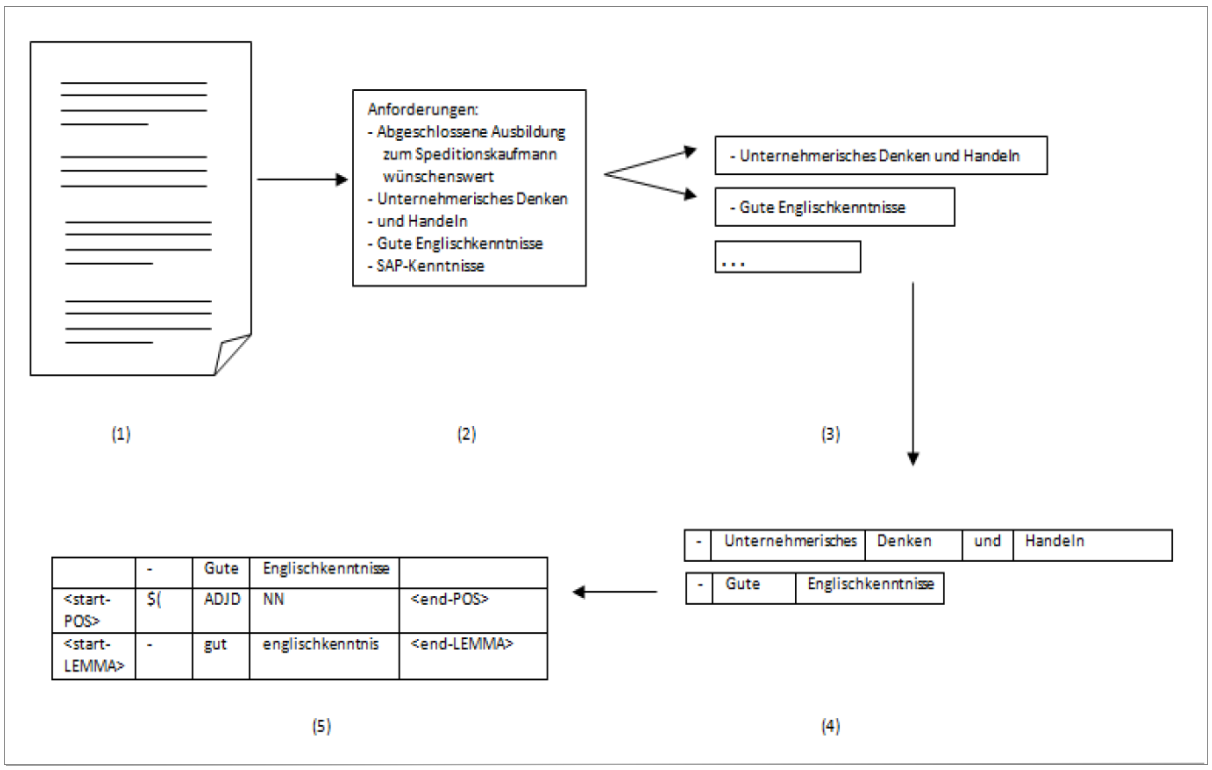


Abbildung 5: Schematische Darstellung der Vorverarbeitung eines Paragrafen. Aus der Stellenanzeige (1) wird zunächst ein Paragraf (2) selektiert, der der Klasse der Bewerber:innenprofile zugehörig ist. Dieser wird in Sätze (3) und in Tokens (4) zerlegt, die mit linguistischen Informationen wie POS-Tag oder Lemmata angereichert werden (5). (Quelle: Geduldig 2017)

In einem nächsten Schritt wurde ein Regelformalismus entwickelt, der Extraktionsregeln in Form linguistischer Kontexte beschreibt. Diese Schablonen beschreiben eine Folge von geforderten Tokens und das zu extrahierende Token über einen Index. Indem die Muster auf die in Sätze zerlegten Paragrafen angewendet wurden, wurde ein Set aus Entitäten ermittelt, das als initiales Startset für einen darauf aufbauenden Bootstrapping-Algorithmus genutzt werden konnte. Via Stringmatching wurden neue Kontexte der Seed-Kompetenzen aufgesucht, die zu neuen Extraktionsmustern generalisiert wurden. Während der nächsten Iteration werden diese

neuen Extraktionsmuster zusätzlich zu den bereits bestehenden genutzt, um neue Entitäten zu ermitteln. Der Kreislauf lässt sich wie in Abbildung 6 darstellen:

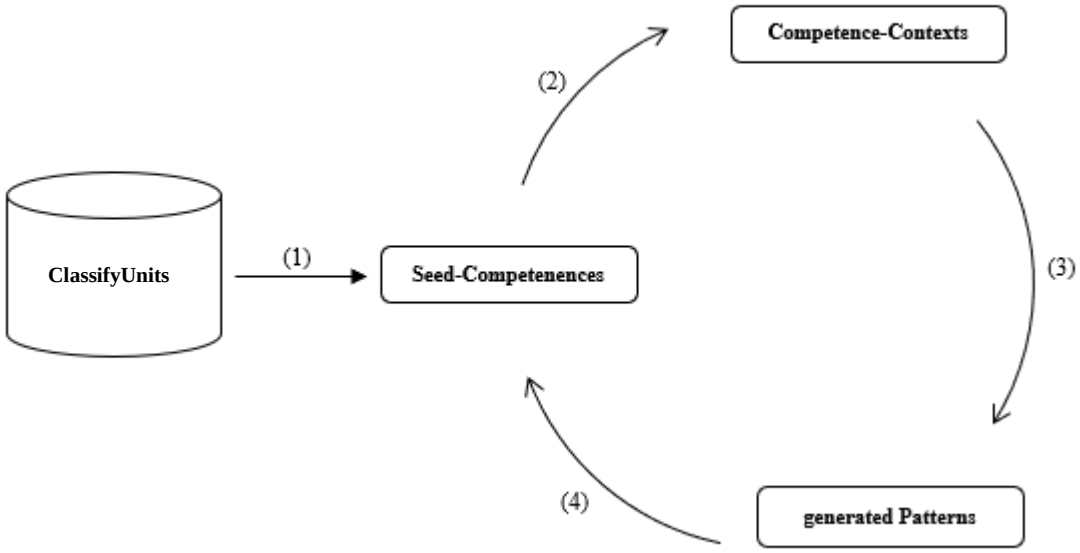


Abbildung 6: Schematische Darstellung des Bootstrapping Ansatzes zur Kompetenzextraktion nach Geduldig 2017. (1) Regelbasierte Extraktion der Seed-Kompetenzen. (2) Stringmatching der Seeds innerhalb der ClassifyUnits. (3) Automatisierte Mustergenerierung. (4) Regelbasierte Extraktion mit neuen Mustern.

Bei der Verarbeitung natürlichsprachlicher Texte in der Informationsextraktion besteht immer die Herausforderung, den Text mit semantischem Inhalt zu füllen. Der Computer allein ist nicht in der Lage die Bedeutungsunterschiede von bspw. zwei polysemen oder homonymen Wörtern zu erkennen.[27] So können schnell ambige Entitäten extrahiert werden, die, wenn man sie in ihrem Kontext betrachtet, nicht für die gesuchte Entität stehen können. Andererseits können auch Entitäten aufgrund verschiedenster Formulierungen bei der Extraktion übersehen werden. Die Entwicklung einer Word Sense Disambiguierung auf die Domäne der Stellenanzeige in Bezug auf die Extraktion von Bewerber:innenkompetenzen ermöglicht es computergestützt zu entscheiden, welche Bedeutung eines Wortes in einem spezifischen Kontext aktiviert wird, und beschreibt den aktuellen Forschungsstand des Projekts (Vgl. Binnewitt 2020). Ausgehend von dem nun beschriebenen Framework wird im weiteren Verlauf die Entwicklung eines Formalismus zur Evaluation der Extraktionsmuster beschrieben. Es ist zu vermerken, dass die

[27] Das Konzept der Polysemie beschreibt, dass alle Bedeutungen eines Ausdrucks auf eine gemeinsame Kernbedeutung zurückgeführt werden können, also, dass ein sprachliches Zeichen für viele verschiedene Bedeutungsinhalte bzw. Begriffe steht. Im Gegensatz bezeichnet das Konzept der Homonymie die Zurückführung auf keinen gemeinsamen Bedeutungsursprung, derselbe sprachliche Ausdruck steht für verschiedene Begriffe (Vgl. Schwarz & Chur 2007: 56).

Entwicklung des Bootstrapping-Ansatzes als Grundlage diente, ohne die WSD mit einzubeziehen. Diese Entscheidung wurde aus dem Grund getroffen, dass es vorerst nicht notwendig ist, mögliche ambige Entitäten für die Evaluation bzw. Verbesserung der genutzten Extraktionsmuster miteinzubeziehen.

4. Evaluation von Extraktionsmustern

Das von Geduldig 2017 entwickelte Bootstrapping-Verfahren nutzt einen iterativen Prozess, um sowohl Kompetenzen wie auch aufgeführte Arbeitsmittel aus Stellenanzeigen zu extrahieren. Für die Extraktion werden manuell erstellte wie auch automatisch generierte Extraktionsmuster genutzt. Durch die Anwendung dieser Kombination erreicht der Formalismus bei einer Kontextgröße von links zwei und rechts drei Tokens[28] einen F-Score von ca. 84%. Precision liegt in diesem Fall bei ca. 92% und Recall bei ca. 77%. Bei der Betrachtung dieser Ergebnisse ist einsehbar, welche Stärken der Algorithmus aufweist: Es gelingt, viele Kompetenzen richtig zu labeln im Vergleich zu der Gesamtzahl an getätigten Extraktionen. Dabei gelingt es nicht, den Recall, also die Vollständigkeit der Extraktionen, gleichermaßen stark zu besetzen. Das Ziel der Überarbeitung des Ansatzes ist also demnach, den Recall zu verbessern, ohne bei der Precision einsparen zu müssen.

Der im Vergleich zur Precision niedrige Recall kann verschiedene Ursachen haben. Zum einen kann bei der Anwendung eines Bootstrapping-Verfahrens auf ein großes Korpus, wie das der Stellenanzeigen, das bereits im Kapitel 2.2 aufgeführte Semantic Drifting auftreten, bei dem durch den iterativen Prozess einzelne Fehler häufig wiederholt werden. Durch falsche Extraktionen fließen falsche bzw. schlechte Muster in den Prozess mit ein, die wiederum zu neuen Falschextraktionen führen. Zum anderen können die als Ausgangspunkt genutzten Seeds sehr spezifische Entitäten extrahieren bzw. Muster generieren, so dass andere, unspezifische übersehen werden können. Andererseits können bereits bei der Klassifikation der Paragrafen Fehler auftreten, wenn diese die Abschnitte nicht der richtigen Klasse zuordnet, oder bei der Zerlegung der Paragrafen in einzelne Sätze, wenn bspw. Satzgrenzen nicht richtig erkannt werden. Auch erschweren die bereits beschriebenen möglichen ambigen Wörter eine eindeutig richtige Extraktion. Diese Probleme haben dann wiederum auch Auswirkungen auf die weitere

[28] Bei der automatischen Generierung der Extraktionsmuster muss angegeben werden, wie viele Token links und rechts neben der eigentlichen zu extrahierenden Entität vorzufinden sein dürfen. Bei der Evaluation wurden verschiedene Kontextgrößen erprobt, jedoch erzielten die Kontexte mit links zwei und rechts drei Tokens die besten Ergebnisse (Vgl. Geduldig 2017: 36).

Verarbeitung der Daten. Zur Verbesserung des Recalls werden nun Evaluationsschritte zwischengeschaltet, die sowohl die Extraktionsmuster als auch die extrahierten Entitäten bewerten und bei „schlechter“ Performance aussortieren, so dass diese für die nächste Iteration nicht genutzt werden.

4.1 Manuelle Inspektion der Extraktionsmuster

Bei der manuellen Inspektion der zur Verfügung gestellten Extraktionsmuster, die bei der Extraktion von Kompetenzen aus Stellenanzeigen genutzt werden, wurde zuerst die Funktionalität der Muster betrachtet. Dies war ein notwendiger Schritt, um beurteilen zu können, welche Muster besonders viele bzw. wenige Kompetenzen extrahieren können. Zur Auswertung dieser Inspektion wurde ein Korpus von rund 48.000 Extraktionen der Klasse 3 (Kompetenzen), die aus Stellenanzeigen eines Monats der Jobangebotsplattform Textkernel mithilfe des Bootstrapping-Formalismus extrahiert wurden, und eine Datei mit Extraktionsmustern verwendet. Eine zusätzliche nützliche Information innerhalb des Korpus war die Angabe des Extraktionsmusters für die jeweilige Extraktion. Die Funktionalität der für die Extraktion genutzten Muster wurde von folgenden Hypothesen ausgehend betrachtet:

1) Spezifische Muster extrahieren weniger Kompetenzen als Unspezifische.
2) Unspezifische Muster extrahieren mehr fehlerhafte Kompetenzen als Spezifische.

Spezifische Muster beziehen sich auf diejenigen mit vielen festgelegten Kontexttoken, während die Kontexte von den unspezifischen ausschließlich durch POS-Tags beschrieben werden. Um die genannten Hypothesen nun zu überprüfen, wurden als erstes alle Extraktionsmuster gesichert, die bei der Extraktion verwendet wurden und die Anzahl der durchgeführten Extraktionen gezählt. Im Anschluss wurden diese dann mit einer Liste bereits validierter Kompetenzen, die ebenfalls vom BIBB zur Verfügung gestellt wurde, verglichen.[29] Bei der Auswertung der Ergebnisse fällt auf, dass die Beantwortung der aufgestellten Hypothesen nicht pauschal durchgeführt werden kann. So extrahiert bspw. das unspezifische Muster [TRUNC + Konj|, + COMP][30] die meisten Kompetenzen sowie die meisten Extraktionen, die bereits in der Liste mit validierten Kompetenzen vorhanden sind. In Tabelle 1 sind die Muster mit den besten Ergebnissen aufgeführt.

[29] Die entsprechende ausführbare Klasse befindet sich im mitgelieferten Software-Projekt. (src/main/java/de/uni_koeln/spinfo/preinspection_pattern/ListenQuantity.java)
[30] Die Bezeichnung ist eine vereinfachte Form, sie bezieht sich nur auf den Namen des Musters. Sie kann aber als eindeutige Identifizierung genutzt werden.

	Extraktionsmuster	Anzahl an Extraktionen[31]		
		Gesamt	TP	FP
1	TRUNC + Konj|, + COMP	10521	6071	0
2	Adj|Nom|TRUNC + kenntnis|erfahrung	4204	578	2
3	ausbildung|abschluss|berufserfahrung|... + zu|als|in... + Nom	3763	4	0
4	Adj|NN|TRUNC + -vermögen|bereitschaft|-kompetenz	2679	302	0
5	Adj|NN + kenntnis	2573	16	44
6	KNOWLEDGE + Art|Präp + Nom|Adj + Nom|Zahl	2561	14	0
7	COMP + Konj + Nom|Adj + Nom	1375	174	0
8	Adj|Nom|TRUNC + kenntnis|erfahrung|AJD + Ausbildung	1364	787	0
9	ausbildung|abschluss|berufserfahrung|... + zu|als|in... + Nom|TRUNC + Konj + Nom	1052	0	0
10	ausbildung|abschluss|berufserfahrung|... + zu|als|in... + Nom|Adj + Nom	906	2	0

Tabelle 1: Ergebnisse der manuellen Inspektion der Extraktionsmuster

Betrachtet man die in Tabelle 1 aufgeführten Muster, trifft die Bezeichnung unspezifisch nicht auf alle zu. Trotzdem ist die Differenz zwischen dem besten Muster zu den anderen sehr hoch, auch die Anzahl der bereits bekannten Kompetenzen, die mit diesem Muster extrahiert wurden, ist nicht mit der Funktionalität der anderen zu vergleichen. Die Aussage, dass spezifische Muster weniger extrahieren als unspezifische, kann also unter Vorbehalt als zutreffend gewertet werden. Anders sieht es mit der zweiten Hypothese aus. Fast alle Extraktionen kommen nicht in den bekannten Extraktionsfehlern vor. Lediglich bei drei Mustern (in der Tabelle sind zwei davon aufgeführt) konnten Extraktionsfehler vermerkt werden. Im Vergleich zu den gesamten Extraktionen nehmen diese aber keinen nennenswerten Anteil an. Betrachtet man trotzdem die drei Muster, die Extraktionsfehler durchführen, fällt auf, dass es sich hierbei um unspezifische Muster handelt. Diese Muster könnten evtl. im Zuge der Evaluation aus der Sammlung der zu verwendeten Muster herausgenommen werden.

Zusammenfassend zeigen die Ergebnisse der manuellen Überprüfung eine gute Funktionalität der eingesetzten Extraktionsmuster. Mit den Mustern wurden viele Kompetenz-Kandidaten

[31] Um TP und FP zu ermitteln, wurden alle Extraktionen eines Musters mit den bereitgestellten Listen mit validierten Kompetenzen sowie bekannten Extraktionsfehlern verglichen.

gefunden, die bislang noch nicht bei den validierten Kompetenzen aufgelistet sind. Auf die Anwendung des Bootstrapping-Verfahrens lassen sich jedoch noch keine gewichteten Aussagen treffen, da vorerst nur feststehende Extraktionen und Extraktionsmuster betrachtet wurden, nicht aber wie sich die Extraktionsmuster in einem wiederholenden Workflow verhalten. In einem nächsten Schritt wurde ein Formalismus entwickelt, der die Extraktionsmuster auf Basis ihrer Funktionalität bewertet und entsprechende Werte auch den Extraktionen zuordnet, um diese weiter selektieren zu können und nur die besten für die nächste Iteration zu nutzen.

4.2 Bewertung der Extraktionsmuster und Extraktionen

Ausgehend von der Aufgabenstellung sollen nun Evaluationsschritte in den bereits bestehenden Workflow integriert werden, die sowohl die verwendeten Extraktionsmuster als auch die entstandenen Extraktionen aufgrund ihrer Funktionalität innerhalb des Ablaufs bewerten und ggf. vorzeitig aussortieren. Zur Entwicklung dieser Schritte wurde das *Snowball*-System, das bereits als Grundlage für die Entwicklung des Bootstrapping-Ansatzes genutzt wurde, verwendet, um den dort verwendeten *Confidence*-Wert einzuführen. Der Confidence bezieht sich auf die *selectivity* (Selektivität) und die *coverage* (Reichweite) und beschreibt einen Wert, der angibt, wie sehr dem Extraktionsmuster bzw. der Extraktion zu trauen ist.[32] Die Güte eines Extraktionsmusters wird anhand der Anzahl der damit aufdeckbaren bereits bekannten Kompetenzen ermittelt, die Güte einer Extraktion berechnet sich anhand der Anzahl und Güte der produzierenden Muster (Vgl. Geduldig 2017: 37). Ausgehend vom *Snowball*-System kann die Güte eines Extraktionsmusters folgendermaßen berechnet werden:

$$Conf(P) = \frac{P.pos}{(P.pos + P.neg)} \tag{4.1}$$

wobei *P.pos* die Anzahl der bereits bekannten Entitäten beschreibt, die mit dem Muster aufgedeckt wurden, und *P.neg* die Negativbeispiele, also die Entitäten, die als angefragter Type[33] extrahiert wurden, aber eigentlich nicht dem Type zugehörig sind. Um die Positiv- wie auch die Negativbeispiele verwenden zu können, werden Listen benötigt, in denen validierte Extraktionen wie auch bekannte Extraktionsfehler aufgeführt werden, so dass die Extraktionen

[32] Muster sollen also einerseits selektiv sein, also möglichst wenig bis (im Idealfall) keine falschen Entitäten extrahieren, und andererseits möglichst viele neue Entitäten identifizieren können (Vgl. Agichtein & Gravano 2000)

[33] Type bezeichnet in diesem Kontext die Zugehörigkeit einer Entität zu der Klasse der Kompetenzen oder Arbeitsmittel.

mit diesen Listen verglichen werden können.[34] Die Güte einer Extraktion wird wiederum nach Formel 4.2 berechnet.

$$Conf(s) = 1 - \Pi(1 - Conf(P_i)) \tag{4.2}$$

Der Wertebereich der beiden Berechnungen liegt im Intervall [0,1] mit dem Optimum bei 1.0 (100%). Demnach kann eine Extraktion als weniger gut betrachtet werden, wenn sie von vielen Mustern extrahiert wurde, die einen niedrigen Confidence-Wert haben. In einem nächsten Schritt werden dann die Muster und Extraktionen aussortiert, dessen Güte nicht höher ein festgelegter Wert ist.[35] Bei jeder Iteration werden neue Confidence-Werte für die verwendeten Muster und Extraktionen erstellt, so dass vorhandene Werte überschrieben werden. Das Ziel, das durch die Einführung der Evaluationsschritte erreicht werden soll, ist die Reduzierung fehlerhafter Extraktionen durch frühzeitige Aussortierung. Die Annahme, die dahintersteckt, ist die, dass Extraktionen mit einem niedrigen Confidence-Wert mit geringerer Wahrscheinlichkeit tatsächlich eine gesuchte Entität abbilden.

4.3 Workflow

Die bereits verwendeten Formeln zur Berechnung der Confidence-Werte der Extraktionsmuster und resultierenden Extraktionen wurden nun innerhalb eines Formalismus in den bereits bestehenden Bootstrapping-Formalismus integriert. In Abbildung 7 sind die zusätzlichen Schritte schematisch dargestellt. Während jeder Iteration werden nach der Extraktion zuerst die verwendeten Muster herausgesucht, und für jedes Muster wird eine Liste mit den getätigten Extraktionen erstellt. Diese Liste wird mit bereits validierten bzw. bekannten Entitäten und Extraktionsfehlern verglichen, um so die für die Berechnung notwendigen Positiv- und Negativbeispiele der Muster zu erkennen. Ausgehend davon werden die Confidence-Werte aller Muster berechnet, um in einem weiteren Schritt den Wert für die Extraktionen zu ermitteln. Durch die Festlegung eines Minimums werden alle Extraktionen, die unter diesem Minimum liegen, aussortiert. Alle anderen werden für die nächste Iteration wiederverwendet.

[34] Schwierigkeiten bei der Berechnung des Confidence-Wertes innerhalb eines Bootstrapping-Verfahrens werden in Kapitel 4.5 skizziert.
[35] Agichtein und Gravano (2000) sortieren in ihrem System alle Muster und Extraktionen mit einem Conf ≥ 0.8 aus, da mit diesem Wert die besten Evaluationsergebnisse (Precision und Recall) erreicht werden konnten.

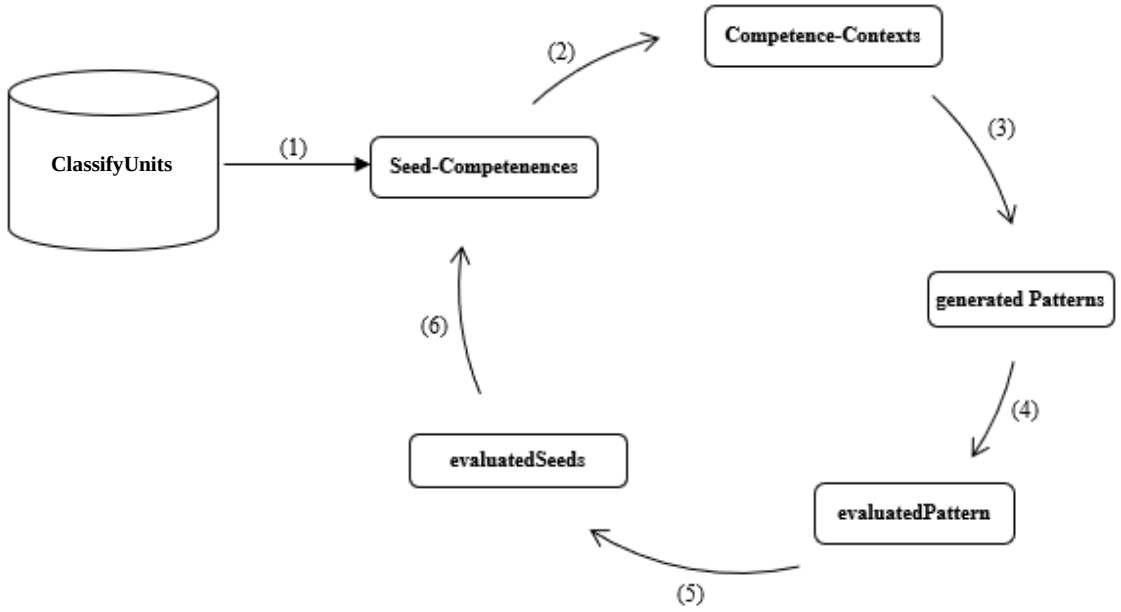


Abbildung 7: Schematische Darstellung des Bootstrapping Ansatzes mit zusätzlichen Schritten zur Evaluation der Extraktionsmuster und Extraktionen. Schritte (1) bis (3) können aus Abbildung 6 übernommen werden. (4) Berechnung des Confidence-Werts der genutzten Extraktionsmuster. (5) Berechnung des Confidence-Werts der Extraktionen. (6) Aussortieren der Extraktionen anhand eines festgelegten Confidence-Werts und Hinzufügen der restlichen Extraktionen zum Initialset.

4.4 Evaluation

Zur Evaluation des Bootstrapping-Ansatzes wurden die bereits in Kapitel 2.3 vorgestellten Evaluationsmaße (Precision, Recall und F-Maß) für IE-Systeme auf das Projekt angewendet. Das Projekt beinhaltete schon einen vollständigen Evaluationsworkflow[36], der um die Komponenten der Berechnung der Confidence-Werte und die daraus resultierende Selektion erweitert wurde. Als Trainingsdaten wurde ein Testkorpus aus *ClassifyUnits* (Paragrafen) genutzt, die zuvor der Klasse der Bewerber:innenkompetenzen zugeordnet und daraufhin in *ExtractionUnits* unterteilt wurden.[37] Außerdem wurden vom BIBB zur Verfügung gestellte Extraktionsmuster und Listen mit validierten Kompetenzen und bekannten Extraktionsfehlern in den bestehenden Workflow integriert. Die Ausgangsbedingungen für die Einführung der Evaluationsschritte in das Bootstrap-Verfahren sind eine Precision von ca. 92% und ein Recall von ca. 77%. Die für die automatische Musterbildung verwendete Kontextgröße beläuft sich

[36] Die entsprechende ausführbare Klasse befindet sich im mitgelieferten Softwareprojekt. (src/test/information_extraction/evaluation/EvaluateBootstrappingExtraction.java)
[37] Für die Ausführung des Workflows zur Extraktion der Bewerber:innenkompetenzen steht ein anonymisiertes Trainingskorpus zur Verfügung. (src/test/resources/classification/input/trainingData.csv)

bei diesen Ergebnissen auf links zwei und rechts drei Tokens. Somit wurde insgesamt ein F-Score von ca. 84% erreicht (Vgl. Geduldig 2017: 36).

In Tabelle 2 und 3 sind die Evaluationsergebnisse aufgelistet, die mit der Einführung der Evaluationsschritte in den bestehenden Algorithmus erreicht wurden. Hierbei ist anzumerken, dass ausgehend von den bisherigen Ergebnissen nur mit der Kontextgröße links zwei und rechts drei Token evaluiert wurde. Es wurden mehrere Durchläufe mit verschiedenen Minimalwerten der Confidence-Werte durchgeführt. Zudem wurde das System mit und ohne Einsatz der Beschränkung der Confidence-Werte der Extraktionsmuster erprobt. Die besten Ergebnisse sind jeweils fett hervorgehoben.

Confidence-Werte		**Evaluationsmaße**			**Iterationen**
Conf(s)	Conf(P)	Precision	Recall	F-Score	
0.1	0.1	0.744	0.886	0.809	5
0.5	0.6	0.749	0.886	0.812	5
1.0	0.9	0.766	0.876	0.817	6
0.9	1.0	0.747	**0.894**	0.814	5
1.0	1.0	**0.78**	0.875	**0.825**	5

Tabelle 2: Evaluationsergebnisse des Bootstrapping-Verfahrens mit Einbindung der Confidence-Werte für die Extraktionsmuster und Extraktionen[38]

Confidence der Extraktionen (Conf(s))	**Evaluationsmaße**			**Iterationen**
	Precision	Recall	F-Score	
0.1	0.744	**0.884**	0.808	5
0.2	0.744	**0.884**	0.808	5
0.3	0.744	**0.884**	0.808	5
0.4	0.744	**0.884**	0.808	5
0.5	0.744	**0.884**	0.808	5
0.6	0.744	**0.884**	0.808	5
0.7	0.744	**0.884**	0.808	5
0.8	0.744	**0.884**	0.808	5

[38] Es ist anzumerken, dass die Tabelle nur einen Teilausschnitt der Ergebnisse wiedergibt, die insgesamt ermittelt wurden. Dies schließt daher, dass bei dem Setzen der Confidence-Werte im Bereich [0.1 – 0.5, 0.6 - 0.9] die gleichen Ergebnisse resultieren. In der Tabelle sind also jeweils die Veränderungen der Werte aufgezeigt. Alle Evaluationsergebnisse befinden sich auf der beigefügten CD. (Ergebnisse/Evaluationsergebnisse.pdf)

0.9	0.744	**0.884**	0.808	5
1.0	**0.765**	0.876	**0.816**	7

Tabelle 3: Evaluationsergebnisse des Bootstrapping-Verfahrens mit Einbindung des Confidence-Werts der Extraktionen

Mit einer Beschränkung sowohl der Extraktionsmuster als auch der Extraktionen, die jeweils nur mit einem Confidence-Wert von 1.0 (also einer 100-prozentigen Wahrscheinlichkeit, dass sie richtige Entitäten extrahieren und die Extraktionen der gesuchten Klasse zuzuordnen sind) weiterverwendet werden, konnte der beste F-Score von ca. 82,5% und eine Precision von ca. 78% erreicht werden. Das ist im Vergleich zu den Ausgangsbedingungen ein schlechteres Gesamtergebnis. Betrachtet man jedoch den Recall, ist eine deutliche Verbesserung zu vermerken. Insgesamt nähern sich Precision und Recall bei einer zunehmenden Einschränkung der Extraktionsmuster und Extraktionen immer weiter an. Vergleicht man diese Ergebnisse nun mit denen aus Tabelle 3, die zeigen, wie sich das System bei der unären Angabe der Confidence-Werte der Extraktionen verhält, fallen die Ergebnisse bei diesem Experiment deutlich schlechter aus als bei der Angabe beider Werte. So zeigt sich erst eine Verbesserung bei einer 100-prozentigen Wahrscheinlichkeit. Daraus lässt sich vorerst schließen, dass die beim Testen entstandenen Extraktionen keinen Confidence-Wert unter 0.9 zugewiesen bekommen haben und dementsprechend auch die Extraktionsmuster vergleichbare hohe Werte erreichen. Zusammenfassend zeigen die Ergebnisse eine deutliche Verbesserung des Recalls bei gleichzeitig leichter Verschlechterung der Precision. Der F-Score ist mit seiner 1,5-prozentigen Differenz aber vergleichbar gut.

4.5 Einschränkungen

Die guten Ergebnisse, die beim Einsatz der Confidence-Werte der Extraktionsmuster und Extraktionen erreicht wurden, lassen sich auf die umfangreichen Listen mit bereits validierten Kompetenzen und Extraktionsfehlern zurückführen, die zur Ermittlung der Positiv- und Negativbeispiele der Extraktionsmuster genutzt werden. Die Liste mit den validierten Entitäten enthält zu Beginn bereits etwa 28.000 Kompetenzen und die der Extraktionsfehler um die 280. Tests mit kleiner angesetzten Listen zeigen, dass das System deutlich schlechtere Ergebnisse erzielt. Besonders der Recall fällt tief ab, vergleichbar mit den ursprünglichen Ergebnissen. Das System ist dementsprechend abhängig von umfangreichen Listen als Grundlage. Da das System aber bereits mehrfach erprobt und bzgl. des Disambiguierungsproblem weiterentwickelt wurde,

sind solche Listen vorhanden.[39] Ein weiterer Test, bei dem weder die Liste mit validierten Entitäten noch die mit bekannten Extraktionsfehlern übergeben werden, sondern das initiale Startset zur Bildung beider genutzt wird, erwiesen sich als schwierig, da die Berechnung der Confidence-Werte und die darauffolgende Selektion erst beginnen darf, wenn die Listen bereits gefüllt wurden. Bei der Berechnung direkt beim ersten Durchlauf endet das System nach einer Iteration und zeigt dementsprechend schlechte Ergebnisse. Fügt man eine Bedingung hinzu, dass die Berechnung erst beim zweiten Durchlauf starten soll, schafft das System mehrere Iterationen, die Ergebnisse verbessern sich aber keineswegs. Im Vergleich dazu verbessert der Einsatz von nur einer vorgefüllten Liste die Ergebnisse in interessanter Weise: Beim Einsatz der Liste mit den validierten Entitäten sinkt die Precision rapide, beim Einsatz der Liste mit bekannten Extraktionsfehlern der Recall. Um nun aber sowohl bei der Precision als auch beim Recall vergleichbar gute Ergebnisse zu erzielen, ist der Einsatz beider Listen notwendig. Die in Tabelle 2 aufgeführten Ergebnisse zeigen außerdem zwei Abhängigkeiten: Je höher das Minimum des Confidence-Werts (sowohl bei den Extraktionsmustern als auch bei den daraus resultierenden Extraktionen) gesetzt ist, desto weniger Entitäten werden gefunden und desto genauer sind die Ergebnisse, d.h. desto weniger FPs werden extrahiert.[40] Die Angabe des Minimums muss sich dementsprechend an die genutzten Listen anpassen. Werden nun Listen mit deutlich weniger validierten Entitäten und bekannten Extraktionsmustern genutzt, so werden weniger TPs ermittelt und so muss auch das Minimum der Confidence-Werte niedriger gesetzt werden.

Auch wenn der Einsatz der Evaluationsschritte einigen Beschränkungen unterliegt, ist die Vermeidung bzw. Reduzierung von vielfacher Wiederholung „schlechter“ Extraktionen und daraus resultierenden „schlechten“ automatisch generierten Extraktionsmustern durch die Berechnung des Confidence-Werts gewährleistet. Bei einer manuellen Einsicht der Extraktionen, die beim besten Durchlauf (s. Tabelle 2) entstanden sind, fällt bloß ein Fehler besonders auf. Der Ausdruck Erfahrung (z.B. in „Erfahrung in Drehtechnik“) wurde aus mehreren Sätzen extrahiert, ohne den jeweiligen Kontext miteinzubeziehen. Die anderen FPs lassen sich meist aber nur auf einen Satz zurückführen, eine iterative Fehlerwiederholung ist hier also nicht der Fall. Um solche Fehler in Zukunft zu vermeiden, ist der Einbezug des

[39] Die hier verwendeten Listen wurden ebenfalls vom BIBB zur Verfügung gestellt.
[40] Die Anzahl der TPs, FPs und FNs sind hier nicht aufgeführt, können aber in der entsprechenden Datei auf der beigelegten CD überprüft werden. (Ergebnisse/Evaluationsergebnisse.pdf)

Kontextes eines Wortes von großer Relevanz. So können Extraktionen auf ihre Vollständigkeit hin überprüft und unvollständige Entitäten nicht als korrekte Extraktion gesichert werden.[41]

5. Fazit und Ausblick

Das Ziel der Arbeit war die Weiterentwicklung des bestehenden Frameworks zur Extraktion domänenspezifischer Informationen auf Basis eines Bootstrapping-Ansatzes, um während des Prozesses auftretende Fehler, wie das Semantic Drifting, zu reduzieren und den verwendeten Formalismus zu verbessern. Das Framework wird innerhalb des Kooperationsprojekts Qualifikationsentwicklungsforschung der IDH mit dem BIBB zur Extraktion von in Stellenanzeigen aufgeführten Bewerber:innenkompetenzen und Arbeitsmitteln eingesetzt.

Um einen theoretischen Rahmen zu schaffen, wurde in Kapitel 2 zunächst der Bereich der maschinellen Sprachverarbeitung beleuchtet, und wichtige Bereiche wurden aufgeführt. Hierbei wurden Aufgaben der Informationsextraktion als Teilbereich des Textminings spezifiziert und besonders auf semiüberwachte Lernverfahren, wie den Bootstrapping-Ansatz, eingegangen. Einblicke in vergangene und aktuelle Forschungsstände, die Bootstrapping-Formalismen auf verschiedene Domänen angewendet haben, wurden hier kurz skizziert. Außerdem wurden die Evaluationsmaße der IE vorgestellt. In Kapitel 3 wurde der aktuelle Forschungsstand des Projekts Qualifikationsentwicklungsforschung und die Struktur von Stellenanzeigen dargelegt. Anhand dieses Rahmens wurde in Kapitel 4 der Entwurf der Evaluationsschritte, die in den Workflow integriert wurden, vorgestellt. Für die Weiterentwicklung wurden Evaluationsschritte in den iterativen Ablauf eingefügt, die sowohl die Extraktionsmuster als auch die daraus resultierenden Extraktionen anhand ihrer Funktionalität innerhalb des iterativen Prozesses bewerten und auf Basis dieser Bewertung eine Selektion durchführen. Extraktionsmuster wurden demnach nach der Anzahl der bereits bekannten Entitäten, die mit dem jeweiligen Muster extrahiert wurden, bewertet, Extraktionen dagegen nach der Güte und Anzahl der Muster, die sie extrahiert haben. Als Referenz für die Bewertung wurden die vom BIBB zur Verfügung gestellten Listen mit validierten Entitäten und bekannten Extraktionsfehlern genutzt. Für die Selektion wurde ein Grenzwert festgelegt: Extraktionsmuster mit einem Confidence-Wert unter 1.0 und Extraktionen mit einem

[41] Hier ist anzumerken, dass der Algorithmus zwar bereits Entitäten aus mehr als einem Wort extrahieren kann, um jedoch neue Kontexte zu entdecken, werden diese häufig in ihre Einzelteile zerlegt. Es ist aber aufgefallen, dass einzelne Teile von komplexeren Entitäten trotzdem als korrekte Extraktion gesichert werden, obwohl ihnen die relevanten Sequenzen fehlen (z.B. wird „Berufserfahrung“ ohne Spezifikation des Bereichs als Kompetenz extrahiert).

Confidence-Wert unter 0.9 entfallen für die nächste Iteration. Die in den Testläufen verwendeten Grenzwerte basieren auf den erzielten Ergebnissen. So konnte keinem Muster ein Confidence-Wert unter 0.6 und keiner Extraktion ein Wert unter 0.9 zugewiesen werden. Mit den verwendeten Grenzwerten konnte der Recall auf bis zu 89% erhöht werden. Leider ist ein deutlicher Rückgang der Precision auf maximal 78% zu vermerken. Das harmonische Mittel zwischen beiden (F-Score) ist aber mit seinem besten Ergebnis von 82,5% vergleichbar mit den ursprünglichen Evaluationsergebnissen. Die in Kapitel 4.5 aufgeführten Einschränkungen zeigen aber, dass die Berechnung der Confidence-Werte nicht optimal gestaltet ist. Aufgrund der Abhängigkeit von umfangreichen Listen mit validierten Kompetenzen und Extraktionsfehlern, die im Laufe des Projekts erstellt wurden und verfügbar sind, ist die Einbindung in einen Bootstrapping-Formalismus generell schwieriger gestaltet, denn der Vorteil des Einsatzes von semiüberwachten Lernverfahren sollte die nicht vorhandene Notwendigkeit großer vorannotierter Trainingsdaten sein. Die Tests mit keinen zugrundeliegenden umfangreichen Listen zeigten dementsprechend weniger gute Ergebnisse. Trotzdem konnte mithilfe der zur Verfügung gestellten Ressourcen eine Verbesserung hinsichtlich der Fehlerhaftigkeit der Ergebnisse erzielt werden. Eine andere Möglichkeit zur Selektion der Extraktionen und Extraktionsmuster könnte sein, eine zweite Ebene zur Auswahl der fünf besten einzufügen, die für die nächste Iteration genutzt werden (Vgl. Riloff 1999; Sun 2009). Aber auch hier stellt sich die Problematik der Grundlage zur Berechnung der Confidence-Werte.

Literaturverzeichnis

Agichtein, Eugene & Gravano, Luis (2000): ‘Snowball: Extraction Relations from Large Plain-Text Collections‘. In: *Proceedings of the fifth ACM conference on Digital Libraries*. San Antonio, Texas. S. 85-94.

Appelt, Douglas E. & Israel, David J. (1999): *Introduction to Information Extraction Technology*. A Tutorial prepared for IJCAI-99. Menlo Parc: SRI International. URL: https://www.dfki.de/~neumann/qa-course/doug-appelt-IJCAI99.pdf (zuletzt aufgerufen: 17.02.2021).

Biemann, Chris & Mehler, Alexander, Hrsg. (2014): *Text Mining. From Ontology Learning to Automated Text Processing Applications*. Cham: Springer Verlag.

Bidgoli, Hossein, Hrsg. (2002): *Encyclopedia of Information Systems*. 1. Aufl. Cambridge, Massachusetts: Academic Press.

Binnewitt, Johanna (2020): *Concept Mining – Disambiguierung extrahierter Terme am Beispiel von Stellenanzeigen*. Masterarbeit. URL: https://dh.phil-fak.uni-koeln.de/sites/dighum/user_upload/5249692_Binnewitt_Johanna_24-03-2020_MA-Arbeit_web__1_.pdf (zuletzt aufgerufen: 18.02.2021).

Brin, Sergey (1998): ‘Extracting Patterns and Relations from the World Wide Web‘. In: *WebDB ’98: Selected papers from the International Workshop on The World Wide Web and Databases*. Berlin/Heidelberg: Springer Verlag. S. 172-183.

Carstensen, Kai-Uwe et al., Hrsg. (2010): *Computerlinguistik und Sprachtechnologie. Eine Einführung*. 3. Aufl., Heidelberg: Spektrum akademischer Verlag GmbH.

Chang, Chia-Hui et al. (2003): ‘Automatic information extraction from semi-structured Web pages by pattern discovery‘. In: *Decision Support Systems 35(1)*. S. 129-147.

Chinchor, Nancy (1991): ‘MUC-3 Evaluation Metrics‘. In: *MUC3 '91: Proceedings of the 3rd Conference on Message Understanding*. San Diego/Kalifornien: Association for Computational Linguistics. S. 17-24. URL: https://dl.acm.org/doi/pdf/10.3115/1071958.1071961 (zuletzt aufgerufen: 18.02.2021).

Chinchor, Nancy (1992): ‘MUC-4 Evaluation Metrics‘. In: *MUC4 '92: Proceedings of the 4th conference on Message understanding*. San Diego/Kalifornien: Association for

Computational Linguistics. S. 22-29. URL: https://dl.acm.org/doi/pdf/10.3115/1072064.1072067 (zuletzt aufgerufen: 18.02.2021).

Collins, Michael & Singer, Yoram (1999): ‘Unsupervised Models for Named Entity Classification‘. In: *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora,: 21-22 June 1999, University of Maryland, College Park, MD, USA*. S. 100-110. URL: https://www.aclweb.org/anthology/W99-0613/ (zuletzt aufgerufen: 18.02.2021).

Geduldig, Alena (2017): *Muster und Musterbildungsverfahren für domänenspezifische Informationsextraktion. Ein Bootstrapping-Ansatz zur Extraktion von Kompetenzen aus Stellenanzeigen.* Masterarbeit. URL: https://dh.phil-fak.uni-koeln.de/sites/spinfo/arbeiten/Masterthesis_Alena.pdf (zuletzt aufgerufen: 18.02.2021).

Grishman, Dan (2019): ‘Twenty-five years of information extraction‘. In: *Natural Language Engineering*. Hrsg. Cambridge University Press. Volume 25, Issue 6. S. 677-692. DOI: https://doi.org/10.1017/S1351324919000512 (zuletzt aufgerufen: 18.02.2021).

Gudivada, Venkat N. & Tolety, Siva Perraju (1997): ‘A multiagent architecture for information retrieval on the World-Wide Web‘. In: *RIAO ’97: Computer-Assisted Information Searching on Internet*. Paris: Le centre de hautes etudes internationales d’informatique documentaire. S. 296-309. URL: https://dl.acm.org/doi/pdf/10.5555/2856695.2856722 (zuletzt aufgerufen: 18.02.2021).

Hermes, Jürgen & Schandock, Manuel (2016): ‘Stellenanzeigenanalyse in der Qualifikationsentwicklungsforschung. Die Nutzung maschineller Lernverfahren in der Klassifikation von Textabschnitten‘. In: *Fachbeiträge im Internet*. Bonn: Bundesinstitut für Berufsbildung. URL: https://www.bibb.de/veroeffentlichungen/de/publication/show/8146 (zuletzt aufgerufen: 17.02.2021).

Jackson, Peter & Moulinier, Isabelle (2002): ‘Information Extraction‘. In: *Natural Language Processing for Online Applications*. Hrsg. von Ruslan Mitkov. Bd. 5. Natural Language Processing. Amsterdam / Philadelphia: John Benjamins Publishing Company, S. 75–118.

Jean-Louis, Ludovic et al. (2011): ‘Text Segmentation and Graph-based Method for Template Filling in Information Extraction‘. In: *Proceedings of the 5th International Joint*

Conference on Natural Language Processing. Chiang Mai: Springer Verlag. S. 723-731.

Jurafsky, Martin & Martin, James H. (2009): *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. (Prentice Hall Series in Artificial Intelligence)*. 2. Aufl., New Jersey, USA: Pearson Prentice Hall.

Manning, Christopher D. et al. (2008): *Introduction to Information Retrieval*. New York: Cambridge University Press.

Nadeau, David & Sekine, Satoshi (2007): *A Survey of Named Entity Recognition and Classification*. Lingvisticae Investigationes. DOI: 30. 10.1075/li.30.1.03nad (zuletzt aufgerufen: 18.02.2021).

Neumann, Günther (2010): 'Text-basiertes Informationsmanagement'. In: *Computerlinguistik und Sprachtechnologie. Eine Einführung*. Hrsg. von Kai-Uwe Carstensen et al., 3. Aufl., Heidelberg: Spektrum akademischer Verlag GmbH. S. 576-615.

Neumann, Mandy (2015): *Analyse von Anforderungsprofilen. Eine Studie zur Informationsextraktion von Stellenanzeigen.* Masterarbeit. URL: https://spinfo.phil-fak.uni-koeln.de/fileadmin/spinfo/projekte/bibb/Masterarbeit_Neumann_final.pdf (zuletzt aufgerufen: 18.02.2021).

Okurowski, Mary Ellen (1993): 'Information Extraction Overview'. In: *TIPSTER '93: Proceedings of a workshop on held at Fredericksburg, Virginia: September 19-23,1993*. Fredericksburg/Virginia: Association for Computational Linguistics. S. 117-121. URL: https://dl.acm.org/doi/pdf/10.3115/1119149.1119164 (zuletzt aufgerufen am: 18.02.2021).

Riloff, Ellen (1996): 'Automatically Generating Extraction Patterns from Untagged Text'. In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence*. Portland/ Oregon: AAAI Press. S. 1044-1049. URL: https://www.researchgate.net/publication/50520410_Automatically_generating_extraction_patterns_from_untagged_text (zuletzt aufgerufen: 18.02.2021).

Riloff, Ellen & Jones, Rosie (1999): 'Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping'. In: *Proceedings of the Sixteenth National Conference on Artificial Intelligence*. Orlando/Florida: American Association for Artificial

Intelligence. S. 474-479. URL: https://www.aaai.org/Papers/AAAI/1999/AAAI99-068.pdf (zuletzt aufgerufen: 18.02.2021).

Riloff, Ellen & Sheperd, Jessica (1997): ‘A Corpus-Based Approach for Building Semantic Lexicons‘. In: *Second Conference on Empirical Methods in Natural Language Processing.* S. 117-124. URL: https://www.aclweb.org/anthology/W97-0313.pdf (zuletzt aufgerufen: 18.02.2021).

Sarawagi, Sunita (2007): ‘Information Extraction‘. In: *Foundations and Trends in Databases*. Vol. 1, No. 3, Hanover/MA: Now Publishers Inc. S. 261-377. URL: https://dl.acm.org/doi/10.1561/1900000003 (zuletzt aufgerufen: 18.02.2020).

Schwarz, Monika & Chur, Jeanette (2007): *Semantik. Ein Arbeitsbuch*. 5. aktualisierte Aufl., Tübingen: Gunter Narr (Narr-Studienbücher). S. 53-60.

Sun, Ang (2009): ‘A Two-staging Bootstrapping Algorithm for Relation Extraction‘. In: *Proceedings of Recent Advances in Natural Language Processing*. Borovets, Bulgarien. S. 76-82.

Thelen, Michael & Riloff, Ellen (2002): ‘A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts‘. In: *EMNLP ’02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*. Volume 10. USA: Association for Computational Linguistics. S. 214-221. URL: https://dl.acm.org/doi/10.3115/1118693.1118721 (zuletzt aufgerufen: 18.02.2021).

Eidesstattliche Versicherung

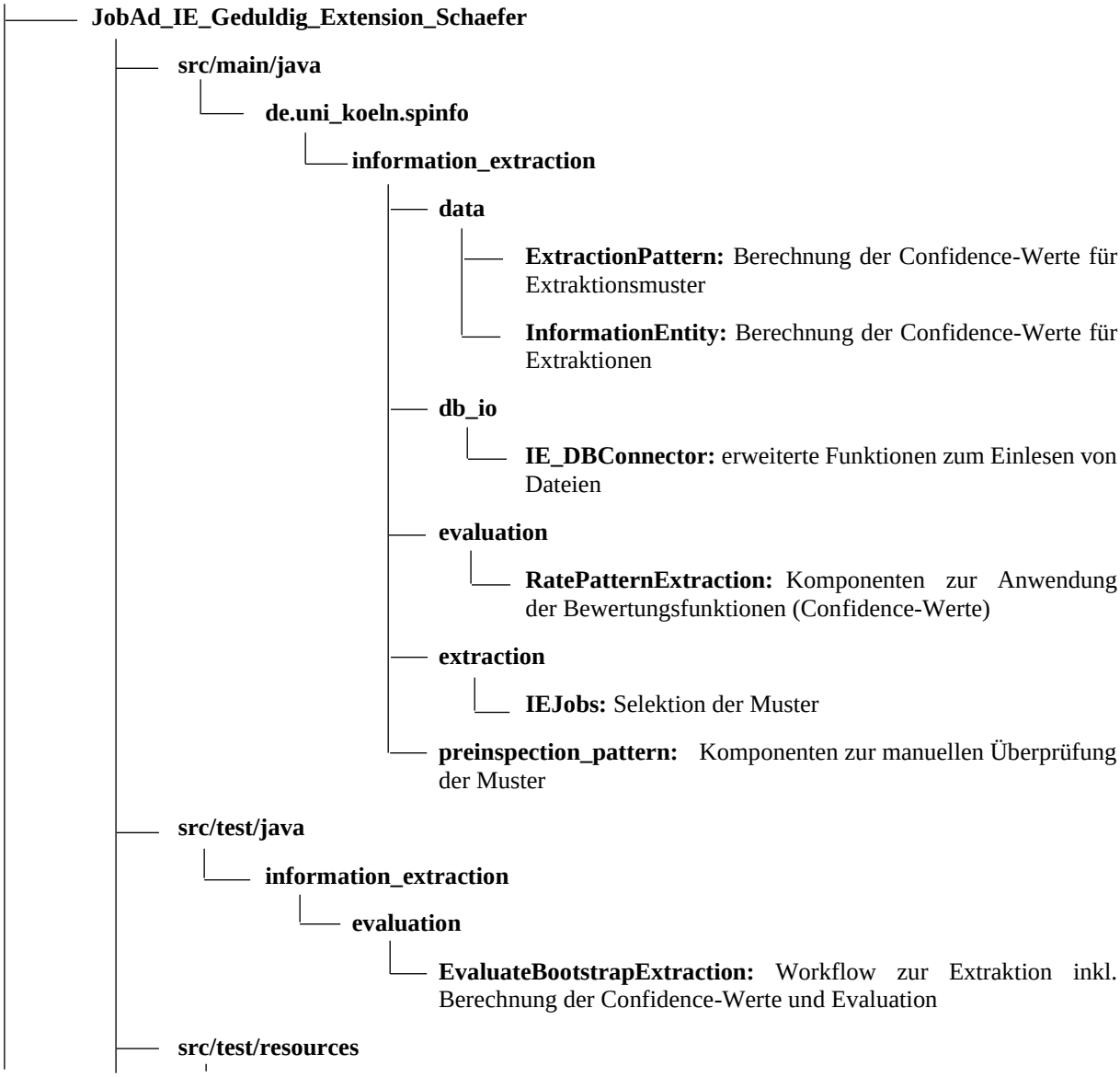
Hiermit versichere ich an Eides statt, dass ich diese Bachelorarbeit selbstständig verfasst und ohne die Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die Stellen meiner Arbeit, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche unter Angabe der Quelle kenntlich gemacht. Diese Arbeit habe ich in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt. Ich versichere, dass die eingereichte elektronische Fassung der eingereichten Druckfassung vollständig entspricht.

Köln, den 12. März 2021 Unterschrift:

Anhang

A Hinweise zur beiliegenden Implementation

Dieser Arbeit ist eine CD beigefügt, die sowohl die Arbeit als elektronische Fassung wie auch den Quelltext der Software-Komponenten, die zur Durchführung der innerhalb der Arbeit aufgeführten Workflows gebraucht werden, beinhalten. Das Programm liegt als Projektarchiv (zip-Datei) vor und kann entpackt und innerhalb einer IDE ausgeführt werden. Das Projekt ist eine Erweiterung des Programms von Geduldig 2017, enthält dementsprechend dessen Code. Alle vorgenommenen Ergänzungen wurden innerhalb des Projekts erkenntlich ausgezeichnet. Die vorliegende Paketstruktur kann vom bestehenden Projekt übernommen werden[42] und wird durch folgende Klassen und weitere Dateien ergänzt:



[42] Die Paketstruktur ist unter https://github.com/geduldia/JobAd_IE aufgeführt.

└── **information_extraction**

    └── **input:** Listen mit bereits bekannten Kompetenzen und Extraktionsfehlern (vom BIBB zur Verfügung gestellt)

B Abkürzungsverzeichnis

ARPA	Advanced Research Projects Agency
BA	Bundesinstitut für Arbeit
BIBB	Bundesinstitut für Berufsbildung
CRF	Conditional Random Fields
FN	False-Negatives, Falsch-Negative
FP	False-Positives, Falsch-Positive
HMM	Hidden Markov Modelle
HTML	Hypertext Markup Language
IDH	Institut für Digital Humanities
IE	Information Extraction, Informationsextraktion
IR	Information Retrieval
MUC	Message Understanding Conference
NE	Named Entity, Eigenname
NER	Named Entity Recognition
NLP	Natural Language Processing
NP	Nominalphrase
POS	Part of Speech
SVM	Support Vector Machines
TP	True-Positives, Richtig-Positive
WSD	Word Sense Disambiguierung