Universität zu Köln

Institut für Linguistik

Sprachliche Informationsverarbeitung

MASTERARBEIT

Muster und Musterbildungsverfahren für domänenspezifische Informationsextraktion

Ein Bootstrapping-Ansatz zur Extraktion von Kompetenzen aus Stellenanzeigen

vorgelegt von: Alena Geduldig

Matrikelnummer: 4455967
E-Mail: ageduldi@uni-koeln.de

Datum der Abgabe: 25.04.2017

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

1 Einleitung

1.1 Motivation

Ein Großteil der enormen und stetig wachsenden Menge an digital verfügbarer Information liegt in Form natürlichsprachlicher Texte vor. Für einen menschlichen Leser stellt es keine allzu große Herausforderung dar, darin gezielt bestimmte Informationen zu finden und zu strukturieren. Schwierigkeiten entstehen aber dann, wenn die Menge der zu durchsuchenden Texte ein angemessenes Kontingent an Zeit und Kosten übersteigt. In einem solchen Fall ist es naheliegend, diese Aufgabe dem Computer zu überlassen, anstelle einzelne oder auch mehrere Personen mit der Sichtung von Tausenden bis Millionen Dokumenten zu beauftragen. Für einen Computer stellt eine Verarbeitung solch großer Datenmengen keine Schwierigkeit dar. Herausforderungen sind dahingegen die vielfältigen Formulierungen und Repräsentationsformen von Informationen in natürlichsprachlichen Texten, die aus maschineller Sicht schwieriger zu erkennen und kategorisieren sind, als für einen menschlichen Leser.

Mit der Problemstellung, aus natürlichsprachlichen Texten Informationen zu gewinnen, befasst sich das Gebiet der maschinellen Sprachverarbeitung (Natural Language Processing, NLP). Ein Teilbereich der NLP ist die Informationsextraktion (Information Extraction, IE), welche die maschinelle Bearbeitung der oben skizzierten Aufgabe zum Ziel hat: In freien Texten relevante Informationen aufzuspüren und einheitlich zu strukturieren. Aus heterogen formulierten Informationen in Form natürlicher Sprache erzeugt die IE strukturierte Daten z. B. in Form einheitlicher Datenbankeinträge. Ein Vorteil von auf diese Weise strukturierten Daten liegt in ihrem Potential zur maschinellen Weiterverarbeitung, beispielsweise für statistische Auswertungen oder Data Mining Analysen.

Ein interessantes Anwendungsbeispiel für die IE bietet die Domäne der Stellenanzeigen. Betrachtet man beispielsweise ein Korpus von Stellenanzeigen, das im Verlauf mehrerer Jahre zusammengestellt wurde, erhält man wertvolle Informationen über die Entwicklung des Arbeitsmarktes in den betreffenden Jahren. Eine strukturierte und einheitliche Form der in den Anzeigen enthaltenen Informationen ließe etwa Auswertungen über die Entwicklung gefragter Bewerberkompetenzen oder den Einsatz von Arbeitsmitteln in bestimmten Berufszweigen zu. Denkbar wären auch Trendanalysen zu zukünftigen Entwicklungen des Arbeitsmarktes. Da Stellenanzeigen in der Regel in Fließtextform, zumindest jedoch in heterogener und schwach

strukturierter Form vorliegen, setzen diese oder ähnliche Analysen eine Aufbereitung der Daten durch ein IE-System voraus.

1.2 Ziele und Gliederung

Das Bundesinstitut für Berufsbildung (BIBB)[1] verfügt über eine Datenbank von mehreren Millionen Stellenanzeigen aus den vergangenen Jahren. Als Kompetenzzentrum zur Erforschung der beruflichen Aus- und Weiterbildung in Deutschland sind diese Daten für das BIBB eine wertvolle Informationsquelle. Für die Sprachliche Informationsverarbeitung (Spinfo) der Universität zu Köln stellt die umfangreiche Datenbank des BIBB eine einzigartige Ressource für die Erprobung und Weiterentwicklung von NLP und Text Mining Methoden dar. Seit Oktober 2015 werden die Intentionen der beiden Parteien in einem gemeinsamen Kooperationsprojekt bearbeitet, welches als langfristiges Ziel die Aufbereitung und analytische Auswertung der Datenbank verfolgt. Die Spinfo ist hierbei insbesondere mit der Entwicklung automatisierter Verfahren zur Informationsgewinnung beauftragt, die in stetiger Rücksprache mit dem BIBB erfolgt. Das BIBB wird die gewonnenen Informationen in Form strukturierter Daten im Anschluss inhaltlich auswerten (Data Mining) und unter anderem für Trendanalysen zum Arbeitsmarktgeschehen nutzen, um diese z. B. in den Zuschnitt und die Neuordnung von Berufsausbildungen einfließen zu lassen.

Gegenstand dieser Arbeit ist die Entwicklung einer Methodik zur Informationsextraktion aus Stellenanzeigen. Im Fokus liegt dabei die Extraktion von Anforderungsprofilen, also vom potentiellen Bewerber geforderte fachliche oder persönliche Kompetenzen, Bildungsabschlüsse und Berufserfahrungen. Langfristig sollen auch weitere Informationen (z. B. die im Job auszuführenden Tätigkeiten und verwendete Arbeitsmittel) extrahiert werden. Aufbauend auf einer Vorstudie zur Evaluation verschiedener Ansätze zur Lösung dieser Aufgabe (vgl. Neumann, 2015), wurde ein IE-System entwickelt, das zur Prozessierung sämtlicher im BIBB vorliegender Stellenanzeigen angewendet wird.

Aus Sicht der Computerlinguistik geht es immer auch darum, Werkzeuge und Methoden zu entwickeln, die nicht nur spezifisch in einem Aufgabenbereich eingesetzt werden können, sondern auch auf andere Domänen übertragbar sind. Die Aufgabe der Informationsextraktion

[1] https://www.bibb.de/

aus Stellenanzeigen (hier spezifisch: Extraktion von Kompetenzen) ist generalisierbar als eine Identifikation von sprachlichen Mustern, mit deren Hilfe die gesuchte Information detektiert werden kann. Ziel dieser Arbeit ist daher, einen Formalismus zu entwickeln, der die Formulierung solcher Muster erlaubt und in einem weiteren Schritt diese Musterformulierung zu automatisieren. Das Resultat soll anhand der spezifischen Aufgabenstellung evaluiert werden, sich aber auch auf andere Domänen übertragen lassen.

In Kapitel 2 (Informationsextraktion) dieser Arbeit werden die methodischen Grundlagen der IE erläutert und anschließend in Kapitel 3 (Domänenanalyse) auf die Domäne der Stellenanzeigen übertragen. Die wichtigsten Ergebnisse der Vorstudie werden zusammengefasst und für ein strategisches Fazit herangezogen. Kapitel 4 (Entwurf eines Formalismus zur Identifikation von Mustern) beschreibt die Implementation und Evaluation eines regelbasierten Ansatzes zur Informationsextraktion, der auf der manuellen Formulierung von Extraktionsregeln basiert. Hierfür wird ein Formalismus zur Codierung von sprachlichen Mustern entworfen. Die Ergebnisse dieses Verfahren werden im folgenden Kapitel (Automatische Musterbildung durch Bootstrapping) als Ausgangsbasis für ein iteratives Lernverfahren, dem sogenannten Bootstrapping, verwendet. Nach dem Vorbild des zuvor entwickelten Formalismus werden neue Regeln automatisch generiert. Kapitel 6 fasst die Ergebnisse dieser Arbeit abschließend zusammen und gibt einen Ausblick auf zukünftige Ideen und Vorhaben im Rahmen des Kooperationsprojektes und darüber hinaus.

2 Informationsextraktion

Die Informationsextraktion (IE) ist ein Teilbereich des Natural Language Processing (NLP) bzw. des Text Minings, also der maschinellen Verarbeitung natürlicher Sprache. Das Ziel der IE ist es, relevante Informationen in natürlichsprachlichen Texten aufzuspüren und in einheitlicher, strukturierter Form zu extrahieren. Im Unterschied zum Information Retrieval (IR), bei dem zu einer Informationsanfrage relevante Dokumente ermittelt werden, haben IE-Systeme den Anspruch, direkt verwertbare Information z. B. in Form von Datenbankeinträgen zu liefern. IR kann allerdings als Vorstufe zur IE dienen, um domänenrelevante Dokumente zu finden, die einem IE-System als Input und Wissensquelle dienen. In den un- bzw. semistrukturierten[2] Textdaten lokalisiert ein IE-System zunächst die für die Informationsanfrage relevanten Textfragmente und überführt diese anschließend in ein einheitliches und strukturiertes Format. Ein IE-System erfüllt somit zwei Kernfunktionen:

1. Erkennen relevanter Information bei gleichzeitigem Überlesen nicht relevanter Information – kurz: der Trennung des Relevanten vom nicht Relevanten.
2. Übertragen heterogen formulierter Informationen in ein einheitliches und somit zur maschinellen Weiterverarbeitung geeignetes Format.

Eine Definition dessen, was als relevant gilt, wird in der Regel[3] in Form von Templates (Schablonen) spezifiziert, welche gleichzeitig die Struktur der extrahierten Daten festlegen. Templates bestehen aus Attribut-Wert-Paaren, die ein tabellarisches Antwortmuster vorgeben. Die Attribute (z.B. Firmennamen, Orte oder auch Bewerberkompetenzen) legen fest, welche Felder (Slots) vom IE-System gefüllt werden müssen und welchem Datentyp die geforderte Information angehört. Die Aufgabe eines IE-Systems besteht in der Instanziierung solcher Templates, die entweder mit direkt aus dem Text entnommenen Textfragmenten oder zuvor normalisierter Information gefüllt werden (vgl. Jurafsky & Martin, 2009: 786).

---

[2] Aus Sicht der Informatik sind Textdaten semistrukturierte Daten, weil sie keiner allgemeinen Struktur unterliegen, sondern einen Teil der Strukturinformationen nur implizit tragen.

[3] Unter dem Überbegriff Open Information Extraction wurden im letzten Jahrzehnt auch vermehrt IE-Systeme entwickelt, die ohne spezifische Informationsanfrage in Form von Templates operieren (vgl. Banko et al., 2007; Wu & Weld, 2010; Etzioni et al., 2005).

Abbildung 1 veranschaulicht den Zusammenhang zwischen Text und instanziiertem Template anhand einer Pressemitteilung zur Oscarverleihung. Im Text sind jeweils die vom Template geforderten Attribut-Werte markiert, die – teilweise normalisiert – in die entsprechenden Template-Slots übertragen wurden.

<table>
<tr>
<td>Am 28. Februar 2016 konnte Leonardo DiCaprio bei der 88. Verleihung der Oscars endlich seinen Fluch besiegen. Im sechsten Anlauf konnte DiCaprio in der von Comedian Chris Rock moderierten Verleihung erstmals den Academy Award als bester Hauptdarsteller abräumen - für seine Rolle in "The Revenant".</td>
<td>
<table>
<tr><td>Preis</td><td>Oscar</td></tr>
<tr><td>Preisträger</td><td>Leonardo DiCaprio</td></tr>
<tr><td>Preiskategorie</td><td>Bester Hauptdarsteller</td></tr>
<tr><td>Film</td><td>The Revenant</td></tr>
<tr><td>Datum</td><td>2016/02/28</td></tr>
</table>
</td>
</tr>
</table>

Abbildung 1: Beispiel für ein instanziiertes Template zu einer Pressemitteilung der Oscarverleihung

Die Einsatzmöglichkeiten von IE sind vielfältig und reichen von der kompakten Darstellung von Informationen für einen Endnutzer, über die Erstellung großer abfragbarer Wissensdatenbanken bis zur Datengenerierung für systematische Data Mining Analysen. Für einen Endnutzer kann Informationsextraktion beispielsweise nützlich sein, um im Internet gezielt nach Jobangeboten in bestimmten Städten oder mit bestimmten Qualifikationsanforderungen zu suchen. Auch für den Preisvergleich von Produkten auf unterschiedlichen Onlineshops werden IE-Systeme verwendet. Für die IE aus Webseiten, die neben reinen Textdaten über umfangreiche Formatinformationen in Form von HTML-Tags verfügen, hat sich die Bezeichnung *Wrapper* etabliert. *Wrapper* operieren vordergründig auf der Grundlage von Formatinformationen (vgl. z.B. Eikvil 1999).

2.1 Geschichte

Die Idee, strukturierte Informationen aus natürlichsprachlichen Texten zu extrahieren, entstand in den späten 60er Jahren und resultierte unter anderem in den IE-Systemen LSP (Linguistic String Project, Sager 1981) zur Extraktion fachspezifischen Wissens aus medizinischen Texten

und FRUMO (Fast Reading Understanding and Memory Program, DeJong 1979), einem Programm zur Extraktion der wichtigsten Fakten aus Nachrichtenartikeln. Die Etablierung der IE als eigenständiges Forschungsfeld zu Beginn der 90er Jahre, ist insbesondere der staatlichen Förderung durch die US-Regierung in Form der Message Understanding Conference (MUC) zu verdanken. Die jährlich stattfindende und wettbewerbsorientiere Forschungstagung förderte die Entwicklung neuer IE-Methoden und Standards zur Evaluation von IE-Systemen. Dass die IE später ein fruchtbares Forschungsfeld wurde, zeigt sich auch mit der Entwicklung erster kommerzieller Systeme wie *Jango* (Doorenbros et al., 1997) und das später vom Onlinehändler Amazon aufgekaufte IE-System *Junglee*[4] zum online Preisvergleich von Produkten (Rajaraman, 1998).

Einer der wichtigsten Fortschritte in der IE-Entwicklung war die Weiterentwicklung von monolithischen und stark domänenabhängigen Systemen hin zur Modularisierung der IE in teilweise domänenunabhängige und adaptierbare Komponenten.

2.2 Aufgaben und Komponenten

So vielfältig Informationsanfragen an IE-Systeme auch sein können, lassen sie sich doch in folgende drei Hauptdisziplinen gruppieren, die sich in Art und Komplexität der zu extrahierenden Information unterscheiden:

1. (Named) Entity Recognition
2. Relation Extraction
3. Event Extraction

Entitäten sind die zentralen Einheiten eines Textes, die sich als Instanzen einer semantischen Klasse zuordnen lassen. Eine spezielle Untergruppe sind Named Entities, also benannte Entitäten wie Personen, Orte oder Organisationen. Bereits Entitäten können eine komplexe Templatestruktur besitzen, zum Beispiel mit zusätzlichen Slots für den Titel und das Geschlecht von Personen. Named Entity Recognition (NER) ist ein in der Regel domänenunabhängiger Task und oft nur ein erster Schritt für weiterführende Extraktionen. **Relation Extraction** hat die Erkennung von Beziehungen zwischen Entitäten zum Ziel. Typische Beispiele sind die Arbeitnehmer-Arbeitgeber-Beziehung zwischen Personen oder die Hauptsitz-Relation zwischen

[4] http://www.junglee.com/

Unternehmen und Orten. **Event Extraction** (auch Scenario Extraction) beschreibt die komplexeste Form der IE. Es werden Templates, die komplexe Szenarien oder Interaktionen zwischen Entitäten beschreiben, instanziiert. Im Rahmen der MUC wurden beispielsweise Templates für terroristische Aktivitäten formuliert (MUC-3, 1991).

Zur Lösung der IE-Aufgaben kommen bekannte NLP-Techniken zum Einsatz, die den betreffenden Text mit unterschiedlichen linguistischen Informationen anreichern und auf dessen Grundlage ein IE-System die Extraktionen vornimmt. Umfang und Tiefe der linguistischen Analyse hängen von der Komplexität der zu extrahierenden Information und den spezifischen Anforderungen an das IE-System ab (vgl. Piskorski & Yangarber, 2012:32). Abbildung 2 zeigt die wesentlichen Kernkomponenten, die den meisten IE-Systemen gemein sind.

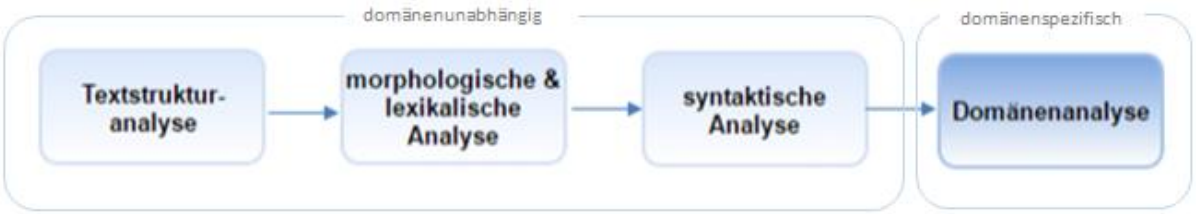


Abbildung 2: Typische Architektur eines IE-Systems – Linguistische Analyse und darauf aufbauende domänenspezifische Komponente

Als ersten Schritt werden die zu verarbeitenden Texte in strukturelle Einheiten, meist Paragraphen, Sätze und Tokens zerlegt**.** Eine darauf aufbauende morphologische und lexikalische Analyse umfasst das Bestimmen von Wortarten (Part-of-Speech, POS), Grundformen flektierter Wörter (Lemmata) und ggf. anderer wortgebundener Merkmale wie Modus und Tempus von Verben. Insbesondere für die Relations- und Eventextraktion spielt auch die syntaktische Struktur der Texte eine wichtige Rolle. Da die IE keinen Anspruch auf ein vollständiges Textverständnis erhebt, kommen meist nur flache und fragmentarische Parsingmethoden zum Einsatz (vgl. Neumann, 2010). Auf die in den vorangehenden Schritten aufgebaute Struktur setzt schließlich die eigentliche Extraktionskomponente auf, welche sich die linguistischen Informationen zur Erkennung domänenrelevanter Muster (Patterns) zu Nutze macht. Patterns sind linguistische Merkmalskombinationen, die als Vorlage fungieren, um templaterelevante Abschnitte im Text zu lokalisieren. Die Kernaufgabe jedes IE-Systems ist somit die Erkennung und Bildung von Patterns für die jeweils relevante Informationsanfrage.

2.3 Design

Zur Entwicklung eines IE-Systems gibt es zwei grundlegende Ansätze, die sich hauptsächlich in der Identifikation und Bildung der Patterns unterscheiden. Beim **Knowledge Engineering Ansatz** werden die Extraktionsregeln manuell erstellt. Das IE-System ist somit abhängig von der Kompetenz eines menschlichen Experten, welcher die Patterns auf der Grundlage seines domänenspezifischen Wissens formuliert. Dies macht den Knowledge Engineering Ansatz einerseits sehr zeit- und kostenintensiv, ermöglicht andererseits aber auch sehr präzise, auf die jeweilige Domäne angepasste Patterns. Dem gegenüber steht der Ansatz des **Automatischen Trainings**, bei dem die Patterns anhand eines vorausgezeichneten Trainingskorpus automatisch induziert werden. Dabei wird die linguistische Struktur um die relevanten annotierten Textstellen schrittweise verallgemeinert, bis sie auch auf neue nicht annotierte Daten anwendbar sind (vgl. Neumann, 2010: 21). Welcher Ansatz letztlich zu bevorzugen ist, hängt unter anderem von verfügbaren Ressourcen wie Zeit, Geld, Expertenwissen und Trainingskorpora ab. Liegt der Fokus auf bestmöglicher Performance, kann eine zeitintensive dafür aber sehr präzise manuelle Regelerstellung lohnenswert sein. Automatisches Training gewährleistet dahingegen in der Regel eine schnellere Adaptierbarkeit auf andere Domänen ohne domänenspezifisches Expertenwissen (vgl. Appelt & Israel, 1999: 8f). Voraussetzung hierfür ist jedoch die Verfügbarkeit eines ausreichend großen Trainingskorpus, sowie das Vorhandensein automatisch erkennbarer Patterns.

In den vergangenen Jahren hat sich zudem ein weiteres Verfahren in der IE-Forschung etabliert. Das sogenannte Bootstrapping ist eine iterative Lernmethode, die in der IE zur Extraktion von Relationen zwischen bereits annotierten Entitäten angewendet wird (vgl. z. B. Agichtein & Gravano, 2000 und Sun, 2009). Methodisch lässt es sich in den Bereich des schwach überwachten Lernens einordnen. Anstelle eines vorausgezeichneten Trainingskorpus bildet die Ausgangsbasis lediglich ein kleines Set von in der gesuchten Relation stehenden Entitäten-Paaren. Ausgehend von diesem Set, das auch Seed (Saatgut) genannt wird, lässt sich das Verfahren als mehrfache Iteration über die folgenden drei Schritte skizzieren:

1. Suche nach Seed-Kontexten im Korpus. (I. d. R. alle Sätze, die beide Entitäten in unmittelbarer Nähe enthalten)
2. Automatisches Erzeugen von Extraktionsmustern (Patterns) durch Generalisierung der gefundenen Textstellen.

3. Erweiterung der Seeds durch Anwendung der neuen Patterns.

Eine bekannte Bootstrap-Anwendung ist das IE-System *Snowball,* ein System, das zur Extraktion von Unternehmens- und Hauptsitzrelationen entwickelt wurde (Agichtein & Gravano, 2000). Es beinhaltet zusätzliche Iterationsschritte zur Evaluation neuer Patterns und Seed-Paare, um zu verhindern, dass ‚schlechte' Patterns oder falsche Seed-Paare in den Iterationsprozess einfließen (vgl. Abbildung 3).

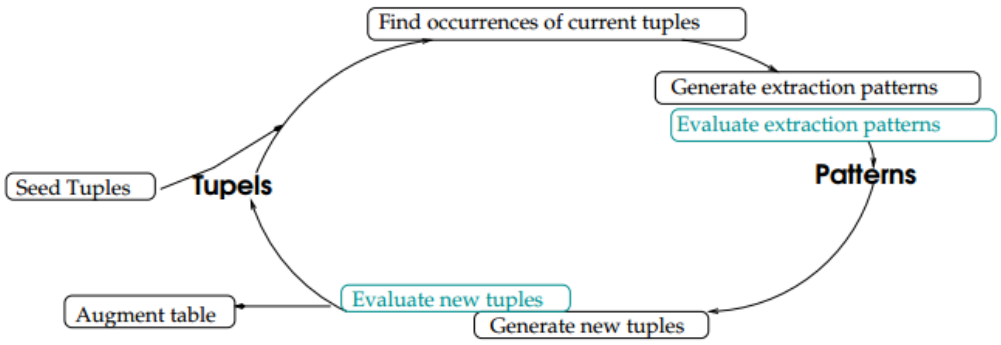


Abbildung 3: Schematische Darstellung des Snowball-Systems zur Relationsextraktion. (Quelle: Agichtein & Gravano, 2000)

3 Domänenanalyse

Nachdem im vorangehenden Kapitel die methodischen Grundlagen der IE dargelegt wurden, befasst sich dieses Kapitel mit der Domäne der Stellenanzeigen und insbesondere mit dem Bereich der Bewerberkompetenzen. Eine genauere Betrachtung des Aufbaus und der Gestalt von Stellenanzeigen soll Aufschluss über ihr Potential für das Feld der IE geben und eine Einordnung in die oben genannten Teilaufgaben ermöglichen.

Eine grundlegende Voraussetzung zur Entwicklung eines IE-Systems ist eine genaue Eingrenzung der zu extrahierenden Information. Für das System selbst, aber auch zur manuellen Auszeichnung von Trainings- oder Evaluationsdaten ist eine klare Definition der Informationsanfrage notwendig. Im Fall der Bewerberkompetenzen zeigten sich hierbei verschiedene domänen- und anwendungsspezifische Herausforderungen, die im Verlaufe dieses Kapitels erläutert werden.

3.1 Stellenanzeigen

Die Domäne der Stellenanzeigen stellt ein typisches und produktives Anwendungsgebiet für die Informationsextraktion dar. Stellenanzeigen stehen in vielfacher Form online und offline zur Verfügung und werden regelmäßig direkt auf den Webseiten von Unternehmen oder auf speziellen Jobangebotsplattformen veröffentlicht. In der Regel werden die Angebote in Fließtextform verfasst – häufig auch mit stichwortartigen Elementen. Man spricht von un- oder semistrukturierten Daten, da kein einheitliches Schema existiert. Aus rein inhaltlicher Sicht weisen Stellenanzeigen jedoch ein relativ ähnliches Informationsmuster auf. So werden mehrheitlich dieselben und immer wiederkehrenden Informationskategorien bedient, die sich daher gut für die Überführung in ein einheitliches Format eignen. Hierbei handelt es sich zum Beispiel um die Bezeichnung der ausgeschriebenen Stelle, den Namen oder Hauptsitz des suchenden Unternehmens oder das vorgesehene Einstellungsdatum. Auch inhaltliche Angaben zum Tätigkeitsfeld oder dem Bewerberprofil sind in der Regel in jedem Jobangebot aufgeführt. Des Weiteren kommt einer automatischen Informationsextraktion zu Gute, dass Stellenanzeigen meist eine kanonische Struktur aufweisen, die das Suchfeld für gezielte Informationsanfragen auf einzelne Abschnitte eingrenzen können. So lassen sich die meisten Anzeigen in die vier thematischen Kategorien Unternehmensbeschreibung, Jobbeschreibung, Bewerberprofil und Formalia gliedern. Diese sind meist auch strukturell, z. B. durch Absätze, hervorgehobene

Überschriften, oder den Wechsel von Fließtext zu Listenform voneinander abgegrenzt, wie das Beispiel in Abbildung 4 zeigt. Der Name des ausschreibenden Unternehmens ist nach einer erfolgreichen Trennung der vier Bereiche dann beispielsweise nur noch im Abschnitt der Unternehmensbeschreibung zu suchen. Für die Extraktion von Bewerberkompetenzen bleibt nur der Abschnitt zum Bewerberprofil relevant.

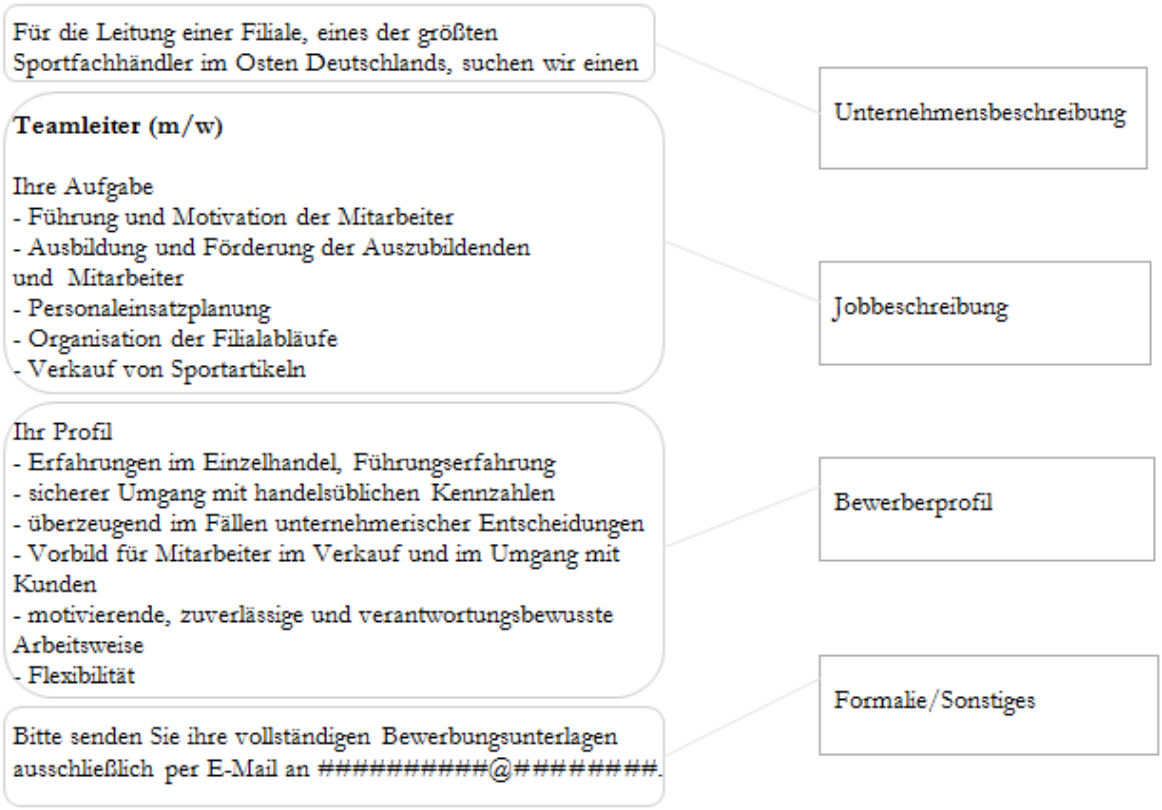


Abbildung 4: Gliederung einer anonymisierten Stellenanzeige in die vier inhaltsmotivierten Kategorien: Unternehmensbeschreibung, Jobbeschreibung, Bewerberprofil und Formalia.

Mit Bezug auf die oben erläuterten Teildisziplinen der IE handelt es sich im Fall von Stellenanzeigen vorwiegend um die Extraktion von Entitäten mit gegebenenfalls mehreren Attributen. Methodisch ist zwischen domänenunabhängigen Named Entities wie Firmennamen, E-Mail-Adressen oder Datumsangaben und domänenspezifischen Entitäten wie zum Beispiel Bewerberkompetenzen oder Tätigkeiten zu unterscheiden. Im ersten Fall kann auf domänenunabhängige Verfahren zur Named Entity Recognition zurückgegriffen werden. Im Gegensatz zu E-Mail-Adresse haben Kompetenzen oder Tätigkeiten jedoch keine

domänenübergreifende Gültigkeit. Letztere erfordern also eine auf Stellenanzeigen angepasste Domänenanalyse.

3.2 Bewerberkompetenzen

Bereits zu Beginn der Arbeiten an diesem Projekt und in den Absprachen mit dem BIBB bezüglich der genauen Leistungsbeschreibung des IE-Systems, zeigte sich, dass eine klare Definition des Inhalts und der Gestalt der zu extrahierenden Information keinesfalls trivial ist und sowohl inhaltlich als auch strukturell an die beabsichtigte Verwendung der extrahierten Daten angepasst werden muss. Eine manuelle Sichtung des Korpus hinsichtlich struktureller Auffälligkeiten und Besonderheiten in der Formulierung von Bewerberkompetenzen macht diese Notwendigkeit deutlich. Unter Berücksichtigung der dabei gewonnenen Erkenntnisse kann schließlich eine genaue Definition der Informationsanfrage erfolgen, die für die weiterführenden Arbeiten als Leitfaden dient.

3.2.1 Manuelle Inspektion des Korpus

Bei einer genaueren Untersuchung des betroffenen Korpus lassen sich in einem großen Teil der Anzeigen Ähnlichkeiten bezüglich der Formulierung von Bewerberprofilen feststellen. Grundsätzlich lässt sich eine Unterscheidung zwischen stichwortartigen Aufzählungen und der Formulierung in vollständigen, teilweise sehr komplexen Sätzen vornehmen. Beispiele für beide Formatvarianten sind in Tabelle 1 aufgeführt.

Listen werden oft durch einen einleitenden Satz (z. B. „Das erwarten wir von Ihnen: “) oder einen kurzen Titel (z. B. „Ihr Profil: “) eingeleitet. Ein einzelner Listenpunkt besteht häufig aus einer durch Kommata oder Konjunktionen getrennten Aufzählung mehrerer Kompetenzen (1b, 2b). Manchmal stellt ein Listenpunkt auch eine eigenständige Kompetenz dar (2a, 2c). Auffällig ist außerdem die Verwendung von Modifizierern (etwa „wünschenswert“ oder „zwingend erforderlich“), die angeben wie stark eine Kompetenz gewünscht ist und meist direkt vor oder hinter der Kompetenz stehen (1a, 1c).

Auch im Fließtext sind Kompetenzen häufig von Modifizierern umgeben, jedoch eingebettet in komplexere und heterogene syntaktische Konstruktionen. Es fällt auf, dass viele Kompetenzen als verbales Argument bestimmter Signalverben, etwa *erwarten, besitzen* oder *benötigen* auftreten.

(1) a) b) c)	Voraussetzung: - **Eigener Pkw**, **Führerschein** und **Kranschein** erforderlich - **zuverlässig, flexibel, motiviert** - **Erfahrung im Messebereich** wünschenswert aber nicht erforderlich
(2) a) b) c)	Ihr Profil: - **Abgeschlossene Ausbildung zum Mechatroniker** - **Verantwortungsbewusstsein** und **handwerkliches Geschick** - **Reisebereitschaft**
(3)	Hierfür werden umfangreiche **Kenntnisse in der Elektrotechnik**, **ein sicherer Umgang mit dem PC** als auch sehr gute **EDV-Kenntnisse** benötigt.
(4)	**Berufserfahrung** wäre wünschenswert
(5)	hierfür setzen wir den **Besitz eines Führerscheins und Pkws** voraus
(6)	Idealerweise haben Sie **erste Erfahrung im Gastronomiebereich**
(7)	Im Idealfall verfügen sie über **umfangreiche Erfahrungen im Pflegebereich** und besitzen eine **adäquate Ausbildung** bzw. **fühlen sich den Aufgaben gewachsen**

Tabelle 1: Beispiele für die Formulierung von Bewerberkompetenzen in Stellenanzeigen in Listenform (1-2) und Fließtextform (3-7)

Zusammenfassend lässt sich festhalten, dass durchaus Regelmäßigkeiten erkennbar sind, die durch eine Generalisierung in linguistischen Patterns beschrieben werden könnten. Im Zuge dessen fallen jedoch auch unregelmäßige und problematische Formulierungen auf. Beispielhaft sind die Bewerberprofile in Tabelle 2 aufgeführt, die insbesondere zwei Herausforderungen deutlich machen:

Einerseits sind viele vom Bewerber geforderten Fähigkeiten nicht als kompakte, vom Kontext zu trennende Entität formuliert, sondern durch ausschweifende Formulierungen, die selbst die manuelle Annotation einer konkreten Kompetenzeinheit schwierig machen. Häufig verschwimmen auch die Grenzen zwischen Bewerberkompetenz und im Job auszuführenden

Tätigkeiten (2, 3). Andererseits scheinen diese Kompetenzen mit generalisierenden Kontextmustern – ob nun manuell oder maschinell produziert – auf Grund ihrer heterogenen Kontexte kaum erfassbar.

(1)	Mit Ihrer kaufmännischen Ausbildung beherrschen Sie nicht nur den PC, sondern sind auch eine gewissenhafte und redegewandte Persönlichkeit, die mit Charme und Pfiff ihren Schreibtisch im Griff hat. Ihr Engagement kommt aus dem Herzen und Sie lassen sich auch von urlaubsreifen Mitmenschen nicht auf die Palme bringen. Ihr gesundes Selbstbewusstsein äußert sich nicht nur in einer gepflegten Erscheinung, sondern auch in Durchsetzungsvermögen und Nachhaltigkeit. Sie integrieren sich mit Leichtigkeit in ein bestehendes Team, behaupten sich durch selbständiges Denken und Handeln mit einer angenehm positiven Einstellung.
(2)	Das Lesen und professionelle Anfertigen von technischen Bauplänen bereitet Ihnen keine Schwierigkeiten.
(3)	Darüber hinaus sind Sie in der Lage auch in schwierigen Situationen den Überblick und die Ruhe zu bewahren
(4)	Stets gut gelaunt behalten Sie auch in stressigen Situationen einen kühlen Kopf
(5)	Zudem besitzen Sie die Kompetenz, erklärungsbedürftige Sachverhalte zu analysieren und schriftlich gut strukturiert und verständlich aufzubereiten.

Tabelle 2: Weitere Beispiele für die Formulierung von Bewerberkompetenzen ohne regelhafte Auffälligkeiten

Das eigentliche Konzept von Entitäten als kompakte Grundeinheiten eines Texts (vgl. Kapitel 2.2) geht verloren und stellt die IE vor methodische Herausforderungen. Für eine maschinelle statistische Auswertung der Extraktionsergebnisse, wie sie im Anschluss der Extraktion vom BIBB geplant ist, sind komplexe Formulierungen, (wie die in Tabelle 2) zudem äußerst ungeeignet. Für den maschinellen Vergleich sind möglichst kanonische und normalisierbare Formulierungen notwendig. Für das zu entwickelnde IE-System liegt somit neben dem Anspruch auf möglichst vollständige Extraktionsergebnisse, der Fokus insbesondere auf der maschinellen Verwertbarkeit dieser. Im Gegensatz zur üblichen NER oder Relationsextraktion ist also eine zusätzliche Herausforderung zu bewältigen. Nur dann macht jedoch die IE auf so großen Datenmengen überhaupt erst einen Sinn.

Mit Rücksicht auf diese Herausforderung wurde ein Kompetenzbegriff entwickelt, der eine klare Ein- und Abgrenzung der vom IE-System zu extrahierenden Information und ihrer Struktur vornimmt. Bei der Implementierung des IE-Systems und der manuellen Auszeichnung von Evaluationsdaten dient dieser als struktureller und inhaltlicher Leitfaden.

3.2.2 Definition eines Kompetenzbegriffs

Zur Definition von Bewerberkompetenzen konnten verschiedene inhaltliche Kategorien ausgemacht werden, die für die Extraktion in Frage kommen. Hierzu zählen sämtliche den potentiellen Bewerber beschreibenden Charaktereigenschaften, berufliche und fachliche Qualifikationen wie Abschlüsse, Aus- und Weiterbildungen, Scheine und Lizenzen sowie Arbeitsmittel und Fachbereiche in denen Kenntnisse oder Erfahrungen gefordert werden. Im Folgenden sind Beispiele für alle vier Kategorien aufgeführt.

1. **Den Bewerber beschreibende Adjektive und substantivische Eigenschaften**
   Beispiele:
   - flexibel
   - tatkräftig
   - technisches Verständnis
   - hohe Auffassungsgabe
   - Verantwortungsbewusstsein

2. **Abschlüsse, Ausbildungen und Berufserfahrungen**
   Beispiele:
   - Ausbildung zum KFZ-Mechaniker
   - Kaufmännische Ausbildung
   - Abgeschlossene Berufsausbildung
   - Realschulabschluss
   - Abitur
   - Abschluss zum staatlich geprüften Betriebswirt

3. **Scheine und Lizenzen**
   Beispiele:
   - Führerschein
   - Gesundheitszeugnis
   - Belehrung nach Paragraph §43.1
   - Gabelstaplerschein

4. **Kenntnisse und Erfahrungen mit/in Arbeitsmitteln und Fachbereichen**
   Beispiele:
   - Schweißkenntnisse
   - Montageerfahrung
   - Kenntnisse im Verkehrsrecht
   - Erfahrung mit ms-office
   - Kenntnisse in Java
   - Übung in der Warenannahme

Mit Blick auf die spätere maschinelle Verwertbarkeit der extrahierten Information wurden zusätzliche strukturelle Einschränkungen vorgenommen:

5. **Kompaktheit/Einhaltung des Konzepts von Entitäten**

   Extrahiert werden nur Nominalphrasen oder Adjektivphrasen:

   - Mindestens 5 Jahre Berufserfahrung in leitender Position eines Autohauses
   - Verhandlungssicheres Englisch und kommunikative Geschicklichkeit sind für diesen Job unabdingbar.
   - Ihr Engagement kommt aus dem Herzen und Sie lassen sich auch von urlaubsreifen Mitmenschen nicht auf die Palme bringen.
   - Ein hohes Maß an Professionalität ist für sie selbstverständlich

   Die einzige Ausnahme bilden Abschlüsse und Ausbildungen. Hier sind auch Präpositionalphrasen zulässig:

   - Ausbildung zum Betriebswirt
   - Hochschulabschluss in Mathematik
   - Abschluss zum medizinisch technischen Assistenten

Tabelle 3 demonstriert die Anwendung dieser Restriktionen anhand einiger Beispielsätze. In Beispiel (3) werden keine Kompetenzen extrahiert, da keine kompakte Entität annotiert werden kann.

(1)	Mit Ihrer **kaufmännischen Ausbildung** beherrschen Sie nicht nur den PC, sondern sind auch eine **gewissenhafte** und **redegewandte** Persönlichkeit, die mit Charme und Pfiff ihren Schreibtisch im Griff hat. Ihr **Engagement** kommt aus dem Herzen und Sie lassen sich auch von urlaubsreifen Mitmenschen nicht auf die Palme bringen. Ihr **gesundes Selbstbewusstsein** äußert sich nicht nur in einer **gepflegten Erscheinung**, sondern auch in **Durchsetzungsvermögen** und **Nachhaltigkeit**. Sie integrieren sich mit Leichtigkeit in ein bestehendes Team, behaupten sich durch **selbständiges Denken** und Handeln mit einer angenehm **positiven Einstellung**.	- kaufmännische Ausbildung - gewissenhaft - redegewandt - Engagement - gesundes Selbstbewusstsein - gepflegte Erscheinung - Durchsetzungsvermögen - Nachhaltigkeit - selbstständiges Denken - positive Einstellung
(2)	Ideal ist, wenn Sie viel **positive Energie** haben, gerne **Verantwortung** übernehmen und gerne Mitarbeiter führen. Gebraucht werden auch **gute Englischkenntnisse** und **Führerschein** Klasse B. Außerdem sollten Sie idealerweise **ausgelernt** haben und mindestens schon einmal in einer Klinik gearbeitet haben und die Abläufe kennen.	- positive Energie - Verantwortung - gute Englischkenntnisse - Führerschein - ausgelernt
(3)	Das Finden von kreativen Lösungen ist für Sie selbstverständlich.	

Tabelle 3: Annotationsbeispiele unter Berücksichtigung des zuvor definierten Kompetenzbegriffs.

3.3 Vorarbeiten und Ausgangspunkt

Vorbereitend auf das im Rahmen dieser Arbeit entwickelte IE-System wurde 2014 eine Vorstudie zur Informationsextraktion aus Stellenanzeigen in Form einer Masterarbeit durchgeführt (Neumann, 2014). In dieser Vorstudie wurden unterschiedliche manuelle und maschinell lernende IE-Verfahren getestet und im Hinblick auf die Extraktion von Bewerberkompetenzen evaluiert. Zur Beurteilung des Knowledge Engineering Ansatzes wurden auf regulären Ausdrücken basierende Extraktionsregeln formuliert, die Kompetenzen als Zeichenkette in bestimmten sprachlichen Kontexten lokalisieren. Die Extraktionsregeln beruhten auf der Beobachtung sich wiederholender Muster, die in Stellenanzeigen zum Ausdruck von Bewerberkompetenzen verwendet werden. So wurde zum Beispiel ein Muster entworfen, dass sämtliche alphanumerischen Zeichen zwischen den sprachlichen Konstrukten ‚wir setzen' und ‚voraus' extrahiert. Darüber hinaus wurde die Extraktion mithilfe eines Dependenzparsers getestet, welcher Kompetenzen als Argumente zuvor ausgewählter Verben (z. B. suchen, wünschen oder erwarten) extrahiert. Beispielhaft für den Ansatz des maschinellen Lernens wurde die Extraktion

durch einen Naive Bayes Klassifikator evaluiert, welcher einzelne Tokens, entweder als (Teil einer) Kompetenz, oder als keiner Kompetenz zugehörig klassifiziert.

Als Ergebnis der Studie wurde die regelbasierte Extraktion mithilfe von regulären Ausdrücken als das erfolgreichste der erprobten Verfahren ermittelt und eine Weiterentwicklung des Knowledge Engineering Ansatzes als vielversprechendsten Lösungsansatz für diesen Anwendungsfall empfohlen (vgl. Neumann, 2014: 39). Ausblickend empfiehlt Neumann, die Mächtigkeit der regulären Ausdrücke mit den linguistischen Informationen aus vorverarbeitenden NLP-Schritten zu vereinen und in komplexeren Extraktionsmustern miteinander zu kombinieren (ebd.). Auch in anderen Studien zur IE aus Stellenanzeigen und weiteren Domänen erreichten Systeme, die mit manuell erstellten Regeln arbeiten, bessere Werte als automatisierte Lernverfahren (vgl. etwa Soderland, 1999; Califf, 1998; Bsiri & Geierhos, 2007).

Aufbauend auf den Ergebnissen der Vorstudie wird auch in dieser Arbeit zunächst ein manueller Ansatz zur Patternbildung verfolgt. Als Fortschritt gegenüber der Verwendung regulärer Ausdrücke für den Eingabestring wird ein Formalismus zur Codierung von mehrdimensionalen Patterns entwickelt, der beispielsweise auch die Referenzierung von POS-Tags, Lemmata oder Satzpositionen ermöglicht. Wichtig ist hierbei, den Output der Kontextmuster auf kompakte Entitäten im Sinne der im Kompetenzbegriff gesetzten Restriktionen zu beschränken. Muster, die sämtlichen Content zwischen zwei Ausdrücken oder die Argumente von Verben extrahieren, sind somit nicht geeignet. Entsprechende Experimente mit shallow-geparstem Input wurden gemacht, fanden aber keinen Niederschlag. Stattdessen sind Patterns erforderlich, die nicht nur den die Entität umgebenden Kontext spezifizieren, sondern auch die Struktur der Entität selbst.

Für die manuelle Erstellung von Regeln spricht in erster Linie die Möglichkeit der präzisen Anpassung auf eine breite Domäne, die in diesem Fall alleine im BIBB mehre Millionen (bereits vorhandene und zukünftige) Stellenanzeigen umfasst. Im weiteren Verlauf der Arbeit sollen die Ergebnisse dieses Ansatzes zudem als Ausgangsbasis (Saatgut) für ein auf die Extraktion von Entitäten angepasstes Bootstrap Verfahren dienen. Das Ziel ist es, die durch manuelle Muster extrahierten Kompetenzausdrücke, auch in anderen Stellenanzeigen aufzuspüren und auf diese Weise neue Kontexte für die Formulierung von Kompetenzen zu ermitteln. Aus diesen Kontexten können neue Extraktionsmuster automatisch generiert werden, die wiederum den nächsten Extraktionsprozess anstoßen.

4 Entwurf eines Formalismus zur Identifikation von Mustern

Dieses Kapitel dokumentiert die Extraktion von Kompetenzen mit Hilfe von manuell erstellten Extraktionsregeln. Es wird ein Regelformalismus entworfen, der die Formulierung von präzisen Extraktionsmustern auf linguistisch vorverarbeiteten Daten ermöglicht. Diese Vorverarbeitung wird im folgenden Abschnitt erläutert.

4.1 Linguistische Vorverarbeitung

Um die Effektivität eines IE-Systems zu erhöhen, kann es hilfreich sein, das Suchfenster für bestimmte Informationen bereits von vornherein einzugrenzen. Um den Gesamttext auf einzelne Textpassagen zu begrenzen, nutzen einige Systeme *Triggering*, also die Suche nach bestimmten Signalwörtern (Triggern), die auf das Vorhandensein der gewünschten Information in unmittelbarer Nähe hinweisen (vgl. Jackson & Moulinier, 2002: 75). Im Fall der Stellenanzeigen bietet sich alternativ ein Klassifikationsverfahren an, welches einzelne Abschnitte den vier thematischen Klassen Unternehmensbeschreibung, Jobbeschreibung, Bewerberprofil und Formalia zuordnet (vgl. Kapitel 3.1). In einem vorgeschalteten Projekt wurde dieser Arbeitsschritt bereits erfolgreich umgesetzt (vgl. Geduldig, 2015a; Geduldig, 2015b; Hermes & Schandock, 2016). Das IE-System operiert somit nur noch auf den Textabschnitten[5], die im Klassifikationsverfahren der Klasse der Bewerberprofile zugeordnet wurden.

Die betroffenen Paragraphen werden anschließend in noch kleine Einheiten (Sätze bzw. einzelne Listenelemente) zerlegt, die vom IE-System nacheinander verarbeitet werden. Diese *ExtractionUnits* werden zunächst tokenisiert und die Tokens im Anschluss mit zusätzlichen linguistischen Informationen wie Lemmata oder POS-Tags angereichert, die später aus den Extraktionsmustern referenziert werden können. Hierfür wurden die an der Universität Stuttgart entwickelten Mate-Tools[6], ein Toolkit für verschiedene NLP-Methoden, verwendet. Darüber hinaus werden am Satzanfang und Satzende zusätzliche Tokens zur Markierung von Satzgrenzen

[5] Die Trennung der Abschnitte erfolgte durch einen selbst implementierten *ClassifyUnitSplitter,* welcher den Anzeigentext nicht nur an Leerzeilen trennt, sondern auch berücksichtigt, dass einige Abschnitte trotz räumlicher Trennung zusammengehörig sind. (Z.B. Listenelemente oder die oft formativ abgesetzte Stellenbezeichnung)

[6] https://code.google.com/archive/p/mate-tools/

eingefügt. Diese können später in den Extraktionsmustern zur Spezifikation von Satzpositionen genutzt werden. Der vollständige Vorverarbeitungsprozess ist in Abbildung 5 schematisch dargestellt.

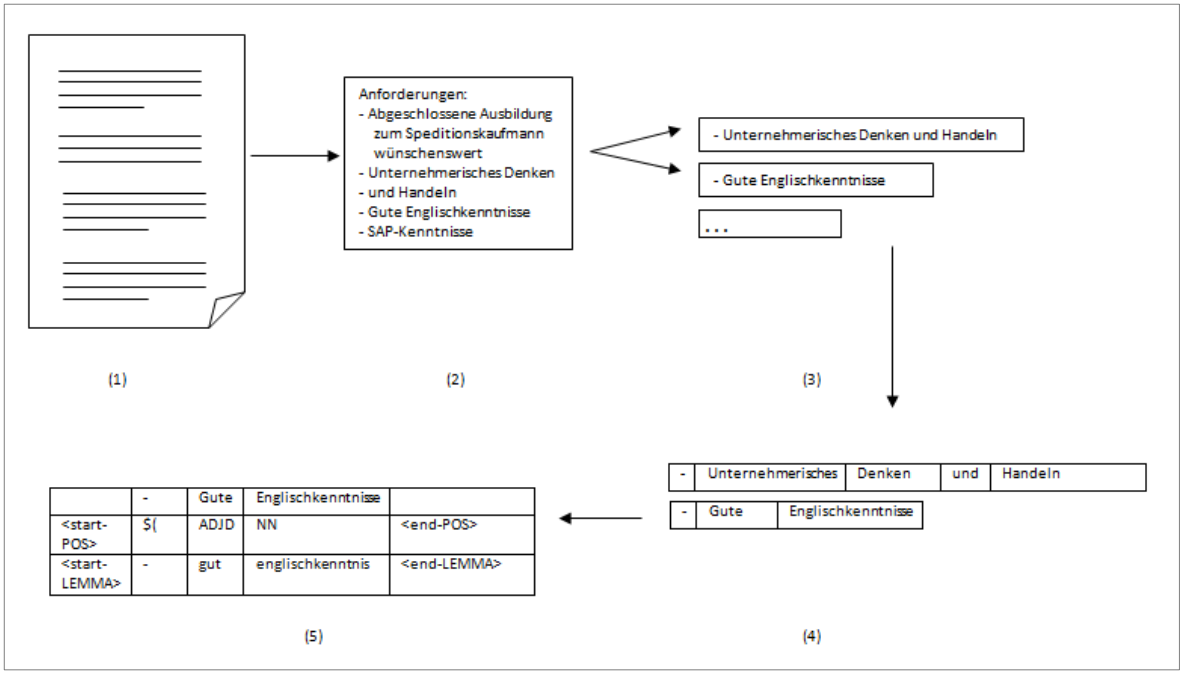


Abbildung 5: Schematische Darstellung der Präprozessierung eines Paragraphen. Aus der Stellenanzeige (1) wird zunächst der Paragraph (2) gefiltert, der in die Klasse der Bewerberprofile klassifiziert wurde. Dieser wird in Sätze (3) und in Tokens (4) zerlegt, welche anschließend durch weitere linguistische Informationen wie POS-Tags und Lemmata angereichert werden (5).

4.2 Extraktionsregeln

Wie bereits erwähnt, existieren zwei grundsätzlich verschiedene Ansätze zur Entwicklung eines IE-Systems. Sie unterscheiden sich hauptsächlich in der Erstellung der Extraktionsregeln. Anders als beim Automatischen Training, werden die Extraktionsregeln beim Knowledge Engineering manuell erstellt. Das Verfahren lässt sich wie folgt beschreiben:

> The Knowledge Engineering Approach is characterized by the development of the grammars used by a component of the IE system by a "knowledge engineer," i.e. a person who is familiar with the IE system, and the formalism for expressing rules for that system, who then, either on his own, or in consultation with an expert in the domain of application, writes rules for the IE system component that mark or extract the sought-after information.
>
> (Appelt & Israel, 1999: 7)

Dieser Ansatz erfordert einerseits umfangreiches domänenspezifisches Wissen und andererseits einen geeigneten Formalismus zur Regelerstellung, in dem die zur Extraktion relevanten Merkmale referenziert werden können. Im dritten Teil der Arbeit konnten bereits typische Schemata für die Formulierung von Kompetenzen ausgemacht werden, die sich für eine Formulierung in generalisierte Extraktionsmuster anbieten. So sind Kompetenzen beispielsweise häufig von bestimmten Signalwörtern umgeben, die die Wichtigkeit derselben modifizieren. Dazu wird im Folgenden ein Formalismus präzisiert, mit dem diese beobachtbaren Muster auf ihre notwendigen und hinreichenden Merkmale abstrahiert werden sollen.

Entwurf eines Regelformalismus

Es wurde ein Formalismus entworfen, der es erlaubt Extraktionsregeln als linguistische Kontexte zu formulieren. Diese fungieren als Schablonen, die wie ein Fenster über den unbekannten Text gelegt werden können und im Falle einer Übereinstimmung den oder die zu extrahierenden Token(s) indexieren. Die Kontexte bestehen aus einer Folge von geforderten Tokens, die entweder über ihren String, ihr Lemma, das POS-Tag und/oder die Satzposition näher spezifiziert werden.[7] Falls die vorgegebene Tokenfolge im zu durchsuchenden Text gefunden wird, gibt das Muster außerdem vor, welche der übereinstimmenden Tokens als Kompetenz extrahiert werden. Beispielhaft sind nachfolgend zwei Kontextmuster abgebildet[8]:

```
PATTERN-ID:  5
TOKEN:       null    ausbildung|abschluss|studium    null
TOKEN:       null    zu|als|in|der                   null
TOKEN:       null    null                            NN
TOKEN:       null    null                            ¬NN
EXTRACT:     0,1,2,3


PATTERN-ID:  12
TOKEN:       null    <root-LEMM>    <root-POS>
TOKEN:       null    null           XY
TOKEN:       null    null           ADJ|NN|NE
TOKEN:       null    null           ADJ|NN|NE
TOKEN:       null    null           <end-POS>|$,|KON
EXTRACT:     2, 3
```

Listing 1: Beispiele für Kontextmuster zur Kompetenzextraktion

[7] Ebenfalls wurde versuchsweise der Output des Dependenzparsers (syntaktische Abhängigkeiten und Rollen von Nominalphrasen) in den Formalismus mit einbezogen. Aufgrund des nur selten satzwertigen Inputs wurde dieser hier jedoch nicht berücksichtigt.

[8] Sämtliche Kontextmuster befinden sich als Textdatei im mitgelieferten Software-Projekt (src/test/resources/information_extraction/input/competence_paterns.txt). Aus expositorischen Gründen sind diese leicht vereinfacht dargestellt.

Die erste Zeile eines Kontextmusters dient der Zuweisung einer eindeutigen Kontext-Id. In den darauffolgenden Zeilen werden die vom Kontext geforderten Tokens in der Reihenfolge in der sie auftreten müssen beschrieben. Jeweils durch Tabstops getrennt werden String, Lemma und POS-Tag aufgeführt, wobei *null* als beliebiger Platzhalter fungiert. Durch einen Senkrechten Strich (|) werden verschiedene Alternativen ausgedrückt. Ein vorangestelltes Negationszeichen (¬) kehrt ein Merkmal um. Der erste Kontext mit der Pattern-Id 5 würde beispielsweise die folgenden Kompetenzen extrahieren:

Gerne Bewerber mit abgeschlossener [Ausbildung im Gastronomiebereich].

Ein [Studium der Informatik] oder einen vergleichbaren Studiengang setzen wir voraus.

Wir suchen eine pädagogische Fachkraft mit [Abschluss als Diplom-Sozialpädagoge_in].

Der mit dem Muster übereinstimmende Bereich ist in eckige Klammern gesetzt; die extrahierten Wörter sind unterstrichen. Die Spezifikation der rechten Kontextgrenze durch ein negiertes Nomen (¬NN) ist notwendig, um beispielsweise unvollständige Extraktionen, wie die folgenden zu vermeiden:

[*Abschluss zum AOK] Betriebswirt*

[Ausbildung im Gewerk] Lüftung Heizung

[Ausbildung im Bereich] Informatik

Der zweite Kontext fordert zuerst ein <root-Lemma>, also eine linke Satzgrenze. Anschließend das POS-Tag XY, welches im STTS-Annotationsschema einem Sonderzeichen (etwa einem Spiegelstrich oder anderen Aufzählungszeichen) entspricht. Es folgen zwei Adjektive (ADJ), Nomen (NN) oder Eigennamen (NE) und eine rechte Satzgrenze (<end-POS>), ein Komma ($,) oder eine Konjunktion (KON). Gültige Formulierungen zu diesem Kontext sind zum Beispiel:

[- gute Menschenkenntnis und] Erfahrung im Umgang mit Kunden

[- mittlere Reife oder] sehr guter Hauptschulabschluss

[◦ zeitlich flexibel,] teamfähig, begeisterungsfähig

[Microsoft Office]*

Für die Verwendung von modifizierenden Ausdrücken wie *wünschenswert, zwingend erforderlich* oder *von Vorteil* als Merkmale im Kontextmuster hat sich der Gebrauch von festen Wortlisten als praktikabel herausgestellt, um die Zahl der Kontexte zu minimieren und Wiederholungen zu vermeiden. Unter der Prämisse, dass diese Modifizierer eine abgeschlossene und begrenzte Menge bilden, wurden die Ausdrücke in einer externen Textdatei gesammelt[9]. In den Kontextmustern können diese nun über das Lemma MODIFIER referenziert werden:

```
    PATTERN-ID:  18
    TOKEN:       null    null          <root-POS>|$,|KON|ART
    TOKEN:       null    null          NN|NE|ADJ
    TOKEN:       null    null          NN|NE
    TOKEN:       null    sein          null
→   TOKEN:       null    MODIFIER      null
    TOKEN:       null    null          <end-POS>|&,|$.|KON
    EXTRACT:     1, 2
```

Listing 2: Beispiel für ein Kontextmuster mit Modifier-Referenzierung

Der in Pattern 18 skizzierte Kontext extrahiert Nominalphrasen, die links von einer Satzgrenze einem Komma, Artikel oder einer Konjunktion und rechts durch das Lemma *sein* und einen modifizierenden Ausdruck begrenzt werden. Beispiele hierfür sind:

[Technisches Verständnis wird ***vorausgesetzt****]*

FS [und eigener PKW sind ***von Vorteil,****] außerdem sollten sie…*

[Ein gültiger Staplerschein ist zwingen notwendig]

Durch die Begrenzung der linken und rechten Kontextseite auf Satzgrenzen, Konjunktionen oder Kommata wird auch hier sichergestellt, dass keine falschen oder unvollständigen Extraktionen wie die folgenden vorgenommen werden:

Führerschein ~~Klasse B~~ ist erforderlich

FS und PKW sind zwecks ~~Arbeitsplatzerreichung~~ zwingend erforderlich

[9] Die Datei mit sämtlichen modifizierenden Ausdrücken befindet sich im mitgelieferten Software-Projekt. (src/test/resources/information_extraction/input/modifiers.txt)

Berufserfahrung im Montieren von ~~Fenstern~~ ist erforderlich

Der Entwurf der Kontextmuster erfolgte auf Grundlage einer manuellen Korpusinspektion. Eine Herausforderung ist hierbei, die richtige Balance zwischen zu spezifischen und zu mächtigen, ungenauen Regeln zu finden. Die Mustererstellung ist ein iterativer Prozess, der nach und nach optimiert wird:

> Building a high performance system is usually an iterative process whereby a set of rules is written, the system is run over a training corpus of texts, and the output is examined to see where the rules under- and overgenerate. The knowledge engineer then makes appropriate modifications to the rules, and iterates the process.
>
> (Appelt & Israel, 1999: 7)

Auch die Ermittlung der relevanten und notwendigen Merkmale erfolgte iterativ. So zeigte sich erst im Entwicklungsprozess, dass die Merkmale für linke und rechte Satzgrenzen zur Vermeidung falscher Extraktionen sehr hilfreich sind. Morphologische Merkmale stellten sich dahingegen als wenig hilfreich heraus. Insgesamt wurden zunächst 40 Extraktionsmuster formuliert. Um eine präzise Grundlage für die im Anschluss geplante Weiterentwicklung des IE-Systems zu bilden, wurde der Fokus hierbei auf eine möglichst hohe Präzision gelegt.

4.3 Workflow

In Abbildung 6 ist der Ablauf der musterbasierten Extraktion schematisch zusammengefasst.[10] Als Ergebnis der vorgeschalteten Klassifikation liegen die einzelnen Paragraphen (*ClassifyUnits*) zusammen mit der ihr zugeordneten Klasse in einer Datenbank vor. Die Paragraphen, die mindestens[11] der Klasse der Bewerberkompetenzen zugeordnet wurden, werden aus der Datenbank gelesen und – wie in Abschnitt 1 dieses Kapitels beschrieben – zu *ExtractionUnits* vorprozessiert. Die Wortliste mit den modifizierenden Ausdrücken wird eingelesen und die betroffenen Tokens als solche annotiert. Im Anschluss werden die Extraktionsmuster eingelesen und paarweise mit den *ExtractionUnits* verglichen. Im Falle einer Übereinstimmung wird der entsprechende Ausdruck extrahiert und in einer neuen Datenbank hinterlegt.

---

[10] Das vollständige Programm ist auf der beiliegenden CD oder unter https://github.com/geduldia/JobAd_IE zugänglich. Für Erläuterungen zur Projektstruktur vgl. Abschnitt A (Hinweise zur beiliegenden CD).

[11] Bei der Klassifikation der Paragraphen wurden auch Mehrfachzuordnungen vorgenommen, da beispielsweise Tätigkeiten und Anforderungen oft innerhalb eines Paragraphen spezifiziert wurden.

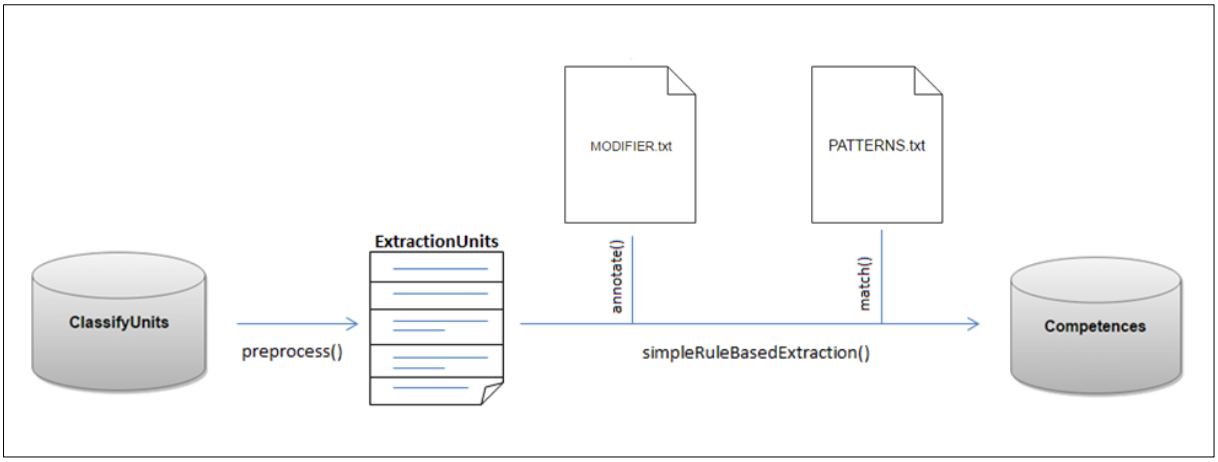


Abbildung 6: Schematische Darstellung der musterbasierten Kompetenzextraktion

Als positiver Nebeneffekt der Wortliste für Modifizierer, können die extrahierten Kompetenzen durch das zusätzliches Attribute *Modifier* ergänzt werden. Tabelle 4 zeigt einen Auszug aus der Ergebnisdatenbank:

ID	Jahrgang	Zeilennr	ParaID	Sentence	Competence	Modifier	Contexts
22	2011	266	e426b-014b-4bf	Technisches Verständnis wird vorausgesetzt.	technisch verständnis	vorraussetzen	1
23	2015	4555	e366f-022b-6cf	Gute Kenntnisse in ms-office sind zwingend erforderlich	ms-office	zwingend erforderlich	2
24	2010	655	b337f-063c-9df	Sie sind teamfähig, flexibel und zuverlässig	flexibel	null	1

Tabelle 4: Auszug aus der Ergebnisdatenbank: Spalten von links nach rechts: ID der Kompetenz, Jahrgang und Zeilennummer der ursprünglichen Stellenanzeige, Identifier des ursprünglichen Paragraphen, Satz der *ExtractionUnit*, extrahierte Kompetenz, Modifizierer, Anzahl der produzierenden Kontexte.

4.4 Evaluation

Zur Evaluation von IE-Systemen wurden im Rahmen der MUC (vgl. Abschnitt 2.1) entsprechende Metriken entwickelt. Die Ausgangsbasis bildeten dabei die im Information Retrieval bereits etablierten Maße Precision (Präzision) und Recall (Vollständigkeit). Übertragen auf die IE gibt die Precision den Anteil der korrekten Extraktionen unter allen vorgenommen

Extraktionen an. Der Recall drückt das Verhältnis zwischen den richtig extrahierten und den im Goldstandard annotiertem Ausdrücken an. Formel 4.1 veranschaulicht die Zusammenhänge:

$$precision = \frac{\#correct}{\#correct + \#incorrect}$$

$$recall = \frac{\#correct}{\#annotated} \tag{4.1}$$

Es ist schwierig, beide Werte gleichzeitig zu optimieren. Optimiert man ein IE-System auf eine hohe Präzision, steigt die Wahrscheinlichkeit, dass relevante Informationen unentdeckt bleiben. Bei einer Optimierung zu Gunsten der Vollständigkeit steigt die Gefahr, dass irrelevante Informationen extrahiert werden. Da beide Maße nur in Kombination ausschlaggebend für die Güte eines IE-Systems sind hat sich das F-Maß[12] - das gewichtete harmonische Mittel aus Precision und Recall – etabliert:

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{4.2}$$

Im Gegensatz zum Information Retrieval steht die Evaluation von IE-Systemen vor der Herausforderung, dass eine Dichotomie zwischen korrekt und inkorrekt – wie sie von den Metriken verlangt werden - nicht unmittelbar gegeben ist. So muss beispielsweise abgewägt werden, wie mit extrahierten Phrasen umgegangen wird, die zwar nicht mit der annotierten Vorgabe übereinstimmen, diese aber (zu Teilen) enthalten. Da diese Entscheidungen die Evaluationsergebnisse beeinflussen, sollten sie zusätzlich dokumentiert werden.

Zur Evaluation der Kompetenzextraktion wurde ein Testkorpus aus *ClassifyUnits* (Paragraphen) erstellt, die zuvor in die Klasse der Bewerberkompetenzen klassifiziert wurden. Die Paragraphen wurden maschinell in insgesamt 550 *ExtractionUnits* (Sätze bzw. Listenelemente) zerlegt und anschließend manuell annotiert.[13] Das Testkorpus liegt als Datenbank vor und beinhaltet sowohl

[12] Durch den Parameter β können Precision und Recall unterschiedlich stark gewichtet werden ($\beta > 1$ legt den Schwerpunkt auf den Recall, $\beta < 1$ auf die Precision.

[13] Hierfür wurde ein interaktiver Annotations-Workflow entwickelt, mit der Annotationen über die Konsole eingegeben werden können. Die entsprechende Klasse befindet sich im mitgelieferten Software-Projekt. (src/test/java/information_exraction/CreateCompetenceTrainingdata.java)

die *ExtractionUnits*, in denen Kompetenzen extrahiert wurden, als auch die, die keine Kompetenzen enthalten:

ID	Jahrgang	Zeilennr	ParaID	Sentence	Competence
22	2011	266	e426b-014b-4bf	Gerne mit Pkw und Führerschein	pkw
23	2011	266	e426b-014b-4bf	Gerne mit Pkw und Führerschein	führerschein
24	2010	655	b337f-063c-9df	Wir wünschen uns:	*NULL*

Tabelle 5: Auszug aus dem annotierten Testkorpus

Insgesamt wurden im Testkorpus 447 Annotationen vorgenommen wobei 271 verschiedene Kompetenzausdrücke annotiert wurden. In der Evaluation wurden diese mit den vom IE-System extrahierten Daten verglichen. Eine Extraktion galt hierbei als korrekt, wenn sie entweder mit der Annotation im Goldstandard übereinstimmt, oder wenn sie diese vollständig enthält. Überlappende Extraktionen wurden als inkorrekt bewertet.

Vom IE-System wurden insgesamt 250 Extraktionen vorgenommen. Von diesen wurden 244 als korrekt evaluiert und nur 6 als fehlerhaft, was einer Präzision von etwa 98% entspricht. Zu Gunsten dieser hohen Präzision, wurde von den 447 annotierten Kompetenzen jedoch nur knapp über die Hälfte identifiziert und damit ein Recall von 54 % erreicht.[14]

Mit dem manuellen Ansatz konnte somit ein gutes, jedoch in der Präzision noch ausbaufähiges Ergebnis erzielt werden. Für das darauf aufsetzend geplante Bootstrapping Verfahren, das ein kleines aber exaktes initiales Startset benötigt, konnte mit diesem Ergebnis eine gute Ausgangsbasis geschaffen werden. Im Fokus der fortgeführten Entwicklung steht nun die Verbesserung des Recalls, bei möglichst geringer Abnahme der Präzision.

[14] Die ausführbare Klasse zur Durchführung und Evaluation der Extraktion befindet sich im mitgelieferten Software-Projekt. (src/test/java/information_extraction/evaluation/EvaluateSimpleRulebasedExtraction.java)

5 Automatische Musterbildung durch Bootstrapping

Bereits bei der manuellen Analyse des Korpus zeigte sich schnell, dass nur ein Teil der gesuchten Kompetenzen in wahrnehmbar-regelhaften Kontexten auftritt. Die Evaluation der musterbasierten Extraktion hat dies ebenfalls bestätigt. Die Stärke des manuellen Verfahrens liegt stattdessen in seiner hohen Genauigkeit. Das Ziel der in diesem Kapitel beschriebenen Extraktionsstrategien ist somit eine Verbesserung der Vollständigkeit (Recall) bei möglichst geringen Einbußen in der Präzision.

Zur Verbesserung der Ausbeute wird das Konzept des Bootstrapping, das in der Regel zur Extraktion von Relationen eingesetzt wird, auf das Problem der Entity Extraction übertragen. Einleitend werden hierfür die drei wesentlichen Komponenten des Bootstrapping - das **initiale Startset**, die **Ermittlung neuer Kontexte** und die **Generierung neuer Extraktionsmuster** – vorgestellt und überprüft, inwieweit die jeweiligen Elemente auch zur Extraktion von Entitäten eingesetzt werden können.

Als eine der wenigen zugänglichen Referenzimplementationen für Bootstrapping in der IE wird im Folgenden das *Snowball*-System herangezogen. Es wurde 2000 entwickelt, um Relationen der Art „<Organisation>'s headquarters in <Location>" in großen Textdaten aufzudecken. Dem System ist eine domänenunabhängige Named Entity Recognition vorgeschaltet, durch die sämtliche Unternehmens- und Ortsnennungen bereits identifiziert und klassifiziert wurden. Die Aufgabe des *Snowball*-Systems ist also von der hier zu bewältigenden relativ stark unterschieden.

5.1 Initiales Startset

Als Ausgangsbasis für das Bootstrap Verfahren dient bei der Relationsextraktion ein initiales Set von in der gesuchten Relation stehenden Entitäten. Bei dem IE-System *Snowball* sind dies Paare von Städtenamen und in der Stadt ansässigen Firmen – beispielsweise <*Redmond, Microsoft*> oder <*Santa Clara, Intel*>. Für die Entity Extraction können die initialen Seeds aus einem Set von Entitäten - in diesem Fall also Kompetenzen - bestehen. Aufgrund der hohen Präzision bieten sich hierfür die Ergebnisse aus dem vorangehenden Knowledge Engineering Ansatz an. Im Unterschied zu den Relationspaaren sind Kompetenzen jedoch losgelöst von ihrem Gebrauchskontext nicht zwingend gültig. Die folgenden Beispielsätze aus dem Trainingskorpus veranschaulichen dies:

(1) *Bereitschaft zur 3-Schichtarbeit und Rufbereitschaft sind unbedingt erforderlich*

(2) *Für Mitarbeit und Verständigung im Bereich Notruf/Rufbereitschaft sind gute Deutschkenntnisse in Wort und Schrift erforderlich*

Während *Rufbereitschaft* in (1) eine Kompetenz darstellt, wird es in (2) zur Beschreibung eines Tätigkeitsbereiches verwendet. Da diese Art von ‚Fehlern' im bereits gut eingegrenzten Suchbereich jedoch als eher gering eingeschätzt wird und um dennoch an das Bootstrap Verfahren anknüpfen zu können, wird diese Problematik zunächst in Kauf genommen.

5.2 Ermittlung neuer Kontexte

Die initialen Seeds werden bei der Relationsextraktion zum Aufspüren von Gebrauchskontexten für die gesuchte Relation herangezogen. Dies sind in der Regel alle Sätze im Korpus, die beide Entitäten eines solchen Tupels beinhalten. Ein Kontext für das Tupel <*Redmond, Microsoft*> könnte zum Beispiel folgender Satz sein:

> *The Redmond-based Microsoft college internship program is highly competitive and highly regarded at a national level.*

Um neue und unbekannte Kontexte für Kompetenzen aufzudecken, werden die aus der musterbasierten Extraktion bekannten Entitäten anhand eines einfachen Stringvergleichs im gesamten Korpus gesucht. Sämtliche Sätze (bzw. *ExtractionUnits*), die einen bereits bekannten, aber in diesem Kontext nicht bereits durch manuelle Muster identifizierten, Kompetenzausdruck beinhalten, werden somit als Kompetenzkontext ausgewählt. Durch dieses simple Vorgehen werden zwar keine neuen Kompetenzen (Types) entdeckt, jedoch eine große Menge neuer bis dato unbekannter Kontexte, von denen das IE-System auf zweierlei Arten profitieren kann: Zum einen können Kompetenzen auch dort extrahiert werden, wo die manuell erstellten Extraktionsmuster nicht greifen. Dadurch wird der Recall erhöht ohne dass durch fälschlicherweise erkannte Kompetenzen die Präzision abnimmt. Zum anderen kann die große Zahl neuer Kontexte als Input und Vorlage für die automatische Generierung neuer Extraktionsmustern dienen.

5.3 Automatische Mustergenerierung

In der Theorie folgt die Generierung neuer Extraktionsmuster einem einfachen Prinzip: Die zuvor gewonnenen Beispielkontexte werden als Vorlage verwendet, um daraus neue

Extraktionsmuster zu abstrahieren. Hierbei spielen zwei Parameter eine wichtige Rolle: Agichtein & Gravano (2000) benutzen in diesem Zusammenhang die Begriffe *selectivity* (Selektivität) und *coverage* (Reichweite). Die neuen Muster sollen einerseits selektiv sein, also möglichst keine falschen Relationspaare extrahieren und andererseits möglichst viele neue Tupel identifizieren können. Die Begriffe sind auch als *precision* und *recall* für einzelne Patterns übersetzbar.

Zur Relationsextraktion im *Snowball*-System wird der jeweils linke, mittlere und rechte Kontext der Fundstellen als Vektor nach dem Vorbild des im Information Retrieval gebräuchlichen Vector-Space-Model[15] repräsentiert. Jeder Kontext kann somit als ein 5-Tupel <*left,* E_1*, middle,* E_2*, right*> repräsentiert werden, wobei *left, middle* und *right* Vektoren sind und E_1 und E_2 die in der Relation stehenden Entitäten. Diese 5-Tupel fungieren als Muster zum Aufdecken neuer Relationspaare. Die vektorbasierte Repräsentationsform hat den Vorteil, dass sie flexible Ähnlichkeitsberechnungen zulässt. Zwei als 5-Tupel repräsentierte Textstellen können als ähnlich identifiziert werden, auch wenn sie im Wortlaut nicht exakt übereinstimmen, zum Beispiel auf Grund eines zusätzlichen Kommas oder Determinierers. Dies ermöglicht ein gutes Gleichgewicht zwischen *selectivity* und *coverage*.

Übertragen auf die Entity Extraction könnte die Mustergenerierung insofern übernommen werden, dass anstelle von 5-Tupeln 3-Tupel <*left ,E, right*> generiert werden, wobei *left* und *right* Vektoren sind, die den linken und rechten Kontext neben einer Kompetenz (*E*) repräsentieren. Dieser Ansatz scheitert jedoch an der Anwendung der Muster zur Identifikation neuer Kompetenzen, wie im Folgenden erläutert wird.

Die Relationsextraktion baut darauf auf, dass im Vorfeld bereits eine domänenunabhängige Named Entity Recognition ausgeführt wurde. Im Fall des *Snowball*-Systems sind daher bereits sämtliche Orts- und Firmennamen als solche annotiert. Für das System ist damit genau eingegrenzt, wo neue potentielle Relations-Paare zu finden sind, nämlich dort wo ein Orts- und ein Firmenname in der vom Muster geforderten Reihenfolge in einem Satz stehen. Von diesen potentiellen Kontexten können somit ebenfalls 5-Tupel gebildet und mit den Mustern verglichen werden. Im Fall der Kompetenzextraktion ist jedoch nahezu jedes einzelne Wort und jede Kombination mehrerer Wörter eine potentielle Kompetenz. Es müsste somit für jede mögliche Wortkombination ein entsprechendes 3-Tupel gebildet werden, um es mit sämtlichen Mustern zu

[15] Zum Vector-Space-Model vgl. z.B. Manning et al., 2008.

vergleichen. Schon in einem Testkorpus übersteigt dies die kombinatorische Kapazität eines einsetzbaren Verfahrens, ganz zu schweigen von einer Anwendung auf mehreren Millionen Stellenanzeigen.

Um den Bootstrapping Ansatz dennoch anwenden zu können, wurde die Mustergenerierung angepasst. Anstelle der vektorbasierten Patterns wurden Muster nach dem Vorbild des in Kapitel 4 beschriebenen Regelformalismus automatisch generiert. Diese können, wie die manuell erstellten Patterns, als Schablonen über den Text gelegt werden. Dies hat zusätzlich den Vorteil, dass der Ansatz direkt in den bereits bestehenden Workflow integriert werden kann. Hierfür wurden jeweils verschiedene Kontextgrößen links und rechts vom entsprechenden Kompetenzausdruck festgesetzt und der entsprechende Bereich, wie in Listing 3, in den bekannten Formalismus übertragen. Die ursprünglichen Kompetenz-Tokens wurden hierbei (bis auf das POS-Tag) durch Default-Werte ersetzt, so dass auch neue Begriffe mit dem Muster aufgedeckt werden können (Zeile 2). Für weitere bereits identifizierte Kompetenzen im Satz wurde außerdem rechts eine zusätzliche Merkmalsspalte (*isCompetence)* eingeführt (Bsp. (1) Zeile 4).

(1) *Teamfähigkeit und Ehrgeiz*

```
TOKEN:      null          root-LEMMA        root-POS     false
TOKEN:      null          null              NN           false
TOKEN:      und           und               KON          false
TOKEN:      Ehrgeiz       ergeiz            NN           true
TOKEN:      null          end-LEMMA         end-POS      false
EXTRACT:    1
```

(2) *Physiotherapeutische Ausbildung oder vergleichbare Qualifikation*

```
TOKEN:      null          root-LEMMA        root-POS     false
TOKEN:      null          null              ADJ          false
TOKEN:      null          null              NN           false
TOKEN:      oder          oder              KON          false
TOKEN:      vergleichbare vergleichbar      ADJ          false
TOKEN:      Qualifikation qualifikation     NN           false
EXTRACT:    1,2
```

Listing 3: Automatisch generierte Muster. Anstelle des im Kontext verwendeten Kompetenzausdrucks wird nur das POS-Tag des Ausdrucks ins Muster aufgenommen.

Das Resultat kann bereits als Muster verwendet werden, ist jedoch in dieser unveränderten Form sehr spezifisch. Um das Verhältnis zwischen *selectivity* und *coverage* auszugleichen wurden daher zusätzlich diverse Default-Mechanismen getestet, von denen die folgenden in das System übernommen wurden:

1. Funktionswörter, etwa Konjunktionen, Artikel oder Pronomen werden auf ihr POS-Tag reduziert. Alle anderen Attribute werden auf *null* gesetzt.
2. a) Weitere Kompetenzausdrücke im Kontextausschnitt werden auf das Merkmal *isCompetence* reduziert.
   b) Modifizierende Ausdrücke werden auf das Merkmal *MODIFIER* reduziert.
3. Folgende funktionsähnliche POS-Tags werden zu Gruppen zusammengefasst:

   a) Konjunktionen (KON), und Kommata ($,)

   b) Rechte Satzgrenzen (<end-POS>) und Satzbeendende Satzzeichen ($.)

   c) Linke Satzgrenzen (<root-POS>) und Auflistungszeichen (XY)

Abbildung 5.2 zeigt ein Muster auf das alle drei Generalisierungsstrategien angewendet wurden.

(3) Technisches Verständnis und Lernbereitschaft zwingend erforderlich

```
TOKEN:      null   null        root-POS|XY        false
TOKEN:      null   null        null               true
TOKEN:      null   null        KON                false
TOKEN:      null   null        NN                 false
TOKEN:      null   MODIFIER    null               false
TOKEN:      null   null        end-POS|$.|$,      false
EXTRACT:    3
```

Listing 4: Automatisch generiertes Muster nach Anwendung der Generalisierungsstrategien 1 (Zeile 3), 2 (Zeile 2) und 3 (Zeile 5).

Durch die Anwendung der Default-Mechanismen werden die Patterns abstrahiert und verlieren ihre kontextgebundenen Spezifität. Sie können somit nicht nur Kompetenzen in identischen, sondern auch in ähnliche Textpassagen identifizieren.

5.4 Workflow

In Abbildung 7 ist der Ablauf des Bootstrapping Ansatzes schematisch dargestellt. Grundlage sind die aus dem Knowledge Engineering Ansatz extrahierten Types, welche als Seed-Kompetenzen die Ausgangsbasis für das Bootstrap-Verfahren bilden. Via Stringmatching werden neue Kontexte für die Seed-Kompetenzen identifiziert und im Anschluss zu neuen Extraktionspatterns generalisiert.[16] Durch die Anwendung dieser Patterns sollen im Idealfall neue Kompetenzen extrahiert werden, welche die Seed-Kompetenzen für den nächsten Schritt ergänzen. Diese Schritte werden so oft iteriert, bis keine neuen Kompetenzen bzw. Kompetenzkontexte mehr gefunden werden.

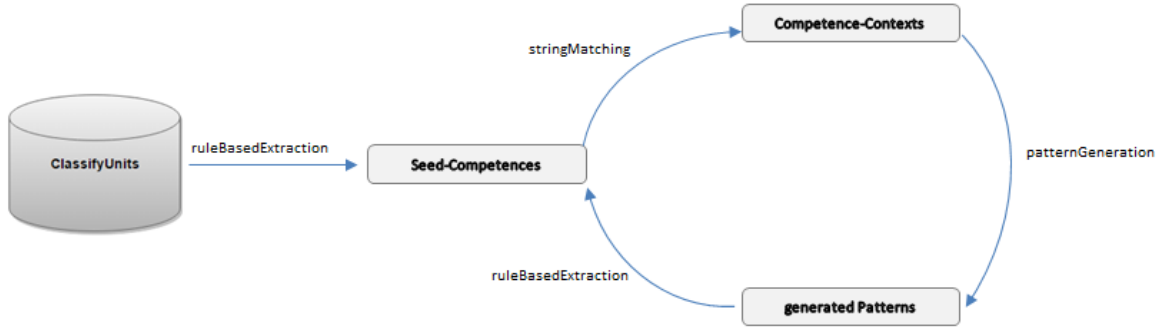


Abbildung 7: Schematische Darstellung des Bootstrapping Ansatzes zur Kompetenzextraktion

5.5 Evaluation

Zur Evaluation des kombinierten Verfahrens wurde dasselbe Trainingskorpus verwendet wie für die Evaluation des Knowledge Engineering Ansatzes. Die Ausgangsbedingungen für das Bootstrap Verfahren sind somit eine Precision von 97 % und ein Recall von 54%. Es wurden mehrere Durchläufe mit jeweils verschiedenen Kontextgrößen getestet.[17] In Tabelle 6 sind jeweils die besten Werte für Precision, Recall und den kombinierten F-Score fett hervorgehoben.

---

[16] Eine Textdatei mit allen automatisch generierten Patterns wird nach Durchführung des Workflows im mitgelieferten Software-Projekt gespeichert. (src/test/resources/information_extraction/output/autoPatterns.txt)

[17] Die entsprechende ausführbare Klasse befindet sich im mitgelieferten Software-Projekt. (src/test/java/information_extraction/evaluation/EvaluateBootstrapExtraction.java)

Kontextgröße		**Evaluationsmaße**			**Iterationen**
links	rechts	Precision	Recall	F1-Score	
1	2	0,485	**0,886**	0,627	9
2	1	0,555	0,856	0,673	6
2	2	0,757	0,777	0,767	5
2	3	0,924	0,768	**0,839**	5
3	2	0,757	0,742	0,75	4
3	3	0,934	0,731	0,82	4
3	4	**0,936**	0,731	0,821	4
4	3	**0,936**	0,725	0,817	4
4	4	**0,936**	0,725	0,817	4

Tabelle 6: Evaluationsergebnisse des Bootstrap-Verfahrens

Mit einer Kontextgröße von links zwei und rechts drei Tokens konnte der beste F-Score von ca. 84 % erzielt werden. Ähnlich gute Ergebnisse werden bei noch größeren Kontexten mit bis zu vier Tokens erreicht. Im Vergleich zum reinen Knowledge Engineering Ansatz konnte der Recall somit um mehr als 20 Prozentpunkte verbessert werden (23 beim besten F-Score-Wert), wobei die Precision um ca. 8 Prozentpunkte gesunken ist.

Zur Bewertung des Bootstrap-Verfahrens ist außerdem Tabelle 7 interessant, welche die Anzahl der jeweils vorgenommenen Extraktionen und tatsächlich unterschiedlichen Kompetenz-Types dokumentiert. Im Vergleich mit dem Knowledge Engineering Ansatz ist festzustellen, dass im Bootstrap-Verfahren nicht nur 68 neue Extraktionen getätigt wurden, sondern insgesamt 40 neue Ausdrücke identifiziert werden konnten. Auch abzüglich der 18 als falsch evaluierten Ausdrücke ist dies in dem begrenzen Testkorpus ein solider Zuwachs, der allein durch automatisch generierte Muster erzielt wurde.

	Testkorpus	**Knowledge-Engineering**	**Bootstrapping**
Annotationen/Extraktionen	447	325	393
Kompetenz-Types	271	184	224
False-Positive (Annotationen/Types)	-	7/6	38/18
True-Positive (Annotationen/Types)	-	273/146	355/206

Tabelle 7: Evaluationsstatistik: Anzahl der vorgenommenen Annotationen/ Extraktionen und unterschiedlicher Kompetenz-Types.

Auffällig ist jedoch die hohe Wiederholungsrate einzelner Fehler. 19 von den insgesamt 38 falsch-positiven Extraktionen gehen allein auf zwei Kompetenzausdrücke zurück. Der Ausdruck g*ute Kenntnisse* wurde in zehn und *Schrift* in neun Sätzen (z.B. in „Englisch in Wort und *Schrift*“) extrahiert. Durch das im Bootstrap-Verfahren angewandte Stringmatching können einzelne Falschextraktionen schnell vervielfacht werden, wenn sie im Gesamtkorpus hoch frequentiert auftreten. Übertragen auf ein sehr viel größeres Korpus können diese sehr schnell ausarten und die Präzision stark beeinträchtigen. Das für das Bootstrap-Verfahren typische Phänomen nennt sich *Semantic Drifting,* also semantisches Abdriften durch die iterative Wiederholung von Fehlern. Durch die falschen Extraktionen fließen auch falsche bzw. schlechte Patterns in den Prozess ein, die wiederum neue Falschextraktionen verursachen können. Für die Anwendung auf großen Korpora sollten daher, wie im *Snowball*-System, Evaluationsschritte zwischengeschaltet werden, die Patterns und Seeds bewerten und gegebenenfalls vorzeitig aussortieren. Als Indikator für die Güte eines neuen Patterns könnte beispielsweise die Zahl der damit aufdeckbaren bereits bekannten Kompetenzen sein. Ein Indikator für neue Kompetenz-Types die Zahl und Güte der produzierenden Patterns.

6 Fazit und Ausblick

Ziel der Arbeit war der Entwurf und die Implementation eines generischen Formalismus zur Musteridentifikation, der für die Extraktion domänenspezifischer Information eingesetzt werden kann. Resultat ist ein Workflow, in dem ein Set manuell codierter, linguistisch angereicherter Muster, durch einen Bootstrapping-Ansatz maschinell erweitert werden kann (siehe 6.1). Eingesetzt wird der Workflow im Rahmen eines Kooperationsprojektes der Spinfo mit dem BIBB zur Extraktion von Bewerberkompetenzen aus einem sehr großen Korpus von Stellenanzeigen (siehe 6.2). Die entwickelten Werkzeuge sind auf weitere Korpora mit Stellenanzeigen übertragbar und können auch zur Extraktion von Entitäten jenseits von Kompetenzen genutzt werden (im ausgeschriebenen Job eingesetzte Arbeitsmittel wurden bereits umgesetzt, Tätigkeiten stehen an). Darüber hinaus ist der Bootstrapping-Ansatz auch auf andere Domänen übertragbar (siehe 6.3).

6.1 Eingesetzte Mittel

Die gesetzten Ziele wurden im Verlaufe der Arbeit wie folgt umgesetzt: In Kapitel 1 wurden zunächst die methodischen Grundlagen der IE dargelegt und verschiedene Ansätze zur Entwicklung eines IE-Systems gegenübergestellt. Übertragen auf die Domäne der Stellenanzeigen und speziell der Bewerberkompetenzen konnte in Kapitel 2 eine Strategie für das methodische Vorgehen entwickelt werden. Zunächst wurde ein Knowledge Engineering Ansatz verfolgt, der auf der manuellen Formulierung von Extraktionspattern basiert. Hierfür wurde ein Regelformalismus entworfen, der die Referenzierung verschiedener linguistischer und struktureller Informationen ermöglicht. Für diesen Ansatz sprach die Beobachtung einiger oft wiederkehrender Muster in der Formulierung von Kompetenzen. Der manuelle Aufwand wird zudem durch die Möglichkeit der gezielten Anpassung der Patterns auf die jeweilige Domäne gerechtfertigt, was besonders bei Stellenanzeigen, die täglich in großer Zahl neu produziert werden und die in ihrer Gesamtheit wirtschaftlich wichtige und auswertbare Informationen enthalten, ein lohnenswerter einmaliger Einsatz ist. Von der hohen Präzision (97 %), die durch manuelle Musterbildung erreicht werden kann, profitiert auch das anschließend eingesetzte Bootstrap-Verfahren. In diesem iterativen Verfahren wurden die zuvor extrahierten Ausdrücke verwendet, um neue Kompetenzkontexte aufzudecken. Durch verschiedene Default-Mechanismen konnten diese Kontexte automatisch in neue Patterns übersetzt werden und einen neuen Extraktionsprozess in Gang setzen. Auf diese Weise wurde der Recall in jedem

Iterationsschritt erhöht, bis schließlich Werte von über 75 % erreicht werden konnten (77 % bei der Konfiguration mit bestem F-Score von 83 %). Gegenüber dem manuellen Verfahren ist dies eine Verbesserung von 23 Prozentpunkten. Die Ausbeute konnte also um fast 50 % gesteigert werden.

6.2 Nutzung

Derzeit wird der entwickelte Workflow mit den hier genannten Mustern im BIBB zur Extraktion von Kompetenzen eingesetzt. Eine Klassifikation der Stellenanzeigen in Sinnabschnitte wurde bereits für das gesamte Korpus umgesetzt, so dass die Kompetenzextraktion direkt auf den Paragraphen zum Bewerberprofil ansetzen kann. Zur Extraktion von Arbeitsmitteln wurde ein weiteres Musterset entwickelt, welches auf den Abschnitten zum Tätigkeitsprofil angewandt wird. Die Extraktion befindet sich derzeit noch in einer überwachten Lernphase, in der jeweils aus kleineren Auszügen der Datenbank extrahiert und die Ergebnisse von Mitarbeitern im BIBB gesichtet werden. Es wird z. B. eine Liste mit typischen, sich wiederholenden Extraktionsfehlern geführt, die als Negativbeispiele in den weiteren Workflow integriert werden. Typischer ‚Beifang' von eigentlich bewährten Patterns sind beispielsweise v*ergleichbares* (etwa in „gelernter Schlosser oder *Vergleichbares"*) oder *gennanter bereich* (etwa in „Erfahrung in den *genannten Bereichen"*. In Planung ist auch die Erprobung von automatischen Evaluationskomponenten für extrahierte Entitäten und Patterns, wie in Abschnitt 5.5 bereits erwähnt. Patterns werden dabei einen Confidence-Wert erhalten, der sich nach der Anzahl der bereits bestätigten Entitäten richtet, die mit Hilfe des Patterns identifiziert werden können. Analog dazu sollen die neuen Entitäten hinsichtlich der Anzahl und Güte der produzierenden Patterns bewertet werden. Zur Vermeidung von *Semantic Drifting* könnten dann Grenzwerte für die Aufnahme von Patterns und Entitäten festgesetzt werden.

Trotz der Extraktion möglichst kompakter und kanonischer Ausdrücke, sind im Anschluss der Extraktionsphase zudem einige Normalisierungsschritte notwendig, um beispielsweise *Führerschein, Fahrerlaubnis* und *fs*, oder *Flexibilität* und *flexibel* zu verbinden. Teilweise können auch hierfür die Kontexte zur Identifikation ähnlich- oder gleichbedeutender Entitäten herangezogen werden. Dazu bieten sich auch Distanzmaße wie die Levenshtein-Distanz an, etwa zur Erkennung von unterschiedlichen Schreibweisen (*power point – powerpoint, ms-office – ms office*) oder auch Schreibfehlern.

6.3 Adaptierbarkeit

Die in dieser Arbeit manuell codierten Muster sind nur für die Aufgabe, Kompetenzen innerhalb von deutschen Stellenanzeigen zu finden (eigentlich noch spezifischer: In den das Bewerberprofil enthaltenden Abschnitten aus Stellenanzeigen) nutzbar. Auch die im Bootstrapping-Verfahren automatisch ermittelten Muster sind spezifisch nur für diese Aufgabe verwendbar, da sie durch das initiale Seedset auf die Detektion von Kompetenzen geeicht wurden. Trotz dieser eingeschränkten Nutzungsmöglichkeit lohnt sich der manuelle Aufwand allein schon deshalb, weil Stellenausschreibungen aufgrund ihrer großen, täglich wachsenden Zahl und der aus ihrer Gesamtheit ableitbaren, wirtschaftlich hochrelevanten Arbeitsmarktdaten ein lohnendes Forschungsobjekt sind (s.u., Kooperationsanfragen).

Da aber Computerlinguistik auch immer darauf zielen sollte, nachhaltige und nachnutzbare Werkzeuge zu erschaffen, wurde bei der Implementierung Wert auf die Wiederverwertbarkeit des Workflows gelegt. So kann und wird dasselbe IE-System im BIBB nicht nur zur Extraktion von Kompetenzen, sondern auch zur Extraktion von Arbeitsmitteln angewendet. Hierfür ist lediglich ein Austausch der externen Textdatei notwendig, in der die Extraktionspatterns codiert sind. Zwar müssen diese für Arbeitsmittel neu formuliert werden, der Regelformalismus ist jedoch für die Codierung sämtlicher linguistisch formulierbarer Kontexte anwendbar und somit unabhängig von einer bestimmten Domäne oder Informationsanfrage. Selbiges gilt für das darauf aufsetzende Bootstrap-Verfahren, das auch isoliert – mit einem manuell annotierten Seedset – verwendet werden kann. Da zwar die Regeln, nicht aber der Regelformalismus selbst angepasst werden muss, sind das Bootstrap-Verfahren und insbesondere die automatische Mustergenerierung aus Beispielkontexten direkt adaptierbar.

Generelle Voraussetzung für ein auf Kontextpattern basierendes Extraktionssystem ist natürlich das Vorhandensein von regelhaften linguistisch ausdrückbaren Mustern. Bei Entitäten, die sich (wie eben Bewerberkompetenzen in Stellenausschreibungen) innerhalb einer Domäne häufig wiederholen (etwa *führerschein* oder *berufserfahrung)* genügt es jedoch, wenn nur ein Vorkommen durch Muster aufgedeckt wird. Die restlichen werden im nächsten Iterationsschritt via Stringmatching erkannt und als Vorlage für neue Patterns herangezogen. Wichtig ist dahingegen

aber ein gut eingegrenzter Suchraum, der in diesem Anwendungsfall durch die vorgeschaltete Abschnittsklassifikation erreicht wurde.[18]

Dass nicht nur das Verfahren, sondern auch die erzielten Ergebnisse nachnutzbar sind, zeigt das Interesse von gleich drei Firmen an den entwickelten Softwarekomponenten. Während das Bundesinstitut vor allem an einer statistischen Auswertung der extrahierten Informationen interessiert ist, um Trends zu erkennen, geht es Firmen wie textkernel[19], get-in-it[20] sowie meinestadt.de[21], die bereits alle Kontakt zu uns aufgenommen haben, darum, Bewerberprofile und Jobausschreibungen zu matchen, wozu sie genau die Information benötigen, deren Extraktion in dieser Arbeit beschrieben wurde.

[18] Kompetenzen tragen häufig nur im Kontext eines Bewerberprofils ihre Bedeutung als Kompetenz und können im Tätigkeitsprofil gleichzeitig z.B. als Arbeitsmittel oder Tätigkeit auftreten „Ihre Aufgabe: Führung und *Motivation* der Mitarbeiter“ vs. „Voraussetzungen: Hauptschulabschluss, *Motivation*, Einsatzbereitschaft und Zuverlässigkeit.“

[19] https://www.textkernel.com/de/

[20] https://www.get-in-it.de/

[21] http://www.meinestadt.de/

Literaturverzeichnis

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

Appelt, Douglas E. & David J. Israel (1999). *Introduction to Information Extraction Technology.* A Tutorial prepared for IJCAI-99. Menlo Parc: SRI-International. URL: http://www.dfki.de/~neumann/esslli04/reader/overview/IJCAI99.pdf (zuletzt aufgerufen am 03.04.2017).

Banko, Michele et al. (2007). 'Open Information Extractin from the Web'. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence,* Hyderabad, 2670 – 2676.

Bsiri, Sandra & Michaela Geierhos (2007). 'Informationsextraktion aus Stellenanzeigen im Internet'. In: *LWA 2007: Lernen – Wissen – Adaption, Workshop Proceedings.* Hrsg. Von Alexander Hinneburg. Halle: Martin-Luther-Universität Halle-Wittenberg, 229-236.

Califf, Mary Elaine (1998). *Relational Learning Techniques for Natural Language Information Extraction.* Dissertation.

DeJong, Gerald (1979). 'Prediction and Substantiation: A New Approach to Natural Language Processing'. In: *Cognitive Science* 3(3), 117-123.

Doorenbos, Robert B. et al. (1997). 'A Scalable Comparison-Shopping Agent for the World-Wide-Web'. In: *Proceedings of the first International Conference of Autonomous Agents.* Marina del Rey, 39-48.

Eikvil, Line (1999). *Information Extraction From World Wide Web. A Survey.* Technical Report 945. Oslo: Norwegian Computing Center.

Etzioni, Oren et al. (2005). 'Unsupervised Named-Entity Extraction from the Web: An Experimental Study'. In: *Artificial Intelligence* 165 (1), 91-134.

Geduldig, Alena (2015a). *Textklassifikation mit Support Vector Machines.* Hausarbeit.

Geduldig, Alena (2015b). *Evaluation des Suffixtree-Document-Models als Repräsentationsmodell zur Textklassifikation.* Hausarbeit.

Hermes, Jürgen & Manuel Schandock (2016). 'Stellenanzeigenanalyse in der Qualifikationsentwicklungsforschung. Die Nutzung maschineller Lernverfahren in der Klassifikation von Textabschnitten'. In: *Fachbeiträge im Internet.* Bundesinstitut für Berufsbildung. URL: https://www.bibb.de/veroeffentlichungen/de/publication/show/8146 (zuletzt aufgerufen am 04.04.2017).

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

Jurafsky, Martin & James H. Martin (2009). 'Information Extraction'. In: *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: Pearson Prentice Hall, 759-798.

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

MUC-3 (1991). *Proceedings of the 3th Conference on message understanding.* San Diego.

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, 576-615.

Neumann, Mandy (2015). *Analyse von Anforderungsprofilen. Eine Studie zur Informationsextraktion aus Stellenanzeigen.* Masterarbeit.

Piskorski, Jakub & Roman Yangarber (2013). 'Information Extraction: Past, Present and Future'. In: *Multi-source, Multilingual Information Extraction and Summarization.* Hrsg. von Thierry Poibeau et al., Berlin: Springer, 23-50.

Rajaraman, Anand (1998). 'Virtual Database Technologie, XML and the Evolution of the Web'. In: *Data Engineering 21(2).*

Sager, Naomi (1981). *Natural Language Information Processing: A Computer Grammar of English and Its Applications.* Boston: Addison-Wesley Longman Publishing Co., Inc.

Soderland, Stephen (1999). 'Learning Information Extraction Rules for Semi-Structured and Free Text'. In: *Machine Learning* 34. Hrsg. von Claire Cardie & Raymond Mooney, 233-272.

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

Wu, Fei & Daniel Weld (2010). 'Open Information Extraction using Wikipedia'. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* Uppsala, 118-127.

Selbstständigkeitserklärung

Hiermit versichere ich an Eides Statt, dass ich diese Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Die Stellen meiner Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken und Quellen, einschließlich der Quellen aus dem Internet, entnommen sind, habe ich in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht. Dasselbe gilt sinngemäß für Tabellen, Karten und Abbildungen.

Diese Arbeit habe ich in gleicher oder ähnlicher Form oder auszugsweise nicht im Rahmen einer anderen Prüfung eingereicht.

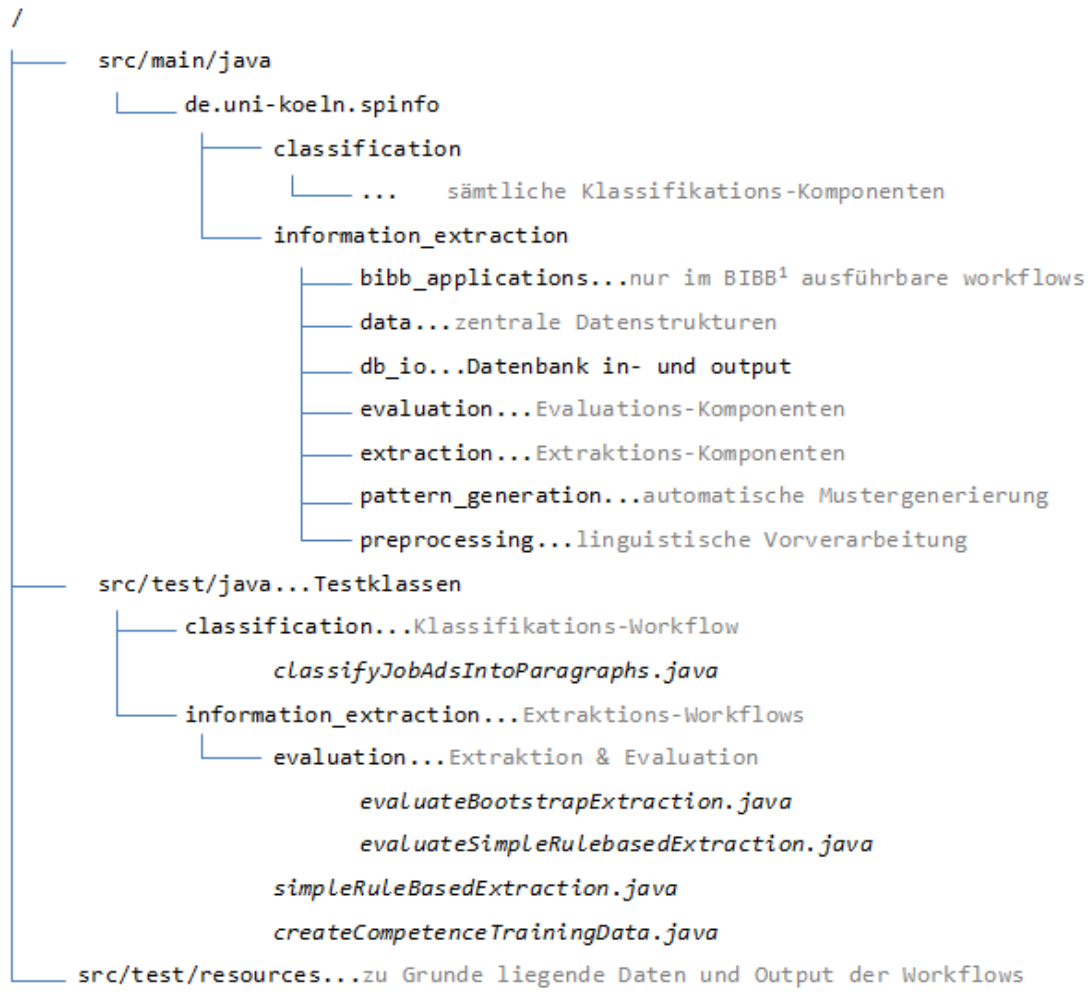
Ich versichere zudem, dass der Text der eingereichten elektronischen Fassung mit dem Text der vorgelegten Druckfassung identisch ist.

Köln, den 25. April 2017 Unterschrift_____

A Hinweise zur beiliegenden CD

Auf der mit dieser Arbeit abegegebenen CD befindet sich neben einer elektronischen Fassung dieses Dokuments auch der Quelltext aller zur Durchführung der beschriebenen Workflows benötigten Software-Komponenten. Das Programm liegt als Projektarchiv vor und kann wahlweise entpackt oder in eine IDE wie beispielsweise eclipse importiert werden. Alternativ kann das Projekt auch unter https://github.com/geduldia/JobAd_IE herunter geladen werden.[22]

Die Klassen und weiteren Dateien des Projekts sind in der folgenden Paketstruktur geordnet, welche die jeweilige Funktionalität wiederspiegeln soll:



[22] In diesem Fall müssen zur Ausführung der Workflows zwei zusätzliche Dateien hinzugefügt werden. Eine Anleitung hierfür ist im README-File zu finden.

Sämtliche ausführbaren und in dieser Arbeit beschriebenen Klassen liegen als JUnit-Testklassen vor und stellen vollständige Workflows dar.

Mit `classifyJobAdsIntoParagraphs` kann ein lokal vorliegender Auszug aus der Datenbank des BIBB klassifiziert werden. Die Ergebnisse werden als Datenbankfiles gespeichert (/test/resources/classification/output/).

`SimpleRulebasedExraction` verwendet diese als Input zur Kompetenzextraktion und speichert die Ergebnisse ebenfalls als Datenbankfile (test/resources/ information_extraction/output/).

Mit `CreateCompetenceTrainingData`, einem interaktiven Workflow zur Annotation von Kompetenzen, wurde das dieser Arbeit zu Grunde gelegte Testkorpus erstellt (test/resources/information_extraction/trainingdata/).

`EvaluateSimpleRulebasedExtraction` und `EvaluateBootstrapExtraction`, führen eine Extraktion mit dem jeweiligen Verfahren durch und evaluieren die Ergebnisse im Anschluss. Ausführliche Evaluationsergebnisse (inklusive aller richtig und falsch extrahieren Entitäten) werden als Textfiles gespeichert (…/information_extraction/ output/evaluation_files/). Bei der Evaluation des Bootstrapping-Ansatzes werden außerdem sämtliche automatisch generierten Patterns hinterlegt (…/information_extraction/output/).

Im Resource-Ordner befinden sich neben den in den Workflows erzeugten Dateien und Datenbanken auch sämtliche dafür benötigten Input-Daten inklusive aller manuell codierten Extraktions-Patterns.

B Abkürzungsverzeichnis

NLP Natural Language Processing

IE Information Extraction, Informationsextraktion

BIBB Bundesinstitut für Berufsbildung

Spinfo Sprachliche Informationsverarbeitung

IR Information Retrieval

HTML Hypertext Markup Language

LSP Linguistic String Project

FRUMO Fast Reading Understanding and Memory Program

MUC Message Understanding Conference

NER Named Entity Recognition

POS Part of Speech