Parsing von Stellenanzeigen zur Vorhersage von Bewerberqualifikation

Job Ad Parsing for Applicant Qualification Prediction

Master-Thesis von Tobias Krönke aus Frankfurt am Main

September 2013

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Informatik
Fachgebiet Sprachtechnologie

Parsing von Stellenanzeigen zur Vorhersage von Bewerberqualifikation
Job Ad Parsing for Applicant Qualification Prediction

Vorgelegte Master-Thesis von Tobias Krönke aus Frankfurt am Main

1. Gutachten: Chris Biemann
2. Gutachten: Martin Riedl

Tag der Einreichung:

Erklärung zur Master-Thesis

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 25. September 2013

(T. Krönke)

Zusammenfassung

Der Arbeitsvermittlungsmarkt lässt sich generell in die Bereiche der suchenden Unternehmen und Arbeitskräfte unterteilen. Der digital vernetzte Arbeitsvermittlungsmarkt generiert dabei große Mengen an veröffentlichten, semi-strukturierten Stellenanzeigen. Für beide Parteien ist es wegen der großen Datenmengen schwer, relevante Treffer bei der Suche zu erzielen.

Vor diesem Hintergrund beschreibt die vorliegende Arbeit den Entwicklungsansatz, die Kerninformationen von Stellenanzeigen zu extrahieren und gegebenenfalls zu normalisieren. Dafür wird mit dem hier vorgeschlagenen Tokenisierungsverfahren, den vorgestellten Features der natürlichen Sprachverarbeitung und den speziell für die Hypertext Markup Language (HTML) entwickelten Features ein Conditional Random Field als „Parser“ gelernt und evaluiert. Dies geschieht auf teils automatisch, teils überwacht getaggten Trainingsdaten. Darauf aufbauend wird ein Matching-System mit den Ansätzen aus dem Information Retrieval entwickelt. Es vergleicht die extrahierten Informationen mit strukturierten Bewerberprofilen in den jeweils passenden Feldern, um somit qualitativ hochwertige Jobvorschläge für suchende Arbeitskräfte zu generieren. Dabei werden auch die persönlichen Metakompetenzen („Talente“) sowohl als Anforderungen der Stellenanzeigen als auch als Eigenschaften der Arbeitskräfte berücksichtigt.

Der inhaltliche Fokus der geparsten Stellenanzeigen liegt dabei im Bereich der Informationstechnologie. Das Parsing liefert dennoch auch auf einem größeren, heterogenen Datensatz zufriedenstellende Ergebnisse. Indem das Matching ohne und mit den extrahierten Informationen verglichen wird, kann bei der Verwendung der extrahierten Informationen eine Verbesserung der Qualität der Vorschläge und ihrer Reihenfolge festgestellt werden.

Schlüsselbegriffe: Stellenanzeigen, Talente, Informationsextraktion, natürliche Sprachverarbeitung, Conditional Random Fields, Matching, Information Retrieval

Abstract

The field of employment exchange can be separated into two decisive groups of participants: searching companies and job seekers. Many semi-structured job ads are published for the digitally connected employment exchange market. Because of this big amount of data, it is difficult for both searching parties to find relevant hits.

Based on that, this thesis comprises an approach towards developing a solution for extracting and possibly normalizing the most valuable core information from job ads. For the purpose of this parsing idea, a conditional random field is trained with the here developed tokenization. The tokens are combined with the presented features from natural language processing and the developed features specifically designed for the Hypertext Markup Language (HTML). The model is trained and evaluated with both, automatically tagged and supervised labels. Subsequently, a matching solution with approaches from the field of information retrieval is developed. It aims for comparing the extracted information from job ads with structured candidate profiles within their matching fields of content. Its main goal is the proposal of high-quality job suggestions for job seekers. This solution also takes the personal competences or talents into consideration by comparing the job ads' respective requirements to the seekers' attributes.

In this thesis, job ads in the field of information technology will be analyzed. But the tagging also works well on a bigger, heterogeneous job ad corpus. By comparing the two implemented matching solutions, one without and one with the extracted information, it can be concluded, that the extracted information has a positive influence on the quality and and ranking of the proposed job ads.

Keywords: job ads, talents, information extraction, natural language processing, conditional random fields, matching, information retrieval

Inhaltsverzeichnis

Abbildungsverzeichnis **5**

Tabellenverzeichnis **6**

Algorithmenverzeichnis **6**

Abkürzungsverzeichnis **7**

Danksagung **8**

1. Einführung in die Thematik von Jobvorschlägen **9**
1.1. Motivation . . . 9
1.1.1. Forschung . . . 9
1.1.2. Umfeld des Arbeitsvermittlungsmarkts . . . 10
1.2. MeCruiting . . . 10
1.2.1. Qualitativ psychologischer Ansatz . . . 11
1.2.2. Aktuelle Vorgehensweisen der Rekrutierung . . . 12
1.3. Zielsetzung . . . 12
1.3.1. Datengrundlagen seitens der Unternehmung und des Arbeitssuchenden . . . 12
1.3.2. Überblick . . . 13

2. Verwandte Arbeiten zur Entwicklung des Parsers und des Vorschlagesystems **16**
2.1. Vorverarbeitung . . . 16
2.2. Informationsextraktion (IE) . . . 16
2.3. Information Retrieval (Informationsrückgewinnung) (IR) . . . 17
2.4. Data-Mining . . . 18

3. Verwendete Grundlagen aus dem Natural Language Processing (NLP) **19**
3.1. Notation . . . 19
3.1.1. Englische Begriffe . . . 19
3.2. Conditional Random Field (CRF) . . . 21
3.2.1. Das Problem . . . 21
3.2.2. Motivation . . . 21
3.2.3. Das Modell . . . 21
3.3. Information Retrieval (Informationsrückgewinnung) (IR) . . . 23
3.3.1. Theoretische Grundlagen . . . 23
3.3.2. Ausgewählte Systemdetails für Lucene und Solr . . . 24

4. Parsing von Stellenanzeigen **26**
4.1. Trainingsdaten sammeln . . . 26
4.1.1. Vorarbeit . . . 26
4.1.2. Entfernt überwacht getaggter Korpus . . . 27
4.1.3. Tags manuell vervollständigen . . . 27
4.1.4. Taggingrichtlinien . . . 31
4.1.5. Überblick über die Verteilung der wichtigsten Tags . . . 33
4.2. Implementierung des Parsers . . . 33
4.2.1. Tokenisierung als Beobachtungen . . . 33
4.2.2. Features der Beobachtungen (Tokens) . . . 36
4.2.3. Normalisierung . . . 39

4.3. Evaluation . . . 39
4.3.1. Entfernt überwacht getaggter Datensatz . . . 40
4.3.2. Manuell vervollständigter Korpus . . . 40
4.3.3. Ablation der visuellen Features . . . 42
4.3.4. Ablation der unüberwachten Part-of-Speech (POS)-Tags . . . 45

5. Vorhersage der Bewerberqualifikation **46**
5.1. Datengetriebener Entwicklungsprozess . . . 46
5.1.1. Überblick über die Rohdatengrundlage von Stellenanzeigen . . . 46
5.2. Erster naiver Prototyp . . . 47
5.3. IR anhand der extrahierten Informationen . . . 49
5.3.1. Überblick über den geparsten Korpus . . . 49
5.3.2. Normalisierung der extrahierten Informationen . . . 49
5.3.3. Umsetzung . . . 50
5.4. Evaluation . . . 52
5.4.1. Metriken zur Bewertung der IR-Systeme . . . 52
5.4.2. Erster naiver Prototyp . . . 52
5.4.3. Unterstützt durch extrahierte Informationen . . . 52
5.4.4. Vergleichende Analyse . . . 53

6. Fazit und Ausblick **54**
6.1. Fazit . . . 54
6.1.1. Parsing . . . 54
6.1.2. Matching . . . 54
6.2. Ausblick . . . 54
6.2.1. Parsing . . . 54
6.2.2. Matching . . . 55

Literaturverzeichnis **61**

Anhang **62**

Anhang A. URLs in Fußnoten **63**

Anhang B. Taggingrichtlinien **64**

Abbildungsverzeichnis

1.1. Klassische Stellenanzeige als Hypertext Markup Language (HTML)-Dokument. 10
1.2. Aktiver und passiver Recruitingprozess aus [Wei11]. 11
1.3. Problem der Qualifikationsvorhersage (aus dem EXIST Businessplan). 12
1.4. Talente / Kompetenzen bewertet von mit im sozialen Netzwerk verbundenen Personen (Screenshot der MeCruiting Anwendung `http://mecruiting.de` vom 23. September 2013.). 15
1.5. Gesamtvorstellung der Zielsetzung von MeCruiting (aus dem EXIST Businessplan). 15

3.1. Graphische Repräsentation des gerichteten Hidden Markov Model (HMM). Die X_i sind die Beobachtungen und Y_i die jeweiligen Zustände. [LMP01] 22
3.2. Faktorgraph des linear verketteten CRF. [LMP01] 23
3.3. Faktorgraph des linear verketteten CRF in HMM-Form. [LMP01] 23

4.1. Metadaten als überwachte Lerngrundlage. 27
4.2. Kumulatives Histogramm über (angezeigte) Tokens in den Stellen. 28
4.3. WebAnno [YGdCB13] „named entity“-Ebene. 29
4.4. Überblick über alle zu extrahierenden Felder. 30
4.5. Histogramm über die Verteilung der wichtigsten Tags im manuell vervollständigten Korpus. 34
4.6. Histogramme über die Länge der Labels in Tokens. 35
4.7. Unicode-Block der geometrischen Formen. Werden häufig als Aufzählungszeichen verwendet und sollen immer abgetrennt werden. 35
4.8. Verschiedene Satzzeichen und Aufzählungszeichen, die immer zu einer Trennung führen sollen. . . . 35
4.9. Anführungszeichen und Zeichen, die als solche missbraucht werden. 36
4.10. Zeichen, die lediglich innerhalb von Ziffern nicht trennen sollen. 37
4.11. Punktzeichen und Zeichen, die als solche missbraucht werden. 37
4.12. Lernkurven mit 10% Testdaten vom entfernt überwacht getaggten Korpus. 40
4.13. Lernkurven mit 10% Testdaten vom manuell getaggten Korpus. 41
4.14. Lernkurven mit 10% Testdaten vom manuell getaggten Korpus ohne visuelle Features. 42
4.15. Veränderung der durchschnittlichen F-Scores nach Ablation der visuellen Features. 43
4.16. Relativen Häufigkeiten der visuellen Features von als Ausbildung getaggten Tokens und den anderen. 44
4.17. Veränderung der durchschnittlichen F-Scores nach Ablation der unüberwachten POS-Tags. 45

5.1. Kumulatives Histogramm über (angezeigte) Tokens in den Stellen von cesar. 47
5.2. Jobvorschlag mit der Bitte um explizites Feedback. 48
5.3. Histogramm über die Verteilung der wichtigsten Tags im automatisch getaggten Korpus. 49
5.4. Sortierte Häufigkeiten von getaggten Talenten auf logarithmischen Achsen. 50

Tabellenverzeichnis

1.1. MeCruiting im Kontext heutiger Rekrutierungsverfahren (aus dem EXIST Businessplan). 12
1.2. Harmonisierung der wissenschaftlichen Datengrundlagen 13
1.3. Verwendete Norm von Talenten / Kompetenzen in vier Kategorien. 14

3.1. Konventionen zur einheitlichen mathematischen Notation. 19
3.2. Englische Fachbegriffe im Fließtext dieser Arbeit. 20

4.1. Vorkommen der semantischen Felder in 20 manuell getaggten Stellen. 26
4.2. Unterteilung von Stellenanzeigen in Boilerplate und relevanten, stellenbezogenen Inhalt beim Tagging. 31
4.3. Auszug aus den Richtlinien der wichtigsten Labels entsprechend ihrer englischen Benennung. 32
4.4. Absolute Häufigkeiten ganzer Spannen von Labels im manuell getaggten Datensatz von 1010 Stellen. 33

5.1. Die häufigsten Talente, die in mindestens 1% der Stellen vorkommen. 48
5.2. Beispiele zur Normalisierung von extrahierten Informationen. 51
5.3. Pro Zeile ein ExtendedDisMax wie in Abschnitt 3.3.2. 51
5.4. Normalized Discounted Cumulative Gain (NDCG)@k und Precision@k für den naiven Prototyp und 179 Queries bei jeweils 5 bewerteten Ergebnissen. 52
5.5. NDCG@k und Precision@k für die Umsetzung mit den extrahierten Feldern und 179 Queries bei jeweils 5 bewerteten Ergebnissen. 53

B.1. Auszug aus den Taggingrichtlinien der verbleibenden Labels mit Beispielen. 64

Algorithmenverzeichnis

4.1. Framework zur rückabwickelbaren Tokenisierung 36

Abkürzungsverzeichnis

AGG	Allgemeine Gleichbehandlungsgesetz
BIO	Begin-of/Inner-of/Out-of
CRF	Conditional Random Field
CSS	Cascading Style Sheets
CV	Curriculum Vitae
DOM	Document Object Model
$H\mathcal{LR}T$	Head-Left-Right-Tail
HMM	Hidden Markov Model
HR	Human Resources (Humankapital)
HTML	Hypertext Markup Language
IE	Informationsextraktion
IR	Information Retrieval (Informationsrückgewinnung)
MEMM	Maximum Entropy Markov Model
MLE	Maximum-Likelihood Estimation
NDCG	Normalized Discounted Cumulative Gain
NER	Named Entity Recognition
NLP	Natural Language Processing
POS	Part-of-Speech
PTB	Penn Treebank
URL	Uniform Resource Locator
XML	Extensible Markup Language

Danksagung

Ich möchte an dieser Stelle einigen Personen meinen Dank aussprechen. Der Unterstützung meiner Eltern verdanke ich ein sorgenfreies Studium. Mein Dank geht auch an meine Mitgründer für das manuelle Tagging und die weitere Unterstützung. Unseren Freunden gebührt Dank für das Nutzen der ersten Versionen, um weitere wichtige Daten zu gewinnen. Vielen Dank auch an meine Gutachter für meine intensive Betreuung. Ich danke Annika für das Korrekturlesen.

1 Einführung in die Thematik von Jobvorschlägen

Ziel dieser Arbeit ist es, ein Computersystem zu entwickeln, welches die Zusammenkunft von Arbeitskräfte suchenden Unternehmen und entsprechend qualifizierten Arbeitssuchenden vereinfacht. Hierfür muss der Computer in die Lage versetzt werden, Stellenanzeigen besser zu verstehen, um diese anschließend mit einheitlich strukturierten Profilen von potentiellen Bewerbern zu vergleichen. Gemachte Vorschläge sollen sowohl die Fähigkeiten und Aufgaben als auch die persönlichen Eigenschaften und Vorlieben beider Parteien berücksichtigen.

Zunächst soll die Aufgabe dieser Arbeit aus verschiedenen Blickwinkeln motiviert werden. Das System wird von und für die Gründer der Unternehmung MeCruiting entwickelt und dessen Ansätze in Abschnitt 1.2 in den Kontext der bislang üblichen Rekrutierungsvorgehen gesetzt. Abschnitt 1.3 gibt einen groben Überblick über die so vorgestellte Zielsetzung.

1.1 Motivation

Vor der relativ offensichtlichen monetären Begründung zum Zweck dieser Arbeit folgt eine kurze Motivation vom forschenden Standpunkt aus.

1.1.1 Forschung

In ihrer kürzlich veröffentlichten Arbeit zum „Online Recruiting System [...] iHR“ [HLLP13] stellen die Autoren die zwei sinngemäßen Kernfragen, denen auch diese Arbeit nachgeht:

1. Wie werden Nutzer des Systems und ihre Interessen repräsentiert und modelliert?
2. Wie können darauf aufbauend gute Vorschläge erzeugt werden?

Sie vergleichen hierzu verschiedene implementierte Systeme anhand der modellierten Nutzerzufriedenheit, gewonnen aus einem Umfrageprozess. Die Techniken dieser Systeme können vor allem in „inhaltlich basierte“, „wissensbasierte“ und verhaltenstechnische (mittels „colaborative filtering“) unterteilt werden. Erstere seien laut [HLLP13] die häufigsten, bei denen vor allem die eingegeben Rohdaten von Nutzern verglichen werden. Der wissensbasierte Ansatz geht einen Schritt weiter und versucht, diese Rohdaten in Ontologien einzugliedern. Diese Ontologien können beliebig komplex formuliert sein, um den rohen Begriffen eindeutige Bedeutungen, Eigenschaften und Beziehungen zueinander zuzuweisen.

Beim kollaborativen Filtern wird eine reellwertige Tabelle über Arbeitssuchende und Jobangebote erstellt. Jeder Eintrag in dieser Tabelle soll ausdrücken, wie sehr die jeweilige Stelle zur jeweiligen Person passt. Betrachtet man die Spalten und Zeilen als Vektoren, können z. B. mittels des Winkels zwischen den Vektoren Ähnlichkeitsbeziehungen zwischen den Personen respektive Jobs angegeben werden. Mit diesen Ähnlichkeitswerten können nun Vorhersagen zu unbekannten Werten in der Tabelle erzeugt werden. Damit wird das Verhalten von natürlichen Personen auf ein einfaches Vektormodell reduziert, mit der Annahme, dass sich ähnliche Personen für ähnliche Stellenangebote interessieren.

Viele der unter Abschnitt 2.2 vorgestellten Arbeiten evaluieren Algorithmen der Informationsextraktion (IE) auf Stellenanzeigen. Damit wird jedoch größtenteils nicht versucht, ein ganzheitliches System aufzubauen, das den Suchprozess im Arbeitsmarkt tatsächlich erleichtert. Viele der Arbeiten beschränken sich zusätzlich speziell auf Arbeitsangebote aus der Informatik. Dabei werden sehr domänenspezifische Informationen extrahiert wie z. B. benötigte Erfahrungen bestimmter Programmiersprachen. Das immense Vokabular vollständig allgemeiner Fähigkeitsanforderungen erschwert das Erstellen einer Ontologie. Verschiedene Schreibweisen und unbekannte implizite Anforderungen vermindern dabei den „Recall“ (siehe Tabelle 3.2). Es soll daher untersucht werden, inwieweit eine Normalisierung persönlicher Metakompetenzen („Talente“, siehe Tabelle 1.3) helfen kann, relevante Angebote zu finden.[1]

[1] Meist wird diese Arbeit bewerberseitig formuliert. Der umgekehrte Fall, dass Unternehmen oder Vermittler qualifizierte Arbeitssuchende für ihre Stellen finden, soll deshalb jedoch nicht ignoriert werden. Vielmehr wird die Symmetrie der beiden Suchprozesse genutzt, die Ansätze dieser Arbeit kürzer zu motivieren und zu entwickeln. Es ist jedoch wesentlich einfacher, Stellenangebote zu akquirieren. Deshalb soll vor allem dem Arbeitssuchenden eine relevante Liste von Jobvorschlägen angeboten werden.

1.1.2 Umfeld des Arbeitsvermittlungsmarkts

In der Bundesrepublik Deutschland beträgt die Größe des Gesamtarbeitsmarktes über € 18 Mrd., globale Schätzungen belaufen sich auf ein Recruitinggesamtmarktvolumen von ca. $ 500 Mrd [ANA10]. Trotzdem verhält sich der Markt konservativ gegenüber neuen Technologien. Stellenanzeigen werden weiterhin in großer Zahl in klassischer Darstellungsweise, wie beispielhaft in Abbildung 1.1 zu sehen, digital veröffentlicht.

FPGA Entwickler (m/w)

Nutzen Sie Ihre Kompetenz, um Engineering-Projekte mitzugestalten, sich interessanten Herausforderungen zu stellen und Zukunftsentwicklungen anzustoßen.

Ihre Aufgaben

- Enge Zusammenarbeit mit dem FPGA Design-Ingenieur und Übernahme der FPGA Simulations und Prüfaufgaben
- Entwickeln eines geeigneten Prüf- und Simulationskonzeptes für den FPGA Anteil
- Erstellen von Prüfvorschriften in den unterschiedlichen Entwicklungsphasen
- Erstellen einer Testbench zur vollständigen Simulation der FPGA Funktionalität
- Durchführung und Dokumentation der Simulationen und Prüfungen

Ihr Profil

- Gute abgeschlossenes Studium in Industrie-Elektronik, Nachrichtentechnik oder vergleichbare Fachrichtung
- Sehr gute Kenntnisse in HDL/VHDL
- Gute Kenntnisse in der Programmierung von FPGA's mit Signalverarbeitungsfunktionen
- Sehr gute Kenntnisse der Mentor Graphics Tools (Mentor-Author, -Designer, -ModelSim) sowie der XILINX FPGA Bausteine der Serie Spartan und Virex
- Zuverlässigkeit und Teamfähigkeit sind ebenso wichtige Eigenschaften

Wenn Technik Ihre Leidenschaft ist und Sie auf der Suche nach neuen Grenzen sind, wenn Sie wissen, dass Sie mehr können, und Sie Ihre Ideen von der Technik der Zukunft heute verwirklichen wollen - dann ist in unserem Team der richtige Platz für Sie.

Bei Fragen steht Ihnen Herr Marcus Will gerne zur Verfügung.

euro engineering
creating future

Ihr direkter Weg zu uns
Marcus Will
euro engineering AG
Lise-Meitner-Str. 15
89081 Ulm
Tel +49 (0)731-93565-0
Marcus.Will@ee-ag.com
www.ee-ag.com

Direkt bewerben
Drucken
Weiterempfehlen

top ARBEITGEBER 2013 INGENIEURE
CERTIFIED BY THE CRF INSTITUTE

Abbildung 1.1.: Klassische Stellenanzeige als Hypertext Markup Language (HTML)-Dokument (`http://stepstone.welt.de/`).

1.2 MeCruiting

Die Gründungsvorgeschichte von MeCruiting lässt sich wie folgt skizzieren. An den Professuren für Arbeits- und Organisationspsychologie von Prof. Dr. Gudela Grote des Instituts Management, Technologie und Ökonomie (ETH) und Human Resource Management von Prof. Dr. Bruno Staffelbach des Instituts für Betriebswirtschaftslehre an der Universität Zürich wird im Zuge einer Masterarbeit der Grundstein von MeCruiting gelegt. Darauf aufbauend wird die Unternehmung im November 2012 in das EXIST-Gründerstipendium Förderprogramm des Bundesministerium für Wirtschaft und Technologie aufgenommen. Neben dem Autor sind Clemens Dittrich und Matthes Dohmeyer die Gründer von MeCruiting.

1.2.1 Qualitativ psychologischer Ansatz

Abbildung 1.2 beschreibt den allgemeinen Recruitingprozess zwischen Stellensuchendem und stellenausschreibender Unternehmung plakativ.

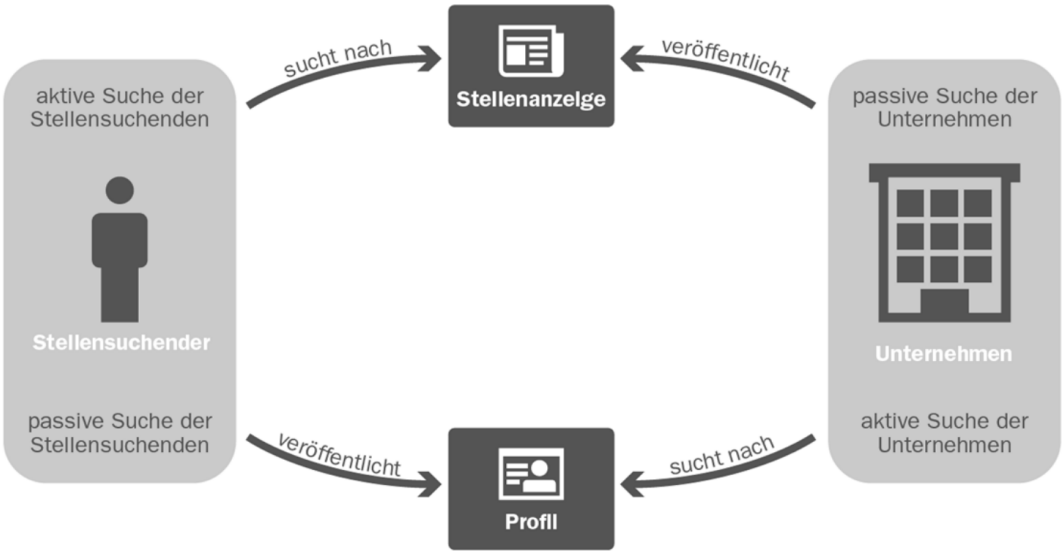

Abbildung 1.2.: Aktiver und passiver Recruitingprozess aus [Wei11].

Stellensuchender

Das Persönlichkeitsprofil eines Stellensuchenden lässt sich in soziodemographische Variablen und verschiedene Persönlichkeitsmodelle kategorisieren [AS02]. Soziodemographische Variablen werden dabei häufig als sogenannte „hard facts“ definiert, wohingegen psychologische Persönlichkeitsmodelle den „soft facts“ zugewiesen werden. Als Beispiele können hierbei die Boundaryless Career [AHL89], Protean Career [BH06], Career Preferences [GC04], Karriereanker [Sch78] oder Big Five-Persönlichkeitsfaktoren [CM03] genannt werden. Des Weiteren lassen sich im Recruitingprozess Fähigkeiten und Kenntnisse in „hard technical skills“ und „soft behavioral skills“ gliedern [AS02].

Stellenausschreibende Unternehmung

Die Determinanten für den qualitativen Personalbedarf einer Unternehmung lassen sich in Aufgaben und Anforderungen der zu besetzenden Stellen unterteilen [Rid99]. Die Spezifikation der Anforderung lässt sich folgendermaßen beschreiben:

Anforderungskriterien:

- Relevanz: Anforderungen müssen für die zu besetzende Stelle erforderlich sein, um sie ausfüllen zu können.
- Vollständigkeit: Alle erforderlichen Anforderungen müssen angegeben werden.
- Überschneidungsfreiheit: Anforderungen sollen sich nicht überlappen.

Diese Kriterien können wie folgt charakterisiert werden:

- Objektivität: Unabhängige Nachweisbarkeit
- Einfachheit: Qualität leicht einzuschätzen
- Reliabilität: Zuverlässigkeit in der Einschätzung
- Validität: Qualität der Einschätzung
- Effizienz: Abwägung zwischen Anzahl (Kosten) und Nutzen

1.2.2 Aktuelle Vorgehensweisen der Rekrutierung

Im Bereich der Personalrekrutierung, respektive Personalvermittlung, sind die Möglichkeiten der Vorgehensweise vielfältig. Dies mag den Tatsachen geschuldet sein, dass zum einen strategisches Personalmanagement stetig an Bedeutung gewinnt [BK99], zum anderen die Dringlichkeit, geeignetes Personal bei sinkender Personalverfügbarkeit zu finden, in den letzten Jahren zunimmt [KP09].

Die heutigen Rekrutierungsverfahren und deren kommerzielle Lösungen im Vergleich zu MeCruiting lassen sich in diverse Kategorien und Ebenen wie in Tabelle 1.1 unterteilen.

	Qualitativer Ansatz (manuelles Matching, technologieunterstützt)	Quantitativer Ansatz (automatisiertes Matching, technologiegetrieben)
Qualität IE / Matching niedrig		„klassische“ (Meta-)Stellenbörsen
Qualität IE / Matching hoch	Personalberater / Headhunter	MeCruiting, semantische Jobsuchmaschinen wie Jobolizer (`http://jobolizer.joinvision.com/`)

Tabelle 1.1.: MeCruiting im Kontext heutiger Rekrutierungsverfahren (aus dem EXIST Businessplan).

1.3 Zielsetzung

Die skizzierte Abbildung 1.3 des Recruiting-Prozessflusses illustriert, wie die Unternehmung und der Bewerber miteinander auf Basis ihrer Merkmale zusammengeführt werden. Voraussetzung dabei ist die jeweilige Kenntnis des verfügbaren Bewerbers beziehungsweise der verfügbaren Stellenausschreibung.

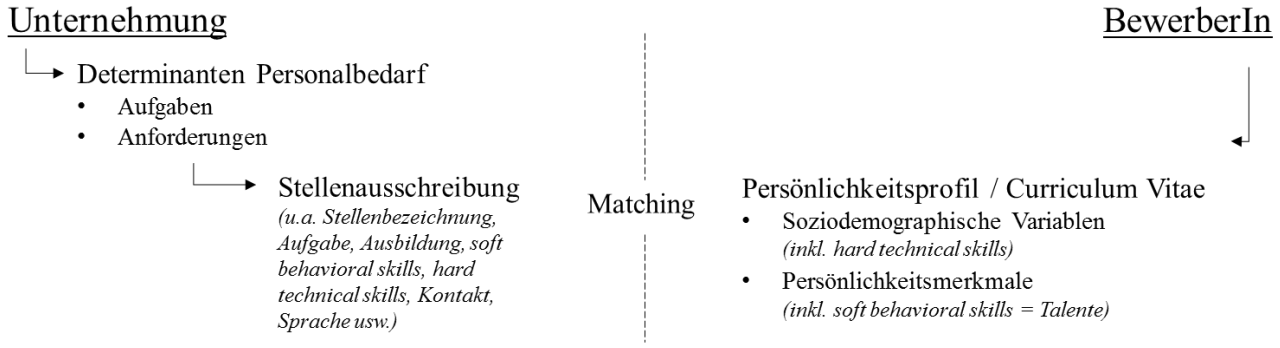

Abbildung 1.3.: Problem der Qualifikationsvorhersage (aus dem EXIST Businessplan).

1.3.1 Datengrundlagen seitens der Unternehmung und des Arbeitssuchenden

Eine Stellenanzeige kann in drei Teile gegliedert werden:

- Aufgabenbeschreibung,
- Anforderungen der ausgeschriebenen Stelle und
- Administratives

Wie bereits erwähnt, lassen sich die Anforderungen der ausgeschriebenen Stellen in zwei generelle Kategorien unterteilen, soziodemographische Variablen sowie Persönlichkeitsmodelle. Diese Unterscheidung ist eindeutig trennscharf [EvR03]. Eine weitere Definition der Anforderung einer ausgeschriebenen Stelle lässt sich durch den Kompetenzbegriff im Kontext des beruflichen Handelns finden [SSR04]. „Neben den fachlich-funktionalen“ Anforderungen, welche den soziodemographischen Variablen gleichzusetzen sind, fließen in die Anforderung einer ausgeschriebenen Stelle „die sozialen, motivationalen und emotionalen Aspekte menschlichen Arbeitshandelns“ mit

Wissenschaft	EuropassCV	Verwendet als / erweitert um
Aufgabenbeschreibung	Desired employment / Occupational field	
Anforderung		
Fachkompetenz		
Ausbildung	Work Experience	Ausbildung & Studium
Arbeitserfahrung	Education and training	Berufserfahrung
Fähigkeit	Personal skills and competences – Other parts	Fähigkeiten
Sprachkenntnis	Personal skills and competences – Language	Sprachen
		Ehrenamtliche Tätigkeiten
		Zusätzliche Informationen
Methodenkompetenz	Personal skills and competences – Other parts	Kompetenzen / Talente
Sozialkompetenz	Personal skills and competences – Other parts	Kompetenzen / Talente
Personalkompetenz	Personal skills and competences – Other parts	Kompetenzen / Talente
Administratives		
Kontakt (Auszug)	Personal Information (Auszug)	Angaben zur Person (Auszug)
Ort	City	Ort
Gehalt	Salary	
Start / Ende	Start end / date	
Referenz	Reference	
Nationalität	Nationality	Nationalität

Tabelle 1.2.: Harmonisierung der wissenschaftlichen Datengrundlagen mit denen des EuropassCV-Standards zur pragmatischen Anwendbarkeit in dieser Arbeit.

ein [SSR04]. Die Autoren von [SS99] sprechen von einer Etablierung der Unterteilung in vier Bereiche im Human Resources (Humankapital) (HR)-Kontext: Fachkompetenz, Methodenkompetenz, Personalkompetenz und Sozialkompetenz. Die Unterteilung in Tabelle 1.2 beschreibt die Harmonisierung der wissenschaftlichen Fundierung mit dem HR-XML[2]- bzw. EuropassCV[3]-Modell, sowie die pragmatische Verwendung bei dieser Arbeit:

Eine Harmonisierung bzw. Zusammenführung der Methoden-, Sozial und Personalkompetenz wird vorgenommen und dem Nutzer als selbst wählbare „Talente“ zur Auswahl vorgeschlagen. Die Gründe dieser zusammenfassenden Maßnahme sind vielfältig. Zum einen existieren keine systematischen Instrumente und Analysemethoden der Kompetenzmessung [SSR04], zum anderen ist bereits die Definition der verfügbaren Kompetenzen und der damit einhergehenden Kategorisierung in die einzelnen Kompetenzfelder nicht einheitlich definiert [EvR03]. Auch widerspricht eine granulare Aufgliederung der einzelnen Kompetenzen aus Sicht eines Nutzers jeglichem Vereinfachungsgedanken. Es zeigt sich außerdem, dass veröffentlichte Stellenanzeigen selbst in der Regel keine derartige Kompetenzunterscheidung beinhalten. Bei qualitativer Untersuchung von Stellenanzeigen wird zudem sichtbar, dass besonders kommunikative Aspekte häufig im Fokus stehen.

Es wird daher eine Norm (Tabelle 1.3) entwickelt, welche sich in vier Kategorien unterteilen lässt.

Die einzelnen Kompetenzen resultieren dabei aus der Extraktion verschiedener Literatur (vgl. [SSR04] [FR92] [Ras06] [AS01] [EB05]) mit anschließender manueller Kompetenzzuweisung der jeweiligen Kategorie.

Die selbst gewählten Talente können von den im sozialen Netz verbundenen Nutzern bewertet werden, um eine gewisse Fremd-Validierung zu erhalten. Eine solche Bewertungsübersicht wird exemplarisch in Abbildung 1.4 dargestellt. Die so gewonnenen Werte werden jedoch noch nicht ausgewertet oder weiterverarbeitet.

1.3.2 Überblick

Abbildung 1.5 fasst abschließend die geplante Zielsetzung zusammen, deren Konzeption weitgehend mit dieser Arbeit bis einschließlich des Information Retrieval (Informationsrückgewinnung) (IR) begleitet wird.

[2] http://www.hr-xml.org/

[3] http://europass.cedefop.europa.eu/

Talente / Kompetenzen			
Kommunikativ	Methodik	Persönlich	Sozial
Kommunikation	Analytisches Denken	Administratives Geschick	Didaktik
Beratungsorientierung	Auffassungsgabe	Anpassungsfähigkeit	Durchsetzungsvermögen
Diplomatisches Geschick	Detailgenauigkeit	Ausdauer	Einfühlungsvermögen
Emotionale Intelligenz	Ehrgeiz	Belastbarkeit	Frustrationstoleranz
Kundenorientierung	Entscheidungsstärke	Disziplin	Integrationsfähigkeit
Moderation	Flexibilität	Dynamische Persönlichkeit	Interkulturelle Kompetenz
Networking	Führungsstärke	Eigeninitiative	Konfliktlösung
Präsentation	Konzeptionelles Denken	Einsatzfreude	Kontaktfreude
Rhetorik	Kreativität	Engagement	Koordination
Selbstvermarktung	Lernfähigkeit	Ergebnisorientierung	Kritikfähigkeit
Serviceorientierung	Logisches Denken	Frustrationstoleranz	Liebevolles Wesen
Stilvolles Auftreten	Multitasking	Geduld	Organisation
Überzeugungskraft	Problemlösung	Handwerkliches Geschick	Professionelles Auftreten
Verhandlungsgeschick	Projektmanagement	Improvisationstalent	Reflexionsfähigkeit
Verkaufsorientierung	Prozessmanagement	Leistungsbereitschaft	Sicheres Auftreten
	Selbstständigkeit	Loyalität	Souveränität
	Stressresistenz	Motivationsvermögen	Teamfähigkeit
	Strukturiertes Arbeiten	Offenheit	Toleranz
	Strukturiertes Denken	Praxisorientierung	Umgangsformen
	Systematisches Handeln	Selbstbewusstsein	
	Unternehmergeist	Sorgfalt	
	Zahlenaffinität	Verantwortungsbewusstsein	
	Zeitmanagement	Verlässlichkeit	
	Zielorientierung	Zuverlässigkeit	

Tabelle 1.3.: Verwendete Norm von Talenten / Kompetenzen in vier Kategorien.

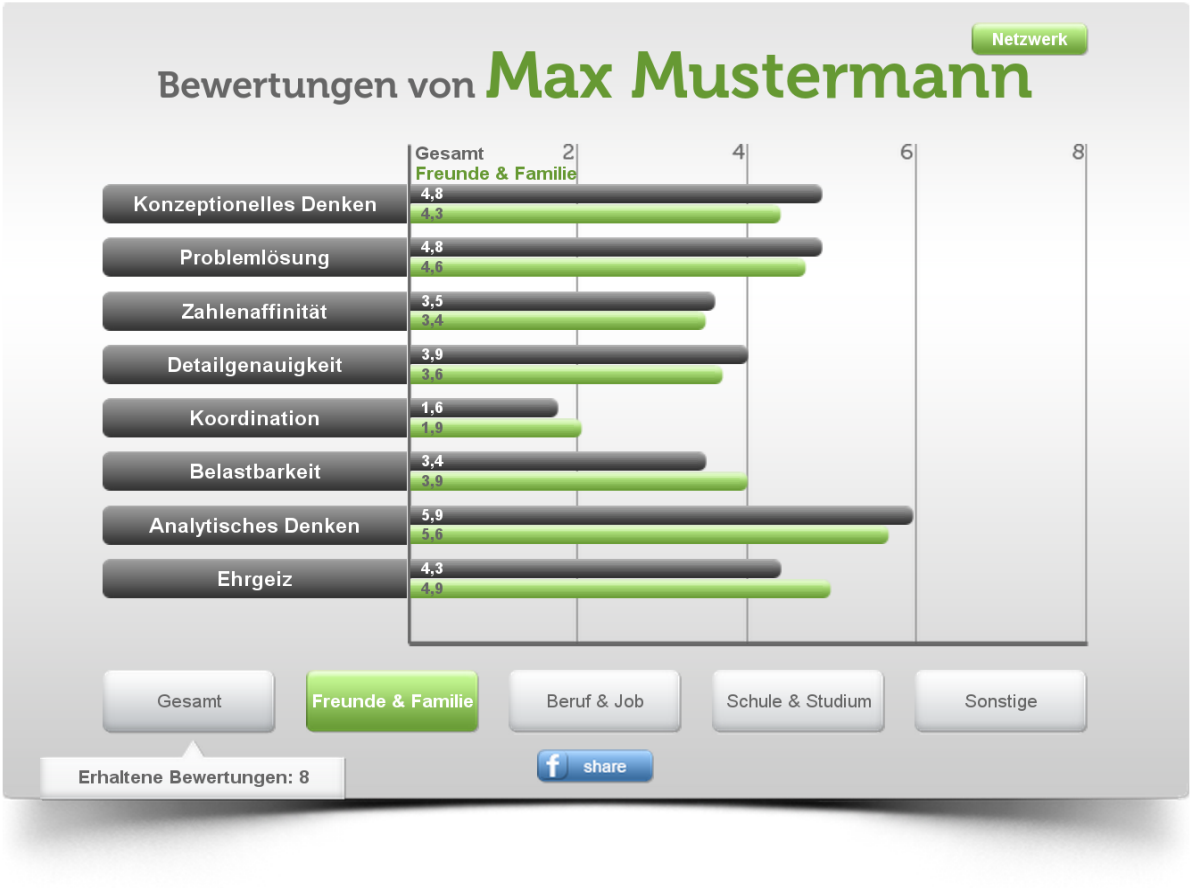

Abbildung 1.4.: Talente / Kompetenzen bewertet von mit im sozialen Netzwerk verbundenen Personen (Screenshot der MeCruiting Anwendung `http://mecruiting.de` vom 23. September 2013.).

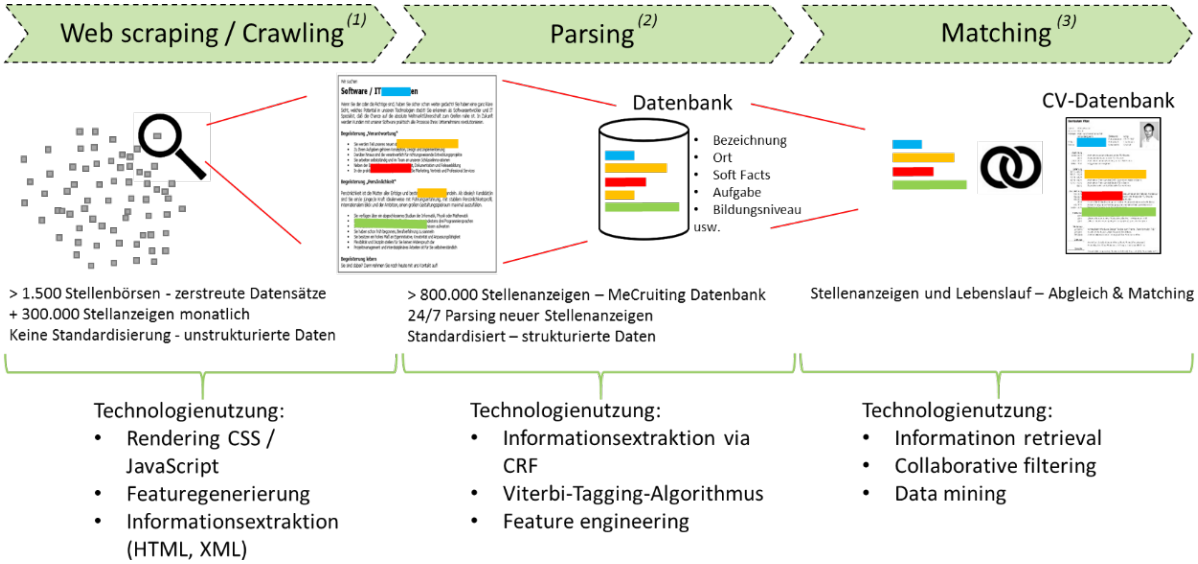

Abbildung 1.5.: Gesamtvorstellung der Zielsetzung von MeCruiting (aus dem EXIST Businessplan).

2 Verwandte Arbeiten zur Entwicklung des Parsers und des Vorschlagesystems

Abschnitt 2.1 stellt im Rahmen des Natural Language Processing (NLP) allgemein nützliche Quellen für diese Arbeit vor. In Abschnitt 2.2 geht es speziell um Arbeiten zur IE, die ihre Methoden auch auf Stellenanzeigen anwenden. Abschnitt 2.3 befasst sich mit dem IR, dessen Ansätze bei der Qualifikationsvorhersage zum Einsatz kommen. Abschließend werden in Abschnitt 2.4 Ideen aus dem Data-Mining präsentiert, mit deren Hilfe sich die Vorhersageergebnisse unter Umständen verbessern lassen.

2.1 Vorverarbeitung

In [DO12] wird erneut das „lange als gelöst geglaubte Problem“ der Tokenisierung aufgegriffen. Die Autoren kritisieren, dass Tokenisierung nicht nur das Auftrennen von Text in sinnvolle Textbausteine umfasst. Vielmehr sei auch eine Vorverarbeitung auf der untersten Ebene der Zeichen notwendig. Insbesondere für verschiedene Anwendungsgebiete und die weitere Verarbeitung und Verwendung der Tokens kann eine individuelle Tokenisierung Vorteile mit sich bringen. Die Autoren sprechen hierbei konkret Probleme mit neuen Kodierungen wie Unicode an. Damit wird das Problem unmittelbar relevant für das Tagging von Text in Stellenanzeigen im HTML-Format.

In Abschnitt 4.2.1 wird die beim Tagging verwendete Erweiterung zur Penn Treebank (PTB)[1]-Tokenisierung[2] vorgestellt. Mit der Ausnahme von „Whitespace“ lässt sich diese vollständig rückabwickeln und im Nachhinein wieder verändern.

In Anbetracht der großen Anzahl manuell zu erstellender Trainingsdaten für das Parsing und die Vorhersage der Bewerberqualifikation stellt Crowdsourcing eine gute Möglichkeit dar, den Arbeitsaufwand zu verteilen. Arbeiten wie [Glo13] zeigen, dass valide Daten in der Crowd gewonnen werden können. Auch WebAnno [YGdCB13], eine Webanwendung zum verteilten manuellen Tagging, befindet sich in der Entwicklung zur Anbindung an den Dienst CrowdFlower[3].

2.2 Informationsextraktion (IE)

Der Artikel [McC05] beschreibt IE als „Destillation strukturierter Daten aus unstrukturiertem Text“. Ihr vorangestellt ist die Sammlung der Textdokumente, auf die IE angewandt werden soll. Mittels Segmentierung werden die Wortgrenzen gefunden, deren beinhaltete Wörter in Datenbankfelder gespeichert werden sollen. Anschließend wird bei der Klassifizierung ermittelt, welche Segmente welchen Feldern (häufig „Slots“ von „Templates“ genannt) zugeordnet werden sollen. Bei der Assoziation wird nun versucht, Relationen zwischen den Feldern zu ziehen, um herauszufinden, welche zusammengehören. Die Segmente (oder auch Entitäten) der Relationen können dabei sinngemäß identisch oder verschieden sein. Es folgt der Prozess der Normalisierung, bei dem die Entitäten standardisiert werden, um Vergleiche zu vereinfachen oder gar erst zu ermöglichen. Ein gutes Beispiel hierfür ist die Umsetzung von Zeitangaben in ein einheitliches Zeitformat. Abschließend können im Schritt der Deduplizierung redundante Einträge identifiziert und entfernt werden. Die so gewonnenen Informationen werden in einer Datenbank gespeichert und stehen nun weiteren Verarbeitungsschritten wie dem Data-Mining zur Verfügung, um „anwendbares Wissen“ [McC05] zu gewinnen.

Im einfachsten Fall ist ein Text syntaktisch so strukturiert, dass die zu extrahierenden Informationen mit stets perfekten Regeln über feste Begrenzungen der Segmente gewonnen werden können. Solche Dokumente nennt man „Head-Left-Right-Tail ($H\mathcal{LRT}$)“-konform [SCM99]. Hierbei begrenzen H und T den zu untersuchenden Teil eines Dokuments. $\mathcal{LR}$ sind die K linken und rechten Grenzpaare $\{(l_k, r_k)\}$, deren beinhaltete Worte (Segmente) einem bestimmten Slot zugewiesen werden. In [KWD97] werden diese sogenannten „Wrapper“ formal vorgestellt

[1] http://www.cis.upenn.edu/~treebank/

[2] http://www.cis.upenn.edu/~treebank/tokenization.html

[3] http://crowdflower.com/

und mittels Induktion gelernt. Wegen des niedrigen Recalls bei nicht so stark strukturierten Dokumenten, wird das Wrapper-Verfahren in [FK00] mittels „Boosting“ [Sch99] verbessert.

WHISK [SCM99] ist ein System, das Slots mit Hilfe von regulären Ausdrücken füllt. Es funktioniert besonders gut auf mindestens semi-strukturierten Texten, bei denen der Kontext der zu extrahierenden Segmente der Syntax wegen sehr einheitlich ist. Hier reichen oft wenige Trainingsbeispiele, um perfekte Regeln statisch aufgebauter Webseiten zu lernen. Problemen bei Freitext kann man mit syntaktischen oder gar semantischen Taggern als Vorverarbeitungsschritt entgegenwirken. Die Arbeit [SCM99] gibt einen guten Überblick über weitere regelbasierte IE-Systeme und ihre Unterschiede. Im Bereich der auf semi-strukturiertem Text operierenden Systeme sind darüber hinaus noch SRV [Fre98] und RAPIER [CM99] zu nennen. Die regelbasierten Verfahren werden in [Cir00] [Cir01a] [Cir01b] weiter verbessert, indem Beginn und Ende von Segmenten unabhängig voneinander und Verbesserungsregeln typischer Fehler gelernt werden.

Betrachtet man natürlichen Text als Sequenzen von Wörtern bestimmter Klassen, lässt sich das Tagging der IE als stochastischer Durchlauf durch ein Hidden Markov Model (HMM) betrachten. In [FM00] wird gezeigt, wie Strukturen dieser graphischen Modelle gelernt werden können. Die Experimente, insbesondere mit Stellenanzeigen aus dem Usenet, scheinen einen generellen Vorteil den regelbasierten Verfahren gegenüber nachzuweisen.

Arbeiten wie [WW07] und [LBC+10] konzentrieren sich gezielt auf Stellenanzeigen. Besonders der Zwischenstand [LBC+10] der Dissertation zum SIRE-Projekt ist relevant wegen der weiteren verfolgten Ziele eines vollständigen Systems:

- Crawling von Stellenanzeigen
- Boilerplate entfernen
- Detailliertere Slots
- Aufbau einer Ontologie aus den extrahierten Informationen
- Kategorisierung
- Zugänglichkeit über einen Index

Bei HTML handelt es sich nicht nur um semi-strukturierten Text.[4] Visuelle Merkmale, u.a. in Form von Cascading Style Sheets (CSS)-Deklarationen, sind ebenfalls Bestandteil dieser Dokumente. Die Arbeit von [LCY08] zeigt, dass das Erscheinungsbild einer Website die Genre-Klassifizierung verbessern kann. Das „Erscheinungsbild“ wird hierbei über diverse Statistiken erfasst, z. B. Flächengrößen, Anzahl Bilder und Platzierungen von Links. Bei der IE sollen andere visuelle Auffälligkeiten das Tagging ebenfalls verbessern.

Nicht nur Crowdsourcing kann dazu verwendet werden, möglichst viele Trainingsdaten zu generieren. Die Autoren von [MBSJ09] extrahieren und lernen Relationen zwischen Entitäten „entfernt überwacht“. Für eine beliebige, binäre Relation aus der Freebase[5] Datenbank sammeln sie möglichst viele unverarbeitete Sätze, die beide Entitäten enthalten. Mit der Annahme, dass diese Sätze wahrscheinlich die zugrunde liegende Relation abbilden, werden sie als überwacht gelabelte Features betrachtet. Metadaten von Metajobsuchmaschinen sollen ebenfalls als Grundstock für Trainingsdaten des Taggingmodells dienen.

2.3 Information Retrieval (Informationsrückgewinnung) (IR)

Um die Bewerberqualifikation mittels Reihungsfunktionen als IR-Problem zu modellieren, dient [MRS08] als Einführung in die Thematik als Nachschlagewerk.

Möchte man verhindern, dass nur schlechte Vorschläge aufgrund ihrer Ähnlichkeit untereinander zurückgegeben werden, kann mittels Diversifikation von Ergebnismengen [HMTK12] ein breiteres Spektrum von Dokumenten in die Systemantwort miteinbezogen werden. Vergleichbar dazu finden die Autoren von [XWD+12] heraus, dass die Wahrscheinlichkeit, auf eine Werbeanzeige zu klicken, sinkt, wenn die umliegenden Anzeigen zu ähnlich sind. Die „Klickvorhersage“ ist damit nicht unabhängig von den umliegenden Angeboten.

[4] Das W3C rät zu einer strikten Trennung von Struktur und Aussehen (`http://www.w3.org/TR/WCAG20-TECHS/G140.html`).

[5] `http://www.freebase.com/`

2.4 Data-Mining

Ein bereits zu einem gewissen Grad funktionierendes System, das erfolgreich bestimmte Informationen extrahiert, kann mit Ideen aus dem Data-Mining weiter verbessert werden. Das in [MB05], [Nah04], [NM00] und [MN03] vorgestellte DISCOTEX wird wiederholt auf den extrahierten Informationen von 600 Informatik-Stellenanzeigen ausgeführt. Mit Hilfe manueller, anwendungsspezifischer Synonym-Wörterbücher werden ausgelesene Betriebssysteme, Computerprogramme, Fachgebiete und Programmiersprachen dedupliziert und Regeln daraus abgeleitet.

Um ebenfalls mit weniger expliziten Informationen über Nutzer oder die Stellenanzeigen bessere Ergebnisse anzuzeigen, kann kollaboratives Filtern helfen. Man modelliert die Annahme, dass ähnliche Nutzer sich für ähnliche Stellen interessieren (nutzerbasiert), beziehungsweise, dass zur aktuell betrachteten Stelle ähnliche Angebote ebenfalls interessant sind (artikelbasiert [SKKR01]). Arbeiten wie [JSZ06] zeigen, dass auch dieses einfache Vorgehen mit graphischen Modellen weiter verbessert werden kann.

3 Verwendete Grundlagen aus dem Natural Language Processing (NLP)

Dieses Kapitel stellt einige Grundlagen der verwendeten Techniken zur Lösung der motivierten Aufgabenstellung vor. Dazu gehören auch die mathematische Notation und ausgewählte Grundlagen aus dem maschinellen Lernen.

3.1 Notation

Wenn im Einzelfall nicht anders erklärt, dient die Notation aus Tabelle 3.1 als Konvention für alle mathematischen Konstrukte in Formeln und Algorithmen. Der erste Teil ähnelt der Notation aus [Bis07], der zweite der aus [SM12]. Falls nicht anders angegeben, ist der Wertebereich kleingeschriebener Variablen die Menge der natürlichen Zahlen von 1 bis zum Wert der entsprechenden großgeschriebenen Konstante.

Beispiel	Erklärung
N	Konstanten
$\mathbf{x}$	Spaltenvektor: fett gedruckte Kleinbuchstaben
$\mathbf{M}$	Matrix: fett gedruckte Großbuchstaben
$\mathbf{x}^{\mathrm{T}}, \mathbf{M}^{\mathrm{T}}$	Hochgestelltes T transponiert
$\mathbf{w}^{\mathrm{T}} = (w_1, \ldots, w_D)$	Elemente eines Zeilenvektors mit D Elementen
$\mathbf{w}^{\mathrm{T}} = (w_d)$	Abkürzung für Elemente eines Zeilenvektors mit D Elementen
$\mathbf{X} = \left[\mathbf{x}_n\right]$	Matrix aus N Spaltenvektoren
$\langle \mathbf{w}, \mathbf{x} \rangle$	Skalarprodukt $\mathbf{w}^{\mathrm{T}}\mathbf{x}$ der Vektoren $\mathbf{w}$ und $\mathbf{x}$
$\mathbf{0}$	Null-Vektor mit Dimension entsprechend dem Kontext
$\mathbf{I}$	Identitätsmatrix mit Dimension entsprechend dem Kontext
$\mathcal{M}$	Mengen und andere abstrakte Konstrukte
$\langle \cdot_t \rangle$	Sequenz (geordnete Menge) von T Elementen
Y	Menge von Zufallsvariablen
s	Index über Y
$\mathcal{Y}$	Diskrete oder kontinuierliche Ergebnismenge für jedes $Y_s \in Y$
$\mathbf{y}$	Beliebige Zuweisung zu Y, $y_s \in \mathcal{Y}$ ist der Y_s zugewiesene Wert

Tabelle 3.1.: Konventionen zur einheitlichen mathematischen Notation.

3.1.1 Englische Begriffe

Zu vielen Fachbegriffen dieser Arbeit existieren kaum oder wenig weitläufig bekannte Übersetzungen. Diese sind teilweise sehr kompliziert, wirken künstlich oder würden den Lesefluss fachkundiger Leser stören. So beinhaltet bereits der Titel das Wort „Parsing“, welches den Prozess des Auslesens von Zeichen in eine dem Computer verständliche(re) Form bezeichnet. Um jedoch möglichst wenig Annahmen zum Kenntnisstand des Lesers zu machen, werden daher in Tabelle 3.2 die wichtigsten eingedeutschten, englischen Begriffe und ihre Bedeutung in dieser Arbeit eingeführt.

Englischer Begriff	Deutsche Erklärung
Begin-of/Inner-of/Out-of (BIO)-Tag	Getaggte Wortsequenzen besitzen ein Starttoken („begin“) und beliebig viele innere Tokens („inner“). Sämtliche anderen Tokens („out“) gehören zu keinem Label.
Crawling	Möglichst automatisches Zusammentragen relevanter Dokumente.
Label	Zuweisung eines Tags zu einem Token.
Feature	Merkmal einer Beobachtung, anhand dessen Regeln oder Wahrscheinlichkeitsverteilungen zur Bestimmung von Labels erstellt oder gelernt werden können.
Overfitting	Übermäßige Anpassung eines Modells an die Trainingsdaten, sodass neue, unbekannte Daten schlechter klassifiziert werden.
Query	Anfrage an ein Suchsystem (Englischer Plural „Queries“ wird verwendet).
Recall	Sensitivität / Trefferquote. Sie ist definiert als die Anzahl der gefundenen relevanten Dokumente geteilt durch die Anzahl aller relevanten Dokumente.
Tag	Element der endlich vielen semantischen Kategorien (Datenbankfelder), das einem Token zugewiesen werden kann.
Tagging	Prozess der Zuweisung eines Tokens zu einem Tag (Segmentierung und Klassifizierung).
Token	Geschlossene Zeichensequenz, die man im Sinne des Taggings als atomare Texteinheit betrachtet. Sinnvollerweise Bedeutungsträger, die weiterverarbeitet werden können.

Tabelle 3.2.: Englische Fachbegriffe im Fließtext dieser Arbeit.

3.2 Conditional Random Field (CRF)

Als wichtigste Grundlage für die IE dient das Conditional Random Field (CRF). Um zu verstehen, wie es in Abschnitt 4.2 verwendet wird, folgt eine Übersicht seiner wichtigsten Eigenschaften und Formalismen.

3.2.1 Das Problem

Allgemein soll einer beobachteten Merkmalssequenz $\mathbf{x}$, die in T Einzelbeobachtungen und Merkmale $\langle \mathbf{x}_1, \ldots, \mathbf{x}_T \rangle$ unterteilt ist, eine entsprechende Klassifizierung $\mathbf{y} = \langle \mathbf{y}_1, \ldots, \mathbf{y}_T \rangle$ vorhergesagt werden. Einzelbeobachtungen im Sinne des NLP sind hier Wörter (Tokens) und ihre Merkmale sind, neben ihrer Identität, beliebige weitere Attribute wie z. B. Part-of-Speech (POS)-Tags, Form oder Vorkommen in anwendungsspezifischen Listen.

3.2.2 Motivation

Das CRF ist ein graphisches, stochastisches Modell. Die verwandten Arbeiten aus Abschnitt 2.2 zur IE zeigen, dass stochastische Sequenz-Tagger generell sehr gut funktionieren, da diese die Abhängigkeiten innerhalb der Sequenz von Klassen berücksichtigen. Sie lassen sich in graphischer Notation mit den Zufallsvariablen als Knoten übersichtlich darstellen. Die Autoren von [SM06] betonen jedoch die Wichtigkeit weiterer Merkmale einzelner Beobachtungen zur Klassifizierung. Generative Modelle wie das HMM, die die Gesamtwahrscheinlichkeit $p(\mathbf{x}, \mathbf{y})$ für die Beobachtungen $\mathbf{x}$ und ihnen jeweils zugeordneten Klassen $\mathbf{y}$ als Belegungen der Mengen von Zufallsvariablen X respektive Y nachbilden, scheitern an der zu hohen Komplexität der Berechnung von $p(\mathbf{x})$ wegen der starken Abhängigkeiten der Merkmale untereinander.

Da jedoch lediglich die Klassifizierung von Interesse ist, genügt ein Modell, das nur $p(\mathbf{y}|\mathbf{x})$ direkt modelliert. Nach [LMP01] leiden jedoch viele solcher Modelle wie das Maximum Entropy Markov Model (MEMM) am sogenannten „Label Bias“. Dieser entsteht, wenn die ausgehenden Kanten eines Knoten ausnahmslos untereinander konkurrieren, die Ausgangswahrscheinlichkeiten also lokal normiert werden. Im Extremfall eines einzigen Übergangs werden die beobachteten Merkmale komplett ignoriert. Das Tagging von Stellenanzeigen soll daher mittels eines CRFs realisiert werden, da hier die Normalisierung über ganze Sequenzen global vorgenommen wird.

3.2.3 Das Modell

Das CRF ähnelt dem HMM mit beliebigen Übergangsfunktionen – auch auf Merkmalen vergangener oder zukünftiger Beobachtungen. Damit werden seine starken Unabhängigkeitsannahmen außer Kraft gesetzt und es handelt sich zwangsweise nicht mehr um ein generatives Modell. Was bleibt ist die bedingte Wahrscheinlichkeit einer Sequenz von Labels, gegeben der paarweise zugehörigen, gleichlangen Sequenz von Beobachtungen. In diesem Abschnitt soll das CRF der Arbeit [SM12] folgend kurz beschrieben werden. Aus dem HMM wird das linear verkettete CRF abgeleitet, das wiederum vollständig verallgemeinert wird.

Gegeben sei eine Sequenz von T Beobachtungen $X = \langle x_t \rangle$ mit der zugehörigen Sequenz von Zuständen $Y = \langle y_t \rangle$ aus den endlichen Mengen O und S. Zur einfacheren Berechenbarkeit von $p(\mathbf{y}, \mathbf{x})$ geht die Markov-Annahme nun davon aus, dass jeder Zustand nur von seinem Vorgänger abhängt und die Beobachtungen wiederum nur vom jeweiligen Zustand. Daraus folgt die Gesamtwahrscheinlichkeit

$$p(\mathbf{y}, \mathbf{x}) = \prod_{t=1}^{T} p(y_t|y_{t-1}) p(x_t|y_t). \tag{3.1}$$

Sei $G = (V, E)$ ein gerichteter, azyklischer Graph, wobei $\pi(s)$ die Indices der Eltern von Knoten Y_s in V liefert. Dann ist Formel (3.1) kompatibel zur Faktorisierung

$$p(\mathbf{y}) = \prod_{s=1}^{S} p(y_s|y_{\pi(s)}). \tag{3.2}$$

In der Notation gerichteter graphischer Modelle entspricht Formel (3.1) damit Abbildung 3.1.

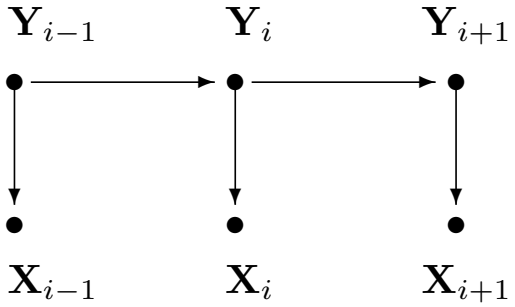

Abbildung 3.1.: Graphische Repräsentation des gerichteten HMM. Die X_i sind die Beobachtungen und Y_i die jeweiligen Zustände. [LMP01]

[SM12] zeigen nun, dass sich Formel (3.1) in

$$p(\mathbf{y},\mathbf{x}) = \frac{1}{Z}\prod_{t=1}^{T}\exp\left(\sum_{i,j\in S}\theta_{ij}\mathbf{1}_{\{y_t=i\}}\mathbf{1}_{\{y_{t-1}=j\}} + \sum_{i\in S}\sum_{o\in O}\mu_{oi}\mathbf{1}_{\{y_t=i\}}\mathbf{1}_{\{x_t=o\}}\right) \tag{3.3}$$

mit

$$\begin{aligned} \theta_{ij} &= \log p(y'=i|y=j) \\ \mu_{oi} &= \log p(x=o|y=i) \\ Z &= 1 \end{aligned} \tag{3.4}$$

umschreiben lässt, was bereits an die logistische Regression[1] erinnert. Mit den notationellen Abkürzungen $f_{ij}(y,y',x) = \mathbf{1}_{\{y=i\}}\mathbf{1}_{\{y'=j\}}$ für Zustandsübergänge und $f_{io}(y,y',x) = \mathbf{1}_{\{y=i\}}\mathbf{1}_{\{x=o\}}$ für Zustands-Beobachtungspaare und dem Trick, dass f_k über alle f_{ij} und f_{io} iteriert, da diese dieselbe Form besitzen, ergibt sich das HMM als

$$p(\mathbf{y},\mathbf{x}) = \frac{1}{Z}\prod_{t=1}^{T}\exp\left(\sum_{k=1}^{K}\theta_k f_k(y_t,y_{t-1},x_t)\right). \tag{3.5}$$

Mit der Verallgemeinerung der Featurefunktionen f_k in die Form $f_k(y,y',\mathbf{x}_t) \in \mathbb{R}$ und Konditionierung von Formel (3.5) auf $\mathbf{x}$ ergibt sich das linear verkettete CRF

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\prod_{t=1}^{T}\exp\left(\sum_{k=1}^{K}\theta_k f_k(y_t,y_{t-1},\mathbf{x_t})\right) \tag{3.6}$$

mit der Normalisierungsfunktion

$$Z(\mathbf{x}) = \sum_{\mathbf{y}}\prod_{t=1}^{T}\exp\left(\sum_{k=1}^{K}\theta_k f_k(y_t,y_{t-1},\mathbf{x_t})\right). \tag{3.7}$$

Mit der Faktorgraph-Notation aus [SM12] und der Faktorisierung von Formel (3.6) in

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\prod_{t=1}^{T}\Psi_t(y_t,y_{t-1},\mathbf{x_t}) \tag{3.8}$$

mit den Faktoren

$$\Psi_t(y_t,y_{t-1},\mathbf{x_t}) = \exp\left(\sum_{k=1}^{K}\theta_k f_k(y_t,y_{t-1},\mathbf{x_t})\right) \tag{3.9}$$

ergibt sich der ungerichtete Faktorgraph in Abbildung 3.2. Dabei sind die $\mathbf{x_t}$ aus Formel (3.9) die Beobachtungen und jene berechneten Features, die zum Zeitpunkt t benötigt werden. Im allgemeinen Fall können dies sämtliche Beobachtungen und Features $\mathbf{x}$ sein. Berücksichtigt man im Zeitpunkt t nur die aktuelle Beobachtung und ihre

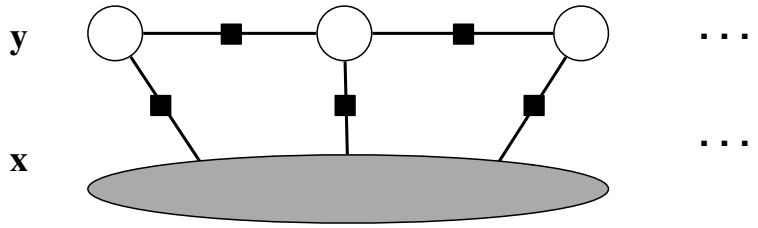

Abbildung 3.2.: Faktorgraph des linear verketteten CRF. [LMP01]

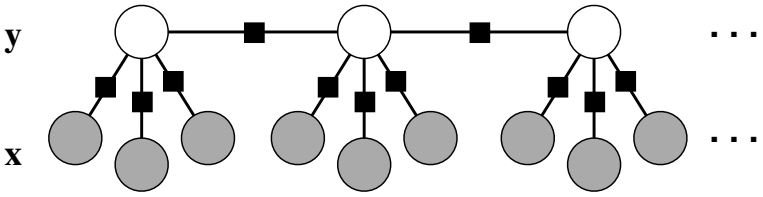

Abbildung 3.3.: Faktorgraph des linear verketteten CRF in HMM-Form. [LMP01]

Merkmale, ergibt sich der HMM-ähnliche Faktorgraph in Abbildung 3.3.

Das CRF aus Formel (3.8) lässt sich nun leicht auf allgemeinere Faktorgraphen erweitern. Jedoch wird in dieser Arbeit wegen Geschwindigkeitsvorteilen die Implementierung der CRFsuite [Oka07] verwendet. Diese beschränkt sich auf die spezielleren, linear verketteten CRFs. Eine Maximum-Likelihood Estimation (MLE) der Parameter θ_k kann mit verschieden Algorithmen für die N Trainingsdaten $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}$ vorgenommen werden. Mit Methoden der Regularisierung wie bei [MC10], kann „Overfitting“ durch zu hohe Normen des Parametervektors $\boldsymbol{\theta}$ bekämpft werden. Mit dem Forward/Backward-Algorithmus aus [Rab90] wird die Dekodierung, also die Bestimmung von $\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}, \{\theta_k\})$ zur Klassifizierung, vorgenommen.

3.3 Information Retrieval (Informationsrückgewinnung) (IR)

Laut [MRS08] ist IR die Beschaffung von typischerweise Textdokumenten aus einer großen Sammlung derselben, die einem bestimmten Informationsbedarf genügen. Im Folgenden sollen theoretische Ansätze und praktische Implementierungen vorgestellt werden, die versuchen diese Aufgabe zu erfüllen.

3.3.1 Theoretische Grundlagen

Die wichtigsten theoretischen Grundlagen des IR sollen [BYRN99] und [MRS08] folgend vorgestellt werden. Zunächst werden die wichtigsten Grundlagen des IR am Beispiel vom „Boolean Retrieval“ erläutert. Darauf folgt eine Motivation des IR im Vektorraum.

Konzeptuell lässt sich laut [BYRN99] ein IR-System als Quadrupel $\langle \mathcal{D}, \mathcal{Q}, \mathcal{F}, R : \mathcal{D} \times \mathcal{Q} \mapsto \mathbb{R} \rangle$ formulieren. Dabei ist $\mathcal{D}$ die Menge der Dokumente, $\mathcal{Q}$ die Menge der Informationsbedürfnisse (Queries), $\mathcal{F}$ als Framework bestimmt die Art, in der $\mathcal{D}$ und $\mathcal{Q}$ modelliert werden und R ist eine Funktion, die einen Wert berechnet, wie relevant ein Dokument für einen Query ist. $\mathcal{D}$ unter $\mathcal{F}$ soll für diesen Abschnitt konsistent definiert werden. Sei $\mathcal{V}$ das Vokabular, also die Menge der eindeutigen Tokens der Elemente aus $\mathcal{D}$. Dann werden Dokumente als Gewichtsvektoren $\mathbf{w}_d = (w_{1d}, \ldots, w_{|\mathcal{V}|d})$ mit den Gewichten $w_{vd} \geq 0$ modelliert, die ausdrücken, wie wichtig Vokabular $v \in \mathcal{V}$ für Dokument $d \in \mathcal{D}$ ist. Diese Wichtigkeit wird meist aus der Häufigkeit des Tokens abgeleitet. Mit diesem Framework lassen sich nun Boolean Retrieval und Ranked Retrieval im Vektorraum erklären.

[1] Diese verhält sich als diskriminierendes Pendant zum generativen Naive Bayes, wie das linear verkettete, diskriminierende CRF zum generativen HMM (Siehe Abbildung 2.4 in [SM12]).

Boolean Retrieval

Beim Boolean Retrieval werden binäre Gewichte aus $\{0,1\}$ vergeben. Diese sollen ausdrücken, ob ein Token im Dokument vorkommt oder nicht. In diesem Framework sollen die Booleschen Operatoren $\mathcal{B} = \{OR, AND, NOT\}$ in einer Algebra für $\mathcal{Q}$ unterstützt werden. Hierfür kann R rekursiv definiert werden:

$$R(d,q) = \begin{cases} w_{vd} & \text{falls } q = v \in \mathcal{V} \\ \max_n R(d,q_n) & \text{falls } q = OR(q_1,\ldots,q_N) \\ \min_n R(d,q_n) & \text{falls } q = AND(q_1,\ldots,q_N) \\ 1 - R(d,q') & \text{falls } q = NOT(q') \end{cases}. \tag{3.10}$$

Um z. B. alle Dokumente zu erhalten, die das Wort „Informatik“, aber nicht das Wort „Darmstadt“ enthalten, könnte der Query $q = AND(\text{Informatik}, NOT(\text{Darmstadt}))$ verwendet werden.

Ranked Retrieval im Vektorraummodell

Der größte Nachteil des Boolean Retrieval ist, dass teilweise Übereinstimmungen nicht gefunden werden, da sie schlichtweg nicht definiert sind. Außerdem können die Ergebnisse in keiner bestimmten Reihenfolge angegeben werden.

Beim Framework des Ranked Retrieval im Vektorraum werden daher Queries genauso modelliert wie Dokumente. Dokumente und Queries entsprechen also Gewichtsvektoren derselben Dimension. Als beliebtes Ähnlichkeitsmaß und daher auch Relevanzmaß kann dann der Kosinus des Winkels zwischen diesen beiden Vektoren

$$R(d,q) = \cos(\angle(\mathbf{w}_d, \mathbf{w}_q)) = \frac{\langle \mathbf{w}_d, \mathbf{w}_q \rangle}{\|\mathbf{w}_d\| \times \|\mathbf{w}_q\|} \tag{3.11}$$

mit der Euklidischen Vektornorm $\|\cdot\|$ verwendet werden. Da keine negativen Gewichte zugelassen sind, bewegt sich dieses Maß zwischen 0 und 1. Mit diesem Maß können auch teilweise Übereinstimmungen gefunden werden.

Ein wichtiges Problem dieser Methode ist jedoch die Wahl der Gewichte der Vokabeln pro Dokument. Es ist selten gut, hierfür einfach die reine Häufigkeit tf_{vd} des Wortes v in Dokument d zu verwenden. Allgemein häufig vorkommende Füllwörter („stopwords“) im Query könnten dadurch die Ähnlichkeitswerte zu Gunsten von besonderes langen Dokumenten verzerren. Sehr gängig ist daher das Konzept der inversen Dokumentenhäufigkeit („inverse Document Frequency“). Diese wird in [MRS08] definiert als

$$idf_v = \log \frac{|\mathcal{D}|}{df_v} \tag{3.12}$$

mit der Anzahl von Dokumenten df_v, die v beinhalten. Dieser Wert ist, laut den Autoren, für allgemein seltene Wörter eher groß und für besonders häufige Wörter eher klein.

Es wird nun vorgeschlagen, diese inverse Document Frequency mit den absoluten Häufigkeiten zu multiplizieren und so die Gewichte

$$w_{vd} = \text{tf-idf}_{vd} = tf_{vd} \times idf_v \tag{3.13}$$

zu erhalten.

3.3.2 Ausgewählte Systemdetails für Lucene und Solr

Lucene[2] implementiert diese und weitere Ansätze auf sehr effiziente Weise mit einem (umgekehrten) Suchindex. Da dieses Programm zusammen mit Solr[3] für das Jobvorschlagesystem verwendet wird, werden ausgewählte Details der beiden Systeme zum besseren Verständnis präsentiert.

[2] http://lucene.apache.org/

[3] http://lucene.apache.org/solr/

Lucene

Das IR-System Lucene implementiert ein eigenes Framework[4] mit einer standardmäßig vektormodellbasierten Ähnlichkeitsfunktion[5]

$$R(d,q) = score(q,d) = coord(q,d) \times queryNorm(q) \times \sum_{t \in q} \left(\text{tf-idf}_{td} \times t.getBoost() \times norm(t,d) \right), \qquad (3.14)$$

die mit den hier nicht näher beschriebenen Erweiterungen zusätzliche einstellbare Parameter bereitstellt. Es sei lediglich erwähnt, dass $norm(t,d)$ Einfluss auf die Gewichte der Felder ausübt. Ein Treffer in kürzeren Feldern wird mit dieser Funktion höher bewertet.

Die wichtigste Eigenschaft von Lucene für diese Arbeit ist, dass es feldbasiert arbeitet. Zu einem Dokument können beliebig viele benannte Felder mit Inhalt hinzugefügt werden. Diese Felder können dann unabhängig voneinander durchsucht werden. Sie sollen später genau den extrahierten Feldern entsprechen.

Ein Beispielquery, der im Feld „Ort“ nach „Darmstadt“ oder im Feld „Ausbildung“ nach „Informatik“ sucht, würde wie folgt aussehen:

$$q = \text{Ort:Darmstadt } \mathit{OR} \text{ Ausbildung:Informatik.} \qquad (3.15)$$

Damit Dokumente, auf die sogar beides zutrifft, zuerst zurückgeliefert werden, wirkt *OR* hier im übertragenen Sinne wie eine Summe von Teilscores.

Solr

Solr stellt die Dienste von Lucene als Webserver bereit. Mit dessen Queryparser ExtendedDisMax[6] lassen sich Queries laut den Verfassern sehr leicht aus syntaktisch ungeprüften Nutzereingaben formulieren. Dabei steht das „Dis“ für die Disjunktion wegen der Suche auf mehreren Feldern und das „Max“ bedeutet, dass ein Treffer in mehreren Feldern nicht jedes mal zur Gesamt-Score addiert wird, sondern nur das Maximum.[7] Die folgenden Parameter werden für die Stellensuche verwendet und deshalb hier kurz erklärt:

- q.alt: Query in Standard-Lucene-Query-Syntax
- qf: Felder, in denen mit q.alt gesucht werden soll
- pf: Felder, in denen mit q.alt als exakte Phrase gesucht werden soll
- qs / ps: Anzahl Positionen, die Tokens in Phrasen aus q.alt / pf auseinander liegen dürfen
- tie: Relativiert „Max“: Die Score ist das Maximum addiert mit $tie \times$ (Score auf den anderen Feldern)

In dieser Arbeit wird wie empfohlen stets $tie = 0,1$ gesetzt. Es wäre außerdem möglich, für jedes Feld konstante Gewichtungsfaktoren anzugeben. Darauf soll jedoch zur Übersichtlichkeit dieser Arbeit verzichtet werden. Dank $norm(t,d)$ in Formel (3.14) ist davon auszugehen, dass Treffer in den extrahierten Feldern sowieso stärker zählen, da diese inhärent kürzer sind. Der Suchindex wird weitestgehend in der Standardkonfiguration mit deutscher Wortstammbildung und den mitgelieferten Füllwortfiltern erstellt.

[4] http://lucene.apache.org/core/4_0_0/queryparser/org/apache/lucene/queryparser/classic/package-summary.html
[5] http://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html
[6] http://wiki.apache.org/solr/ExtendedDisMax
[7] http://wiki.apache.org/solr/DisMax

4 Parsing von Stellenanzeigen

In diesem Teil der Arbeit wird präsentiert, wie die relevanten Felder aus Stellenanzeigen im HTML-Format extrahiert werden. Zunächst wird in Abschnitt 4.1 die Aggregation der Trainingsdaten erklärt. In Abschnitt 4.2 wird die vorgeschlagene Parsing-Lösung formuliert. Diese wird abschließend in Abschnitt 4.3 evaluiert.

4.1 Trainingsdaten sammeln

Zunächst müssen die zu extrahierenden Felder identifiziert werden. Ein anfänglich entfernt überwacht getaggter Korpus wird anschließend manuell vervollständigt. Das entfernt überwachte Tagging bringt zwei große Vorteile mit sich. Zum einen kann schnell ein erster getaggter Korpus erstellt werden, mit dem bereits erste Experimente wie in Abschnitt 4.3.1 durchgeführt werden können. Zweitens wird auch der Aufwand des manuellen Taggings reduziert. Es verbessert in WebAnno ungemein die Übersichtlichkeit und Orientierung, wenn bereits einige Labels vorhanden sind.

4.1.1 Vorarbeit

In einer kurzen Studie wurden 20 zufällig ausgewählte Stellenanzeigen verschiedener Quellen betrachtet. Diese vermitteln ihre Anforderungen und Informationen gut mit den Feldern aus Tabelle 4.1. Sie ähneln dabei denen aus [LBC+10], sind jedoch teilweise stark verallgemeinert und beinhalten keine Subtypen. Noch detaillierter ist der HR-XML-Standard[1] mit der Nomenklatur `PositionOpening`. Diese wird auch von der Jobbörse der Bundesagentur für Arbeit[2] in angepasster Form [fA13] verwendet. So wird im Feld Firma nicht zwischen der einstellenden oder der vermittelnden Firma unterschieden. Im Anschluss werden die Felder noch etwas verändert, um einheitlicheres manuelles Tagging zu ermöglichen. Die Häufigkeit der Vorkommen von Gehaltsangaben wäre 0, wenn man nur konkrete Zahlen akzeptierte. Acht der Stellen bieten nur gehaltlose Aussagen der beispielhaften Form „eine der Stelle entsprechende Bezahlung“. Anscheinend geben nur sehr wenige Firmen hier konkrete Werte an.

Feld	#
Stellenbezeichnung	20
Firma	20
Ort	19
Aufgabenbeschreibung	18
Talente	16
Praktische Erfahrung	16
Spezielle Fähigkeiten	13
Umfang und Art der Stelle	10
Beginn	9
Geforderte Ausbildung	9
Laufzeitende	8
Sprachkenntnisse	8
Gehalt	8

Tabelle 4.1.: Vorkommen der semantischen Felder in 20 manuell getaggten Stellen.

[1] `http://www.hr-xml.org/`
[2] `http://www.arbeitsagentur.de/`

4.1.2 Entfernt überwacht getaggter Korpus

Als Grundlage eines vollständig annotierten Korpus für diese Arbeit wurden 1010 Stellenanzeigen gecrawlt. Als Quelle dient hierfür die Meta-Jobsuchmaschine Jobrapido[3]. Gesucht wird hier nach dem Begriff „Informatik“. Abbildung 4.1 zeigt beispielhafte Suchergebnisse. Umrandet sind die Metadaten, die fast genau so im Text der Stellenanzeige zu finden sind. Es handelt sich dabei um die Stellenbezeichnung, die Firma und den Ort wie in Tabelle 4.1.

Abbildung 4.1.: Metadaten als überwachte Lerngrundlage.

An dieser Stelle sei erwähnt, dass die Beobachtungen im Sinne des CRFs genau die Tokens der Textknoten[4] des HTML-Dokuments sind. Abbildung 4.2 zeigt, wie die Anzahl der Tokens über den Datensatz von Jobrapido verteilt ist. Dabei wird zwischen angezeigten („displayed“) und allen Tokens zusammen unterschieden. „Angezeigt“ meint in diesem Falle, dass der Textknoten des Tokens bei Öffnen der HTML-Datei im Browser[5] ohne Nutzerinteraktion dargestellt wird. Aus dem kumulativen Histogramm geht hervor, dass ca. 70% der Stellen weniger als 500 angezeigte Tokens besitzen. Nimmt man auch die versteckten[6] hinzu, erreicht man erst mit <1600 Tokens diese 70%. Später wird mit derselben Analyse eines größeren Datensatz klar, dass hier vergleichsweise viele Stellen mit vielen versteckten Tokens vorkommen.

Wegen einer Kooperation mit der Meta-Jobsuchmaschine cesar[7] mit denselben Meta-Feldern rückt das Crawling im weiteren Verlauf der Arbeit in den Hintergrund. Von cesar werden regelmäßig neue Stellenanzeigen und ihre Uniform Resource Locators (URLs) zur Verfügung gestellt.

4.1.3 Tags manuell vervollständigen

Der entfernt überwacht getaggte Korpus ist bei weitem nicht fehlerfrei. Die Metadaten sind meistens manipuliert und können nicht exakt so gefunden werden. Auf der anderen Seite werden Metadaten zufällig in Teilen des Dokuments exakt gefunden, die nichts mit der Stelle zu tun haben. Viele Jobanzeigen enthalten solchen sogenannten Boilerplate wie z. B. Navigationselemente oder ähnliche Jobangebote. Diese sollten nicht getaggt werden. In [LBC+10] wird grob skizziert, wie man Boilerplate identifizieren kann. Mit gut gewählten Merkmalen der Beobachtungen sollte es jedoch möglich sein, diese Aufgabe auch dem CRF zu überlassen.

Des Weiteren müssen noch die Annotationen der weiteren Tags erstellt werden. Dies geschieht in paralleler Teamarbeit mit WebAnno[8]. Abbildung 4.3 zeigt die Annotationsoberfläche von WebAnno. Ein Nutzer wählt mit dem Cursor Tokens aus. Ein Dialog erscheint (nicht im Screenshot) und fragt nach dem zu vergebenden Tag. Auf Kosten der Ausdrucksfähigkeit wird nicht überlappend annotiert, um mit einem einzigen CRF-Modell auszukommen. Dies betrifft jedoch hauptsächlich nur die Aufgabenbeschreibung, die oft auch Fähigkeiten und Talente enthält. Ein

[3] `http://de.jobrapido.com/`
[4] Siehe Document Object Model (DOM) des W3C unter `http://www.w3.org/DOM/`.
[5] Es wird stets die Implementierung von WebKit (`https://www.webkit.org/` der Qt-Bibliothek (`http://qt-project.org/`) in der Version 4 verwendet.
[6] Blätter der Textknoten mit Schriftgröße ≤ 0 oder reine Höhe in der Browser-Geometrie ≤ 0.
[7] `http://www.cesar.de/`
[8] `https://code.google.com/p/webanno/`

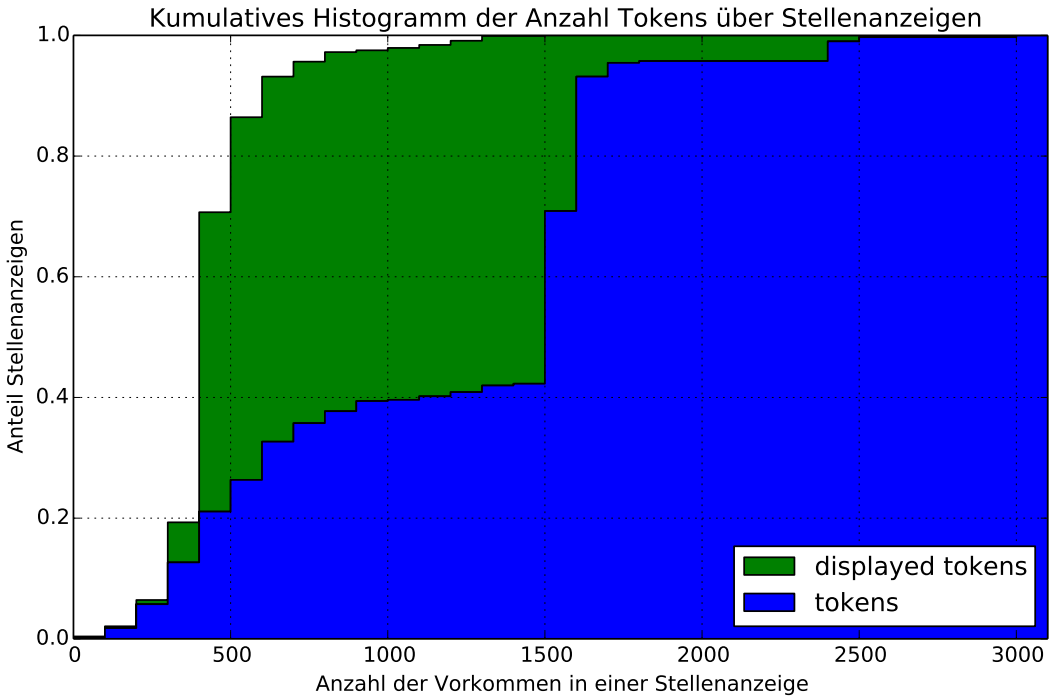

Abbildung 4.2.: Kumulatives Histogramm mit konstanter Schrittgröße 100 und unendlicher Größe des letzten Intervalls. In dem entfernt generierten Datensatz kommen viele Stellen mit nicht angezeigten Tokens vor.

großes Problem stellt die Konsistenz der verschiedenen Annotatoren dar. Ein internes Dokument von Richtlinien mit vielen Regeln und Beispielen wächst kontinuierlich. Die zu annotierenden Felder werden nachfolgend konkretisiert.

Semantische Tags

Erste Annotationsdurchläufe ergaben, dass die Unterscheidung zwischen praktischer Erfahrungen und speziellen Fähigkeiten zu großen Inkonsistenzen führt. Sie werden deshalb schlicht zusammengeführt in ein umfassendes „Skill“-Feld. Weiterhin werden Kontaktdaten (u.a. Uniform Resource Locator (URL) der ausschreibenden Firma) und das Feld der Referenznummer eingeführt. Wichtig ist noch das Auftrennen von Fähigkeiten, Talenten, Sprachkenntnissen und Ausbildungsanforderungen. Diese gewünschten Eigenschaften des Bewerbers werden in die drei Teile

- Name / Beschreibung
- Niveau
- Notwendigkeit

unterteilt. Abbildung 4.4 gibt einen ganzheitlichen Überblick über alle semantischen Felder. Im Endeffekt ergibt sich die doppelte Anzahl an Tags + 1 aufgrund der BIO-Labels. Dadurch lassen sich z. B. aufeinander folgende Labels desselben Typs separat extrahieren.

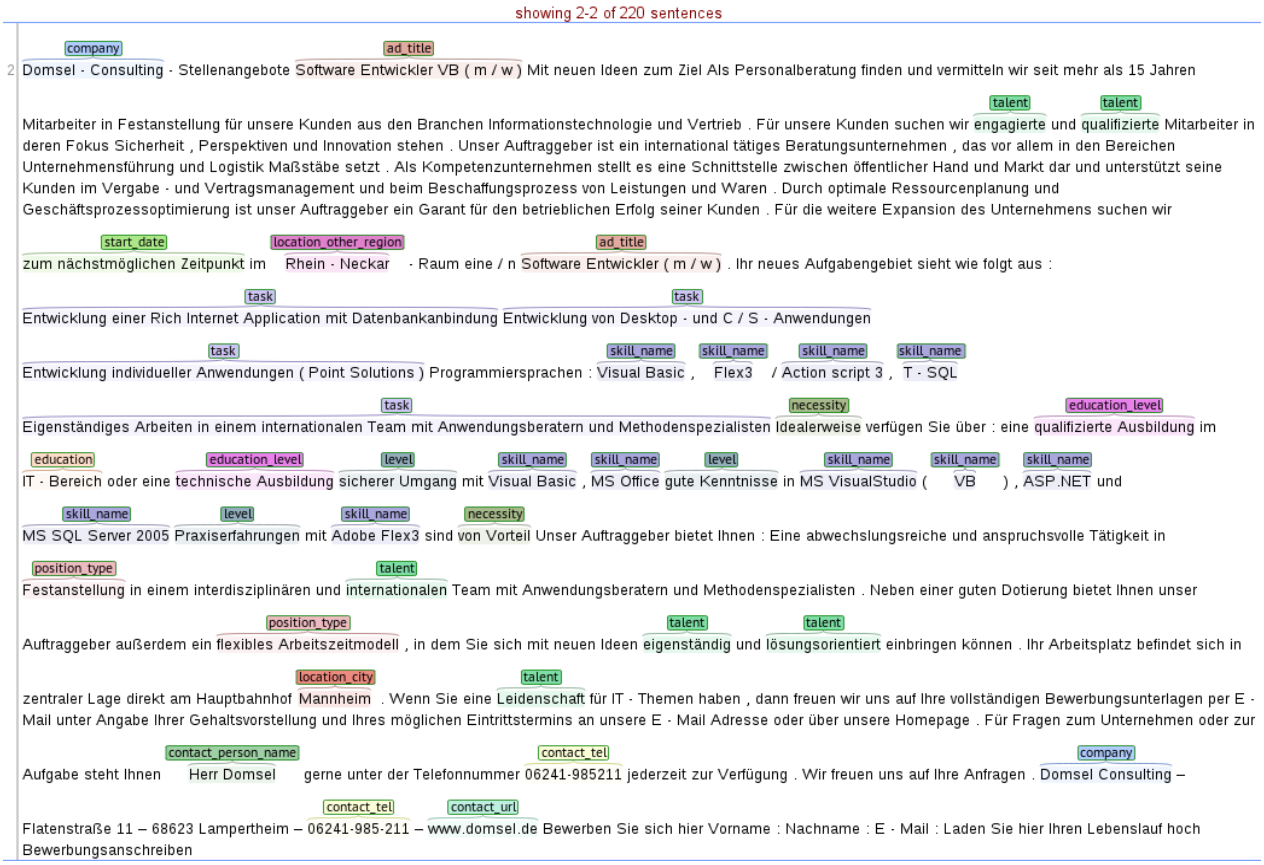

Abbildung 4.3.: WebAnno [YGdCB13] „named entity“-Ebene.

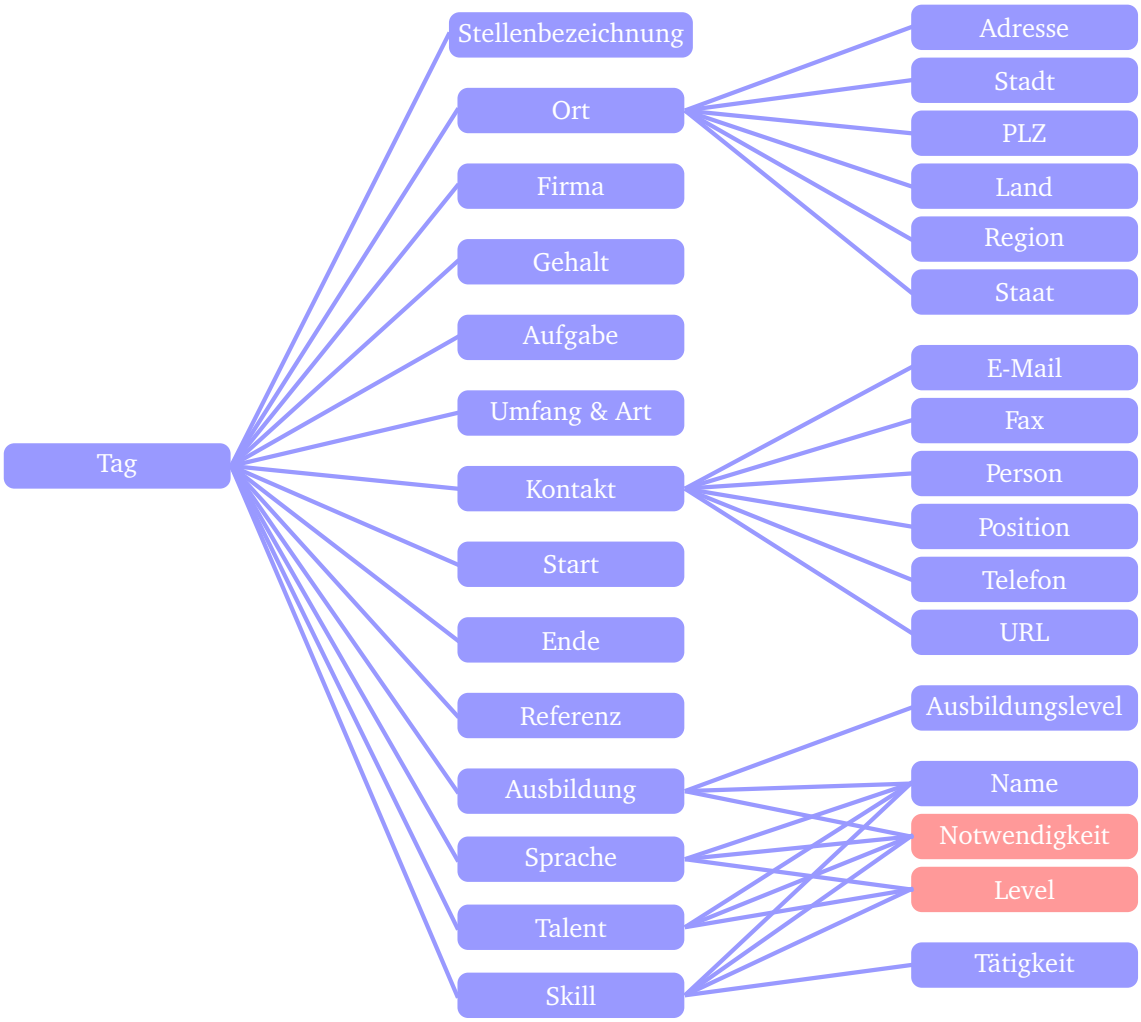

Abbildung 4.4.: Überblick über alle zu extrahierenden Felder. Die rot markierten Felder existieren trotz der mehreren Kanten wegen des ähnlichen Vokabulars nur einmal.

4.1.4 Taggingrichtlinien

Die Richtlinien des manuellen Taggings der Stellenanzeige lassen sich in zwei Rubriken teilen. Die erste Rubrik stellt dabei die Unterteilung der Stellenanzeige in Boilerplate- und zu taggenden Stellenanzeigentext wie in Tabelle 4.2 dar.

Art des Taggings	Taggingrichtline
Boilerplate	• Boilerplate wird nicht getaggt • Indiz (u.a.): <div style="display: none;" ...>, „Diese Stellenangebote könnten Sie auch interessieren: ...“
Stellenanzeige	• Regel von Abschnittsüberschriften • 28 Tagmöglichkeiten mit jeweils unterschiedlichen Taggingregeln

Tabelle 4.2.: Unterteilung von Stellenanzeigen in Boilerplate und relevanten, stellenbezogenen Inhalt beim Tagging.

Im Folgenden werden die Richtlinien der zu extrahierenden Tags im Detail in Tabelle 4.3 beschrieben. Der Stand der Richtlinien basiert auf dem 29.08.13. Als dynamisches Dokument werden die Richtlinien laufend angepasst, optimiert und erweitert. Es handelt sich dabei nur um einen kurzen Auszug des internen Dokuments. Im Anhang in Tabelle B.1 finden sich Auszüge zu den Richtlinien der weiteren Felder.

Label	Richtlinie (Auszug)	Beispiel
Ad title	• Stellenbezeichnung • Meist die Hauptüberschrift, die aber oft auch (in ähnlicher Form) im Angebotstext steht und dort auch getaggt werden muss • Kann Notationsabweichungen beinhalten	„Ihre Aufgabe als SAP WM LES Berater“ (ad_title = „SAP WM LES Consultant (m / w)“)) →„SAP LES Berater“ = ad_title
Education	• Nur so viel markieren, dass klar wird, welche(s) Ausbildung(s-Fach/-Gebiet) gefordert wird • Ausbildungstexte werden nicht getaggt.	„Sie besitzen ein abgeschlossenes Studium in Elektrotechnik oder technischer Informatik“ →„Elektrotechnik“, „technischer Informatik“ = education
Location city	• Stadt der zu besetzende Stelle • Wo die Stelle auszufüllen ist, oft nicht, wo die Firma sitzt • Nur die direkten Namen, kein Raum, Großraum, o.ä.	„Einsatzort im Großraum Darmstadt“ →„Darmstadt“ = location_city
Language name	• Die Sprachfähigkeit, die angefordert wird • So viel markieren, dass klar wird, welche Sprache gefordert wird	„Ihre sehr guten Englischkenntnisse haben Sie idealerweise im Rahmen von Auslandsaufenthalten vertieft“ →„sehr guten“ = level →„Englischkenntnisse“ = language_name
Company	• Firma, bei der man arbeitet oder die die Stelle ausschreibt • Inklusive GmbH, AG usw.	„Die AutoVision GmbH ist eine hundertprozentige Tochtergesellschaft von Volkswagen.“ →„AutoVision GmbH“, „Volkswagen“ = company
Skill name	• Anderweitige Fähigkeit / Kenntnis / Erfahrung, die angefordert wird • Skill hat Priorität vor Level • Programmiersprachen werden als Skills getaggt	„Erfahrung mit Modellierungstools wie MATLAB / Simulink , ASCET oder Rhapsody“ → „Erfahrung“ = level → „Modellierungstools“, „MATLAB“, „Simulink“, „ASCET“, „Rhapsody“ = skill_name „Gute Kenntnisse der HOAI - Verordnung“ →„Gute Kenntnisse“ = level →„HOAI – Verordnung“ = skill_name
Task	• Konkrete (Einzel-)Aufgabe, die zu erledigen ist • Wenn möglich, dann wird der Task nach „einschließlich“, „inklusive“, „sowie“ separat getaggt	„Durchführung des User und Request Managements (z.B . Anlegen , Ändern und Deaktivieren von Usern)“ →„Durchführung des User und Request Managements“, „Anlegen, Ändern und Deaktivieren von Usern“ = task
Talent	• Die persönliche Kompetenz, die angefordert wird • Talente sind ebenfalls Aussagen wie z.B. „überzeugt“, „durchsetzen“, „offen“ • „Denken“, „Handeln“, „Arbeitsweise“ usw. werden nicht getaggt	„Bei Kunden treten Sie professionell und beratungsstark auf“ „Kunden“, „professionell“, „beratungsstark“ = talent „hohe Motivation und schnelle Auffassungsgabe“ →„hohe“, „schnelle“ = level →„Motivation“, „Auffassungsgabe“ = talent

Tabelle 4.3.: Auszug aus den Richtlinien der wichtigsten Labels entsprechend ihrer englischen Benennung zur Abkürzung.

4.1.5 Überblick über die Verteilung der wichtigsten Tags

Tabelle 4.4 gibt in absoluten Zahlen konkret an, wie viele Spannen pro Label auf allen 1010 Stellen zusammen getaggt wurden. Die inhaltlich wichtigsten Felder gehören auch zu denen, die am häufigsten vorhanden sind. Seltener und damit auch schwieriger zu lernen sind hingegen die Felder administrativer Natur wie z. B. Faxnummern oder die Position der Kontaktperson. Auch sind konkrete Angaben zum Gehalt erwartungsgemäß niedrig.

Label	Anzahl	Label	Anzahl
Stellenbezeichnung	1806	Ort: Stadt	1330
Firma	3263	Ort: Land	84
Aufgabe	6099	Ort: Adresse	20
Kontakt: E-Mail	784	Ort: Region	112
Kontakt: Fax	33	Ort: Staat	377
Kontakt: Person	343	Ort: PLZ	118
Kontakt: Position	36	Notwendigkeit	1008
Kontakt: Telefon	341	Umfang & Art	702
Kontakt: URL	801	Referenz	930
Ausbildung	2218	Gehalt	30
Ausbildungslevel	2052	Tätigkeit	713
Ende	610	Skill	5663
Sprache	769	Start	926
Level	4816	Talent	3763

Tabelle 4.4.: Absolute Häufigkeiten ganzer Spannen von Labels im manuell getaggten Datensatz von 1010 Stellen.

Abbildung 4.5 zeigt, wie die acht für das Matching interessantesten[9] Labels über die manuell getaggten Stellen als zusammenhängende Spannen verteilt sind. Es wird keine Deduplizierung vorgenommen. Wie bereits vermutet, enthält jede Ausschreibung mindestens den Titel der Stelle („ad title“). Knapp über 60% enthalten genau drei Firmenbezeichnungen, gefolgt von über 55%, die die Stellenbezeichnung genau einmal verwenden. Gute 40% stellen keine konkreten Sprachanforderungen, fast genau so viele fordern genau eine und >15% fordern mindestens zwei gesprochene Sprachen. Zu den häufiger in einer Stelle vorkommenden Labels gehören die geforderten allgemeinen Fähigkeiten, Aufgabenbeschreibungen und Talente, wobei letztere in über 20% der Stellen nicht konkretisiert werden. Ca. 10% enthalten jedoch 10 und mehr.

In Abbildung 4.6 wird mittels Histogrammen entschlüsselt, welche Labels im Schnitt aus wie vielen Tokens bestehen. Ausbildungen, Sprachen, Städte und persönliche Anforderungen bestehen zumeist aus nur einem Token. Es überrascht außerdem nicht, dass Aufgabenbeschreibungen eher längerer Natur sind. Die erstaunlich hohe durchschnittliche Länge von Stellenbezeichnungen hat zwei Ursachen, welche zusammenspielen. Das Allgemeine Gleichbehandlungsgesetz (AGG) zwingt die Inserenten häufig dazu, die Stellentitel mit Phrasen wie z. B. „(w/m)“ auszustatten. Die Tokenisierung, die in Abschnitt 4.2.1 erläutert wird, erzeugt daraus nun die 5 zusätzlichen Tokens „(“, „w“, „/“, „m“ und „)“. Es ist zu erwarten, dass sich diese Phrasen als besonders hilfreiche Features herausstellen.

4.2 Implementierung des Parsers

Wie bereits erwähnt, wird als CRF die CRFsuite [Oka07] verwendet. Es wird nun erläutert, wie sich den Stellen Beobachtungen in Form von Tokens gewonnen werden. Diese werden ihren Features und BIO-Labels zugeordnet, um damit ein CRF zu lernen.

4.2.1 Tokenisierung als Beobachtungen

Tokenisierung ist die Unterteilung von freiem Text in atomare Bedeutungsträger. In der Literatur wird dies oft mit Wörtern beschrieben. Anwendungsspezifisch gesehen ist diese Beschreibung jedoch unzureichend. Als Beispiel

[9] Wegen ihrer inhaltlichen Relevanz.

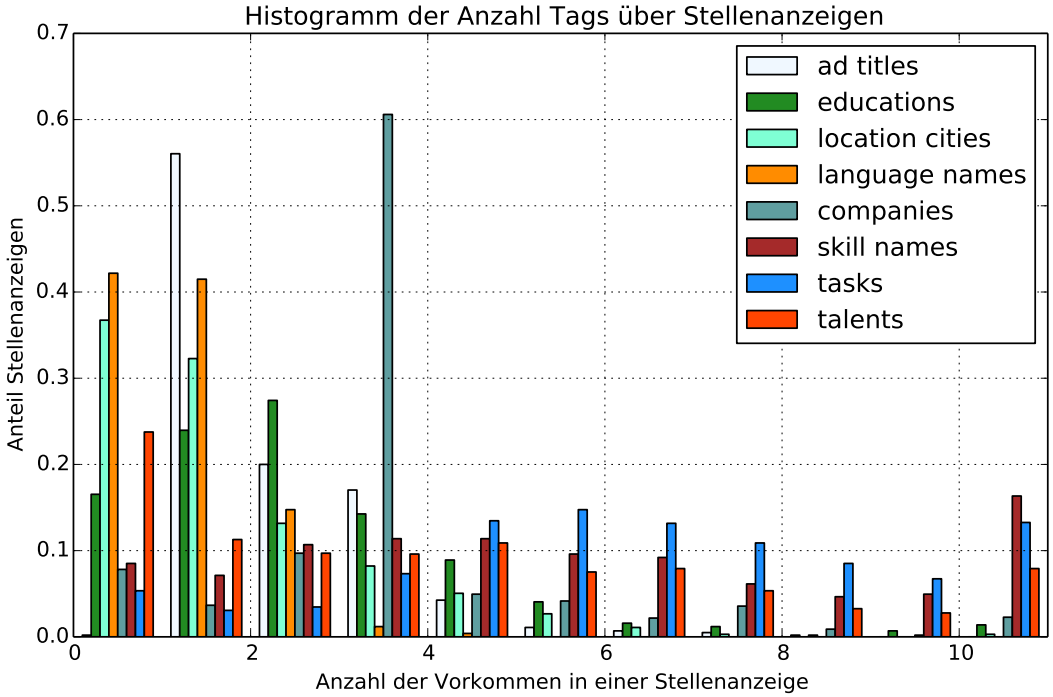

Abbildung 4.5.: Histogramm mit konstanter Schrittgröße 1 und unendlicher Größe des letzten Intervalls. Überblick über die Verteilung der wichtigsten Tags im manuell vervollständigten Korpus.

hierfür soll das grundlegende Element von Hypertext dienen: der Hyperlink in Form von URLs. Deren Bestandteile lassen sich nach ihrer Grammatik[10] in alle Bestandteile aufteilen. Beabsichtigt man jedoch die Extraktion ganzer URLs, wäre eine Auftrennung nicht nur vergeudete Rechenzeit[11], sondern die Segmentierung bei der IE wäre zusätzlich komplexer.

Wichtiger ist jedoch eine schnelle Tokenisierung mit wenig regulären Ausdrücken (Einfachheit). Es wird eine Übertokenisierung in Kauf genommen, damit möglichst exakt annotiert werden kann (Robustheit). Im Extremfall wäre eine zeichenweise Auftrennung denkbar. Diese würde aber das Modell des CRFs deutlich zu stark belasten. Zusätzlich erschwert wird die Tokenisierung durch viele neue Zeichen im Unicode-Standard[12], die in alten Tokenisierungsverfahren nicht berücksichtigt werden. Für Unicode existieren offizielle Standards zur Tokenisierung[13], die z. B. von Ucto[14] implementiert werden. Diese trennen jedoch Datumsangaben z. B. im Format `MM/DD/YYYY` rigoros auf.

Nach der Extraktion sollen die Informationen möglichst in ihrer ursprünglichen Form gespeichert werden. Bei jeder Trennung wird daher mitgeführt, ob ein neues Subtoken Beginn (B) oder ein innerer (I) Teil des Ausgangstokens ist. Als Anker dieser rekursiven Definition dient eine einfache Whitespace-Tokenisierung einzelner Textknoten. Es wird also angenommen, dass sich Tokens niemals über Whitespace oder Textknoten [15] hinweg erstrecken.

Im Detail wird daher die iterative Tokenisierung aus Algorithmus 4.1 vorgeschlagen. Sei $\mathcal{C}$ die Menge der Datenpunkte (Zeichen) in Unicode. Dann ist Algorithmus 4.1 eine Funktion $t : \mathcal{C}^N \mapsto \langle(\mathcal{C}_p^{L_p}, \{B, I\})\rangle$, also eine Abbildung von N Zeichen auf eine Sequenz von P Paaren von Tokens der Länge L_p und Markierungen, ob es sich um den Beginn eines whitespace-getrennten Tokens handelt.

[10] `http://www.w3.org/Addressing/URL/5_BNF.html`

[11] Fehlerfreies Erkennen von URLs mittels ihrer Grammatik wäre genauso komplex wie ihre Auftrennung zur Tokenisierung, könnte aber mit regulären Ausdrücken angenähert werden.

[12] `http://www.unicode.org/`

[13] `http://www.unicode.org/reports/tr29/`

[14] `http://ilk.uvt.nl/ucto/`

[15] Seltene Ausnahmen bilden Hervorhebungen von Bestandteilen von Tokens, z. B. `Bewerber<strong>in</strong>`, was als Bewerber**in** dargestellt werden soll, jedoch unwiederbringlich in „Bewerber“ und „in“ aufgetrennt wird.

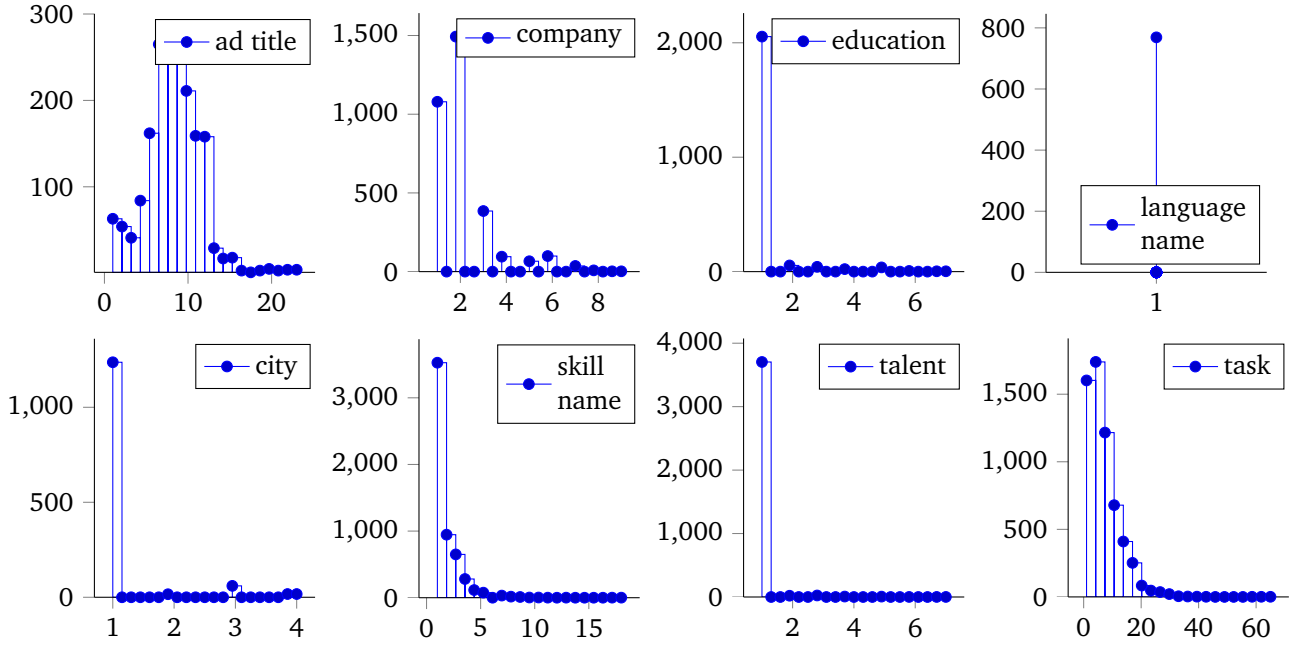

Abbildung 4.6.: Histogramme mit 20 gleich großen Intervallen über die Länge in Tokens der acht wichtigsten Labels.

In Algorithmus 4.1 ist $\mathcal{T}$ durch vier reguläre Ausdrücke r_i in `Python` implementiert. Sei $\mathcal{C}_{4.n}$ die Menge der Unicode-Zeichen aus Abbildung 4.n. Außerdem seien

$$\mathcal{I}nnen = \mathcal{C}_{4.7} \cup \mathcal{C}_{4.8} \tag{4.1}$$

und

$$\mathcal{R}and = \bigcup_{x \in \{4.7,\ldots,4.11\}} \mathcal{C}_x. \tag{4.2}$$

Dann findet r_1 alle möglichst langen[16] Sequenzen von $\mathcal{I}nnen$ inklusive der beliebig vielen[17] umgebenden $\mathcal{R}and$-Zeichen und behandelt diese Sequenzen als ein neues Token. Der Ausdruck r_2 liefert Zeichen aus $\mathcal{C}_{4.10}$ als neue Tokens, wenn links davon keine Ziffer steht. Ausdruck r_3 arbeitet analog für die rechte Seite. Damit bleibt z. B. das Token ⟨11/11/2011⟩ erhalten, ⟨1/a⟩ wird jedoch aufgetrennt in ⟨⟨1⟩, ⟨/⟩, ⟨a⟩⟩. Abschließend schneidet r_4 alle möglichst langen Teilsequenzen von $\mathcal{R}and$-Zeichen, die durchgehend am linken oder rechten Rand eines Tokens stehen, ab.

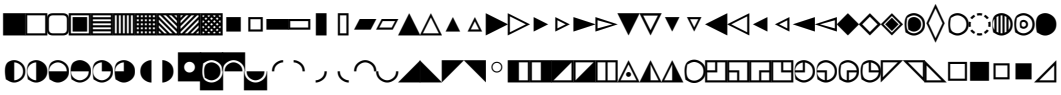

Abbildung 4.7.: Unicode-Block der geometrischen Formen. Werden häufig als Aufzählungszeichen verwendet und sollen immer abgetrennt werden.

Abbildung 4.8.: Verschiedene Satzzeichen und Aufzählungszeichen, die immer zu einer Trennung führen sollen.

[16] Ein „+“ in regulären Ausdrücken.
[17] Ein „*“ in regulären Ausdrücken.

Algorithmus 4.1 Framework zur rückabwickelbaren Tokenisierung

Benötige: Unicode-Datenpunkte $\mathcal{C}$
Benötige: Eingabetext $text \in \mathcal{C}^N$
Benötige: Beginn und innere Markierungen $\{B, I\}$
Benötige: Whitespace-Tokenisierung $ws : \mathcal{C}^M \mapsto \langle \mathcal{C}_p^{L_p} \rangle$
Benötige: PTB-Tokenisierung $ptbt : \mathcal{C}^M \mapsto \langle \mathcal{C}_p^{L_p} \rangle$
Benötige: Weitere Tokenisierungsverfahren $\mathcal{T}$, die vor allem Variationen von üblichen Satzzeichen und Sonderzeichen behandeln, siehe hierzu Abbildungen 4.7–4.11

$\langle t_1, \ldots, t_p \rangle \leftarrow ws(text)$ // Whitespace-Tokenisierung
$ergebnis_{alt} = \langle (t_1, B), \ldots, (t_p, B) \rangle$ // Markiere whitespace-getrennte Tokens stets als Beginn
for all $f \in \{ptb\} \cup \mathcal{T}$ **do**
  $ergebnis_{neu} \leftarrow \langle\rangle$
  for all (t, b) in $ergebnis_{alt}$ **do**
    $\langle t_1, \ldots, t_p \rangle \leftarrow f(t)$
    if $b = B$ **then**
      $neu \leftarrow \langle (t_1, B), (t_2, I), \ldots, (t_p, I) \rangle$ // Erhalte Anfang eines whitespace-getrennten Tokens
    else
      $neu \leftarrow \langle (t_1, I), (t_2, I), \ldots, (t_p, I) \rangle$ // Weitere Tokenisierung eines inneren Tokens
    end if
    $ergebnis_{neu} += new$ // Hänge neu an das neue Ergebnis
  end for
  $ergebnis_{alt} \leftarrow ergebnis_{neu}$
end for
return $ergebnis_{neu}$ // Äquivalent zu $ergebnis_{alt}$

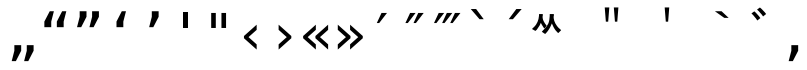

Abbildung 4.9.: Anführungszeichen und Zeichen, die als solche missbraucht werden.

4.2.2 Features der Beobachtungen (Tokens)

Aus den Tokens als Beobachtungen der DOM-Textknoten lassen sich mehr als nur umliegende Tokens als Features generieren. Zum Einsatz kommen hier:

- BI der Tokenisierung
- Form
- HTML-Tags (allgemein XPath)
- POS-Tags[18]
- Wortstamm
- Listen
- Visuelle Auffälligkeiten (CSS)
- Unüberwachte POS-Tags
- Features anderer Beobachtungen
- N-Gramme aller bisherigen Features

[18] Für die Experimente dieser Arbeit wurde ein Naive Bayes POS-Tagger mit dem TIGER Corpus (`http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.html`) trainiert. Es bietet sich aber an, auch für diese Aufgabe ein CRF zu verwenden. Wapiti [LCY10] bietet ein bereits trainiertes Modell zur freien Nutzung an.

Abbildung 4.10.: Zeichen, die lediglich innerhalb von Ziffern nicht trennen sollen.

Abbildung 4.11.: Punktzeichen und Zeichen, die als solche missbraucht werden.

BI

Die *BI*-Markierungen der Tokenisierung aus Abschnitt 4.2.1 können auch als Features der Tokens verwendet werden. Dies soll dem CRF bei der Segmentierung der tendenziell übertokenisierten Wörter helfen.

Form auf Zeichenebene

Das einfachste Feature ist die Form auf der Ebene der Schriftzeichen. Hier wird das klein geschriebene Token als Feature hinzugefügt. Vorkommen von Großschreibung und Zahlen und weitere Muster, wie sie gerne bei der Named Entity Recognition (NER) verwendet werden, kommen auch hier zum Einsatz. Sie sind der CRFsuite entnommen:

- Enthält: alphabetische Buchstaben, Ziffern (jeweils binär)
- Typ ∈ { „Nur Großbuchstaben“, „Nur Ziffern“, „Nur Symbole“, „Nur Großbuchstaben oder Ziffern“, „Nur Großbuchstaben oder Symbole“, „Nur Ziffern oder Symbole“, „Nur Großbuchstaben, Ziffern oder Symbole“, „Großgeschrieben“, „Nur Buchstaben“, „Nur Buchstaben oder Ziffern“ }
- Ersetze zeichenweise:
  - Großbuchstaben durch U
  - Kleinbuchstaben durch L
  - Ziffern durch D
  - Punkt und Komma durch Komma
  - Semikolon, Doppelpunkt, Fragezeichen und Ausrufezeichen durch Semikolon
  - Rechenoperatoren (+-*/=|_) durch Minus
  - Öffnende Klammern durch (
  - Schließende Klammern durch)
- Zusammenfassung der benachbarten identischen zeichenweisen Ersetzungen. So wird z. B. aus dem Wort „Karo5“ zunächst „ULLLD“, was zu „ULD“ zusammengefasst wird.

XPath

HTML in seiner Baumstruktur kann wie ein Baum traversiert werden. Sogenannte XPaths definieren diese Pfade. Als einfachstes Feature dient für jeden Textknoten der Name des Blatts seines XPaths und ob es der erste des aktuellen Textknoten ist.

Part-of-Speech (POS)-Tags

Im NLP ist das POS-Tagging meist der nächste Verarbeitungsschritt nach der Tokenisierung. Jedem Token wird seine Wortart als Feature hinzugefügt.

Wortstamm

Wortstammbildung oder auch „stemming“ ist der ebenfalls beliebte Versuch beim NLP, möglichst viele Zeichen eines Tokens heuristisch schlichtweg abzuschneiden, ohne den Bedeutungsgehalt zu stark zu zerstören. Man hofft, sinnverwandte Wörter so zusammenfassen zu können, um den Recall zu erhöhen, ohne die Anzahl der Falschtreffer im selben Maße zu steigern. Für dieses Feature wird Pattern [DSD12] verwendet.

Listen

Besonders hilfreich für die Erhöhung des Recalls sind anwendungsspezifische Listen. Als binäres Feature dient daher, ob ein Token Teil einer Sequenz von Tokens ist, die exakt so in einer Liste auftaucht. Es werden folgende Listen verwendet:

- Deutsche Stoppwörter
- Firmen
- Deutsche Städte und Bundesländer
- Länder (auf Deutsch)

Weitere Listen können jederzeit hinzugefügt werden.

Visuelle Auffälligkeiten

Nicht nur die HTML-Tags implizieren Aussehen. Mit CSS-Anweisungen werden üblicherweise weitere visuelle Auffälligkeiten eingebracht. Diese Andersheit mancher Elemente soll helfen, markante Textstellen zu identifizieren. Die folgenden Features werden für jeden Textknoten verwendet:

- Angezeigt (binär)
- Schriftdicke (dünn, normal, dick)
- Schriftgröße (größer, gleich oder kleiner als die häufigste von angezeigten Tokens)
- Schriftfarbe (gleich oder anders als die häufigste von angezeigten Tokens)
- Schriftstil (normal, kursiv, schräg)

Unüberwachte POS-Tags

Aus ca. 1 Gb deutscher Stellenanzeigen wird mit dem Programm UnsuPOS [Bie06] ein unüberwacht gelernter POS-Tagger trainiert. UnsuPOS ist parameterfrei und erzeugt generisch nummerierte POS-Tags für Tokens mit einem Viterbi-Tagger. Der unüberwachte POS-Tag und die binäre Information, ob sich der Tagger sicher ist, weil er das getaggte Token kennt, werden als separate Features verwendet.

Features anderer Beobachtungen

Das CRF bietet den großen Vorteil, auch Merkmale der sequentiellen Nachbarn einer Beobachtung dieser hinzuzufügen. Sind alle Features eines Tokens erstellt, stellt sich die Frage, welche Features welcher Nachbarn zusätzlich verwendet werden sollen.

N-(P)-Gramme

Ein Spezialfall von Features anderer Beobachtungen sind N-Gramme. Für ein Token t und ein Feature f werden bestimmte Sequenzen der Länge N dieses Features als ein Feature für t hinzugefügt. Typischerweise verwendet man eine symmetrische Anzahl N-Gramme vor und hinter t. Ein N-P-Gramm sei im Folgenden die Menge der N-Gramme, die Features von höchstens P Positionen vor und nach t und alle dazwischen enthalten. Jedes bisherige, beobachtungsgebundene Feature ist also ein 1-0-Gramm.

4.2.3 Normalisierung

Die Normalisierung wird in die Bewerberqualifikationsvorhersage in Abschnitt 5.1 verlagert. Bei dessen Lösung mittels IR könnte das Ziel der Reduktion mittels Normalisierung umgekehrt auch mittels Expansion des Querys erreicht werden. Darauf wird jedoch wegen der schlechteren zu erwartenden Ergebnisse verzichtet.

4.3 Evaluation

Ziel der Evaluation ist es, zu erkennen, wie gut das CRF mit den angegebenen Features neue Stellenanzeigen taggen kann und ob eine zufriedenstellende maximal zu erwartende Genauigkeit erzielt werden kann. Dafür wird ein Trainingskorpus aufgeteilt in maximal 90% Trainingsdaten und 10% Testdaten. Für einen linear wachsenden Anteil des Trainingskorpus wird jedes mal ein Modell gelernt und auf den selben Testdaten evaluiert. Gemessen werden dabei pro Feld die Präzision

$$p = \frac{\#\text{richtig positiv}}{\#\text{klassifiziert positiv}} \tag{4.3}$$

und der Recall

$$r = \frac{\#\text{richtig positiv}}{\#\text{echt positiv}}. \tag{4.4}$$

Dabei ist #richtig positiv die Anzahl der korrekterweise als positiv (im Sinne eines Labels) klassifizierten Beobachtungen und #echt positiv die Gesamtanzahl der Beobachtungen eines Labels. Die Anzahl der vom Modell als positiv klassifizierten Beobachtungen werden mit #klassifiziert positiv angegeben.

Die Maße werden über ganze extrahierte Spannen gemessen. Das heißt, dass nur komplett richtig klassifizierte Spannen eines Feldes als richtig positiv gewertet werden. Teilweise Übereinstimmungen zählen als Fehler. Die beiden Werte werden über das harmonische Mittel zum F-Maß

$$F_\beta = (1+\beta^2) * \frac{p * r}{(\beta^2 * p) + r} \tag{4.5}$$

mit $\beta = 1$ kombiniert. Es entsteht eine sogenannte Lernkurve, an der man nicht nur erkennt, wie viele Fehler noch gemacht werden und wie viele korrekte Informationen nicht als solche erkannt werden. Sie vermittelt auch, ob die verwendeten Features und das Modell bereits ausgereizt werden, die Qualität mit mehr Trainingsdaten also nicht mehr weiter erhöht werden kann.

Alle Features werden als 1-2-Gramme, die Wortidentität, der HTML-Tag, der POS-Tag, die zusammengefasste Form und der Formentyp zusätzlich als 2-2-Gramme hinzugefügt.

4.3.1 Entfernt überwacht getaggter Datensatz

Abbildung 4.12 zeigt die Lernkurven für die initial gecrawlten Stellen des entfernt überwacht getaggten Datensatz. Location (meist Städte) funktioniert wegen der vielen Fehler und Inkonsistenzen in den Metadaten verhältnismäßig schlecht.

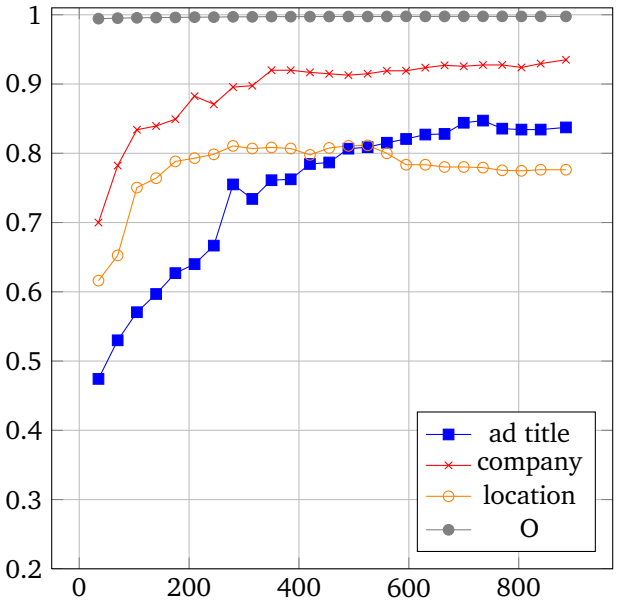

Abbildung 4.12.: Lernkurven mit 10% Testdaten vom entfernt überwacht getaggten Korpus.

4.3.2 Manuell vervollständigter Korpus

In Abbildung 4.13 werden die Lernkurven mit den manuellen Labels dargestellt. Alle Kurven sind Teil eines Durchgangs und werden nur aus Platzgründen in getrennten Diagrammen platziert. Das Diagramm links oben enthält in etwa dieselben Labels wie Abbildung 4.12. Lediglich „location“ ist unten rechts nun unterteilt in die typischen Adressbestandteile. Durch die Liste von Städtenamen als Feature kann besonders „location city“ profitieren. Außerdem scheinen die F-Werte noch nicht zu konvergieren. Die Klasse der nicht zu extrahierenden Tokens O verläuft insgesamt etwas niedriger. Dies sollte jedoch mit der enorm gewachsenen Anzahl von Klassen zu erklären sein.

Einige Felder sind sehr schwer zu lernen und erzielen Werte unter 0.5, da sie zu selten vorkommen. Auf der anderen Seite scheinen persönliche Anforderungen und die Aufgabenbeschreibung wegen des großen Vokabulars längst nicht zu konvergieren, erzielen aber dennoch bereits gute Werte ≥ 0.8.

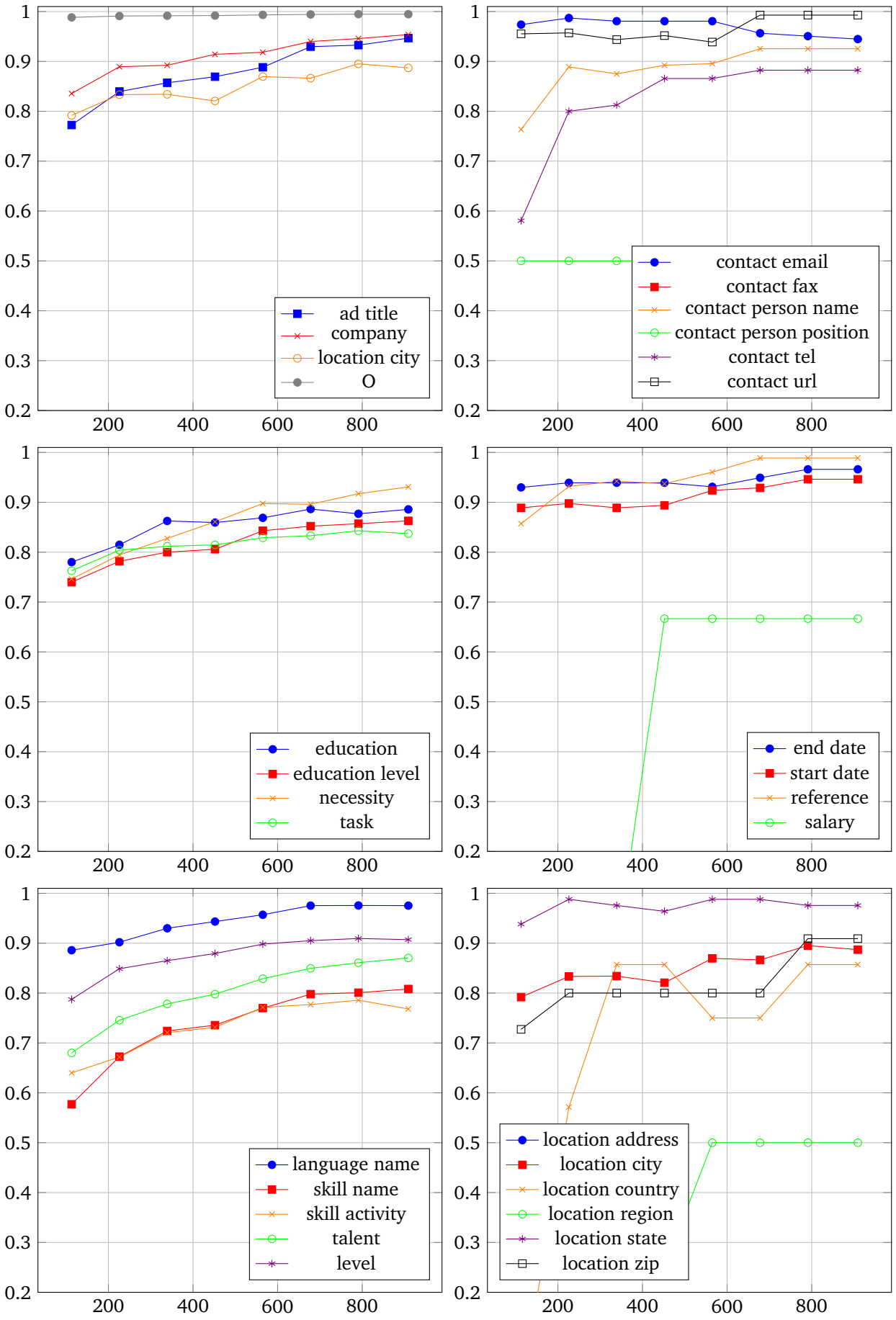

Abbildung 4.13.: Lernkurven mit 10% Testdaten vom manuell getaggten Korpus.

4.3.3 Ablation der visuellen Features

Ablation von Features ist das Wiederholen der Evaluation, wobei bestimmte Feature(-gruppen) zum Trainieren des Modells weggelassen werden. Eine wesentliche Idee in dieser Arbeit stellen die visuellen Features dar. Diese erweitern das sogenannte „Web-Scraping“ bestehend aus dem eher klassischen Web-Crawling mit reinem Extensible Markup Language (XML)-/HTML-Wrapping um die Auswertung der Informationen, wie ein Dokument dem Nutzer im Endeffekt dargestellt wird. Es werden bereits einige visuelle Auffälligkeiten modelliert und als Features verwendet. Hierzu gehört vor allem die Angabe, ob ein Token überhaupt angezeigt wird. Deren positiver Einfluss auf die Qualität der extrahierten Informationen wird in Abbildung 4.14 dargelegt. Generell kann ein Vorteil der visuellen

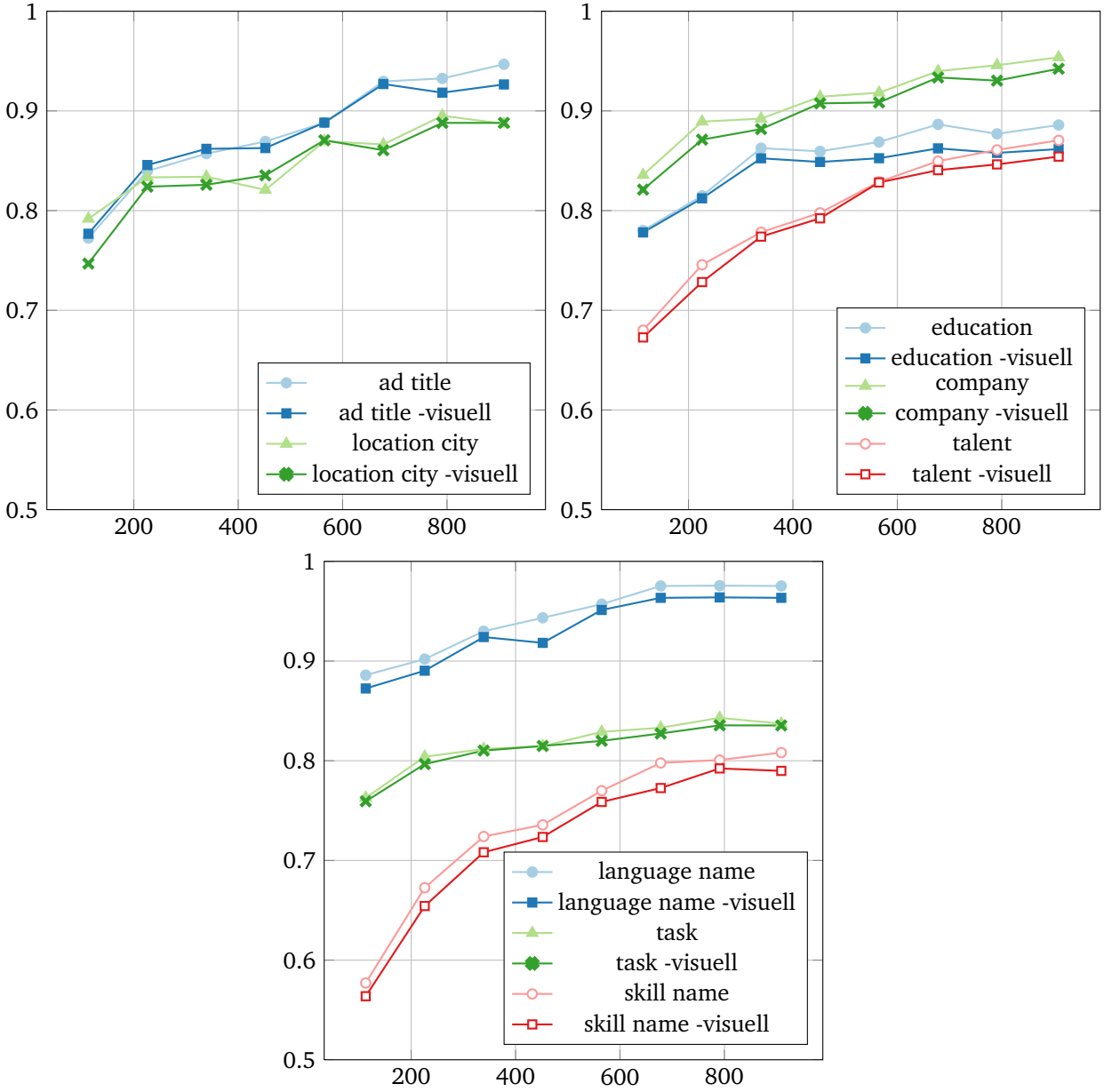

Abbildung 4.14.: Lernkurven mit 10% Testdaten vom manuell getaggten Korpus im direkten Vergleich zu ohne visuell unterstützte Features („-visuell“).

Features festgestellt werden. Häufig ist der Abstand zweier Lernkurven eines Labels bei der größten Trainingsmenge am größten. Den größten Verlust unter Ausschluss visueller Features erleidet das F-Maß bei der Erkennung von Ausbildungsanforderungen.

Um jedoch Aussagen über die Signifikanz dieser Ergebnisse treffen zu können, wird jeweils eine achtfache Kreuzvalidierung auf allen Daten – einmal mit und einmal ohne visuelle Features – durchgeführt. Bei der N-fachen Kreuz-

validierung wird ein Trainingskorpus zufällig[19] in N gleich große Partitionen eingeteilt. In Durchlauf n wird nun ein Modell mit allen Partitionen exklusive der n-ten trainiert und das Performanzmaß auf Testpartition n ermittelt.

Tabelle 4.15 gibt für jedes Label die Differenz der durchschnittlichen F-Scores über alle Kreuzvalidierungsiterationen („Folds“) an. Nur zwei Labels, nämlich die Kontakttelefonnummer und die Region werden ohne die visuellen Features besser erkannt.

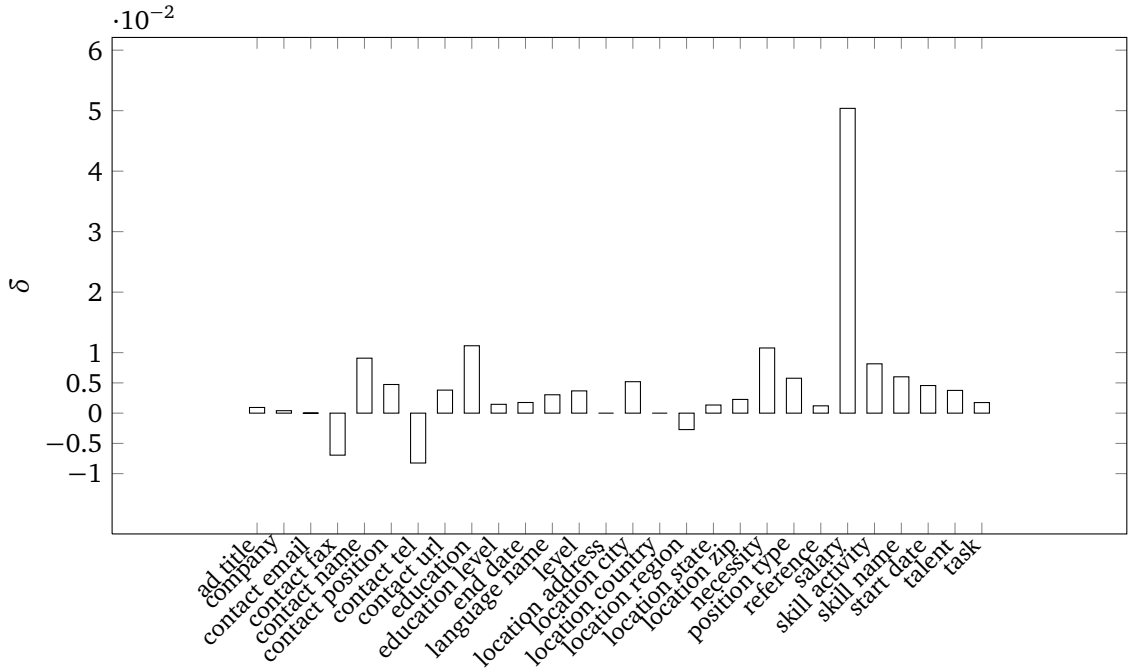

Abbildung 4.15.: Veränderung der durchschnittlichen F-Scores über alle acht Folds. Ein negativer Wert vermittelt eine Verbesserung durch die Ablation der visuellen Features.

Mit dem Wilcoxon-Vorzeichen-Rang-Test [Wil45] kann nun für jedes Label die Signifikanz der jeweiligen Abweichung bestimmt werden. Er eignet sich deswegen besonders, da wegen der niedrigen Stichprobenzahl keine zugrunde liegende Verteilung angenommen werden kann und muss. In der zweiseitigen Variante des Test wird geprüft, ob sich die Mediane der paarweisen Messungen unterscheiden. Bei der einseitigen wird getestet, ob einer der Mediane wahrscheinlich höher liegt als der andere. Die acht Differenzen eines jeden Folds werden als die Belegungen der unabhängig und identisch verteilten Stichprobenvariablen D_n angenommen. Für ein Signifikanzniveau von 5% stellt der Test sowohl zweiseitig als auch einseitig signifikante Unterschiede für die Labels Ausbildung, Notwendigkeit, Skill, und Start fest. Für Kontaktperson und Talent kann jeweils nur eine einseitige Signifikanz bestimmt werden. Die beiden Verbesserungen durch die Ablation können nicht als signifikant bestätigt werden.

Die visuellen Features werden damit zur Weiterverwendung empfohlen, da keine signifikanten Verschlechterungen und sonst nur – teilweise signifikante – Verbesserungen auftreten.

Bei der Frage, warum genau z. B. die Ausbildungsanforderungen mit visuellen Features besser funktionieren, könnte Abbildung 4.16 helfen. Sie zeigt die Unterschiede bei den visuellen Features zwischen Ausbildungstokens, anders getaggten und allen anderen Tokens. Der größte Unterschied liegt darin, ob ein Token angezeigt wird oder nicht. Ansonsten unterscheiden sich die absoluten Differenzen über Stil, Größe, Farbe und Dicke hinweg stark unterschiedlich, was auf die Trennfähigkeit anhand dieser Features schließen lässt. Für ein einzelnes visuelles Feature sind vor allem die relativen Unterschiede anhand der Positionen interessant.

[19] Jedoch jeweils mit und ohne visuelle Features gleich.

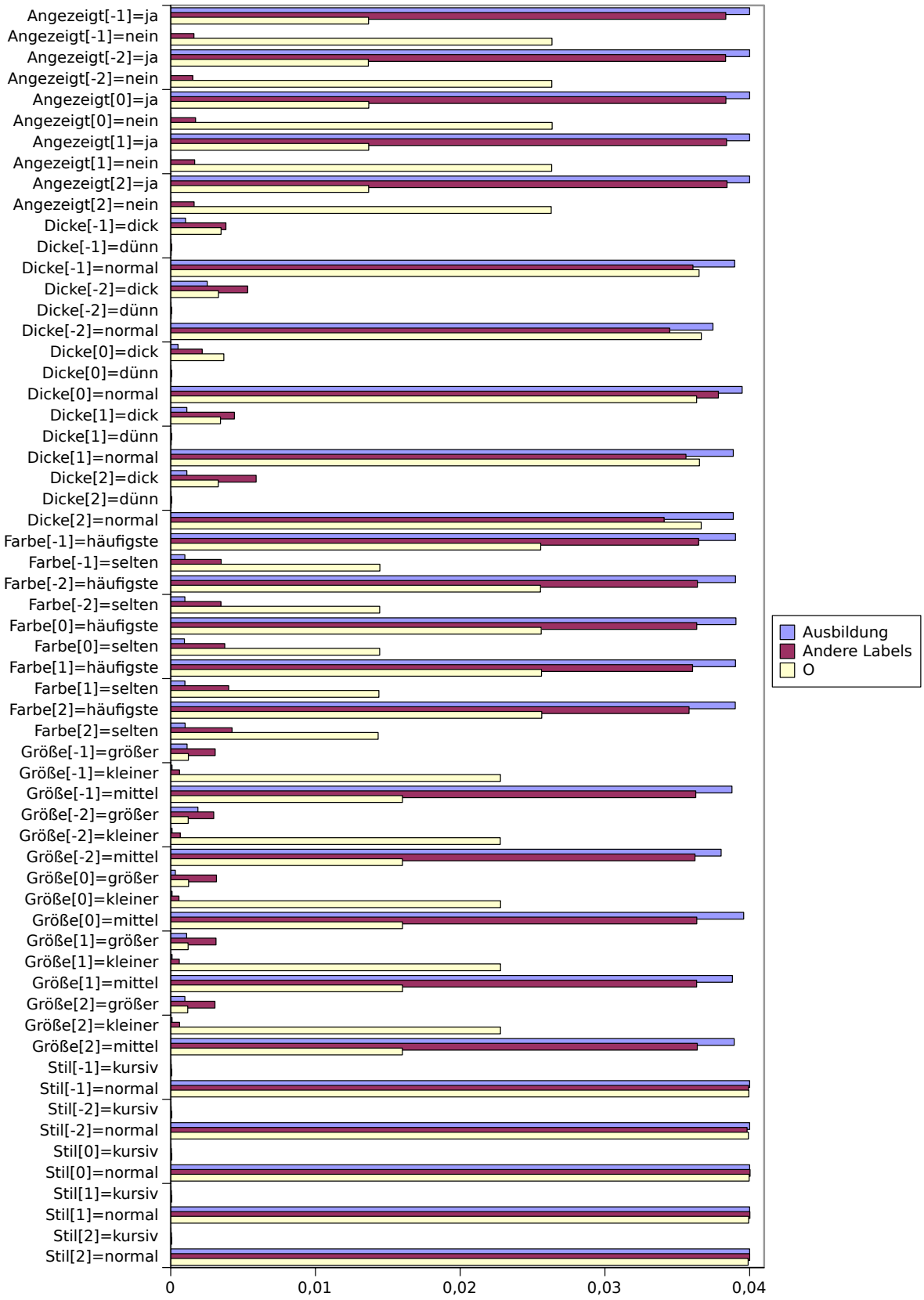

Abbildung 4.16.: Relativen Häufigkeiten der visuellen Features von als Ausbildung getaggten, anders getaggten und ungetaggten Tokens. In den eckigen Klammern steht die Position des Features, relativ zum aktuellen Token.

4.3.4 Ablation der unüberwachten POS-Tags

Für die unüberwachten POS-Tags wird die achtfache Kreuzvalidierung nach deren Ablation wie im vorangegangen Abschnitt wiederholt. Tabelle 4.17 zeigt analog dieselben durchschnittlichen Unterschiede. Sie sind hier im Schnitt deutlich größer. Wieder für das Signifikanzniveau von 5% funktionieren nach der Ablation die Felder Kontaktperson, Level, Skill, Tätigkeit, und Talent signifikant schlechter.

UnsuPOS ist also in der Lage, diese Konzepte implizit gut zu erfassen. Dessen Features sollten daher weiterhin verwendet werden, da sie nur zu signifikanten Verbesserungen und zu keinen signifikanten Verschlechterungen beitragen.

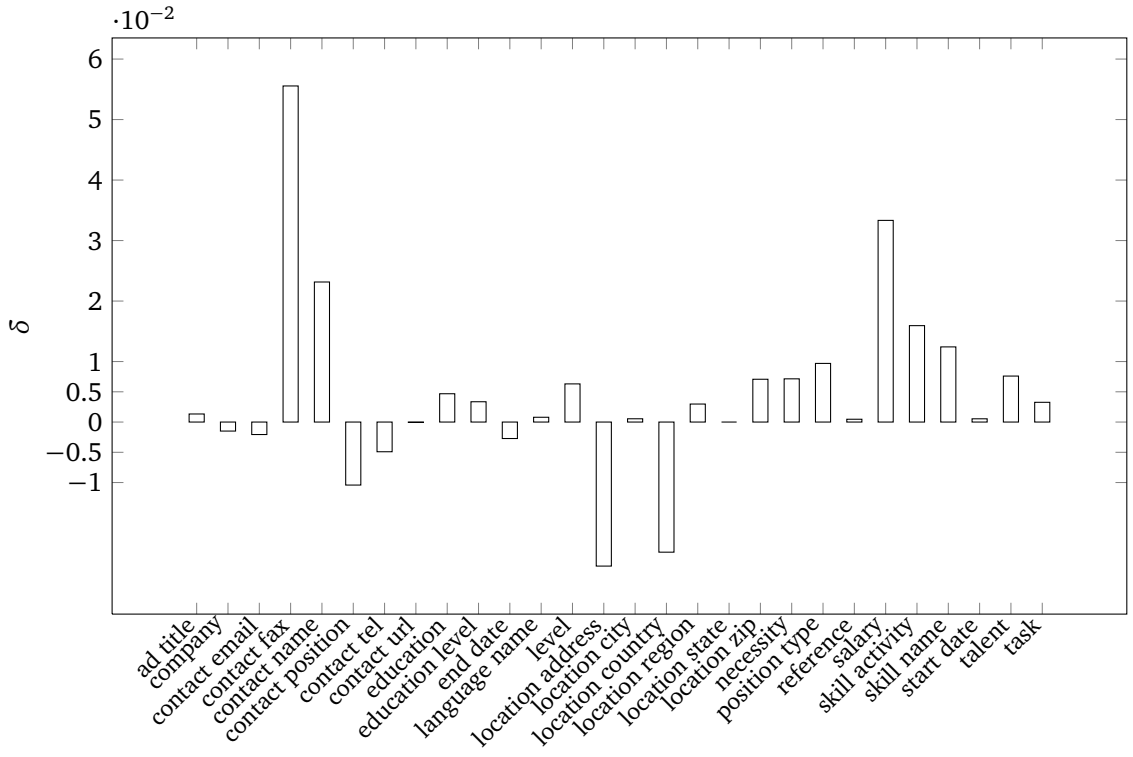

Abbildung 4.17.: Veränderung der durchschnittlichen F-Scores über alle acht Folds. Ein negativer Wert vermittelt eine Verbesserung durch die Ablation der unüberwachten POS-Tags.

5 Vorhersage der Bewerberqualifikation

Die Vorhersage der Bewerberqualifikation soll zunächst für den aktiv Arbeitssuchenden entwickelt werden. Dem Endnutzer sollen also möglichst gute Stellen vorgeschlagen werden. Diese Aufgabe wird hier als IR-Problem behandelt. Gemäß der Einleitung von [MRS08] müssen dazu drei wichtige Aspekte definiert werden. Es ist die Menge der zu durchsuchenden Dokumenteinheiten festzulegen, wie Suchanfragen als Queries formuliert und interpretiert werden und wie sich daraus eine Ergebnismenge mit relevanten Dokumenten ableitet.

Quantitativ betrachtet ist der Suchraum relativ offensichtlich und besteht aus der Menge der (getaggten) Stellenanzeigen. Qualitativ gesehen ist jedoch noch zu überlegen, wie die extrahierten Slots eingesetzt werden sollen. Wenn das ausgefüllte Nutzerprofil mit teilweise vordefinierten Eingabeoptionen als Query verwendet werden soll, bietet es sich an, die entsprechenden Slots der Stellenangebote zu normalisieren, um eine höhere Trefferquote zu erzielen. Relevante Stellen seien jene, auf die sich der Suchende bewerben würde. Dieses Informationsbedürfnis lässt sich nur schwer objektiv durch andere Personen einschätzen.

Erst mit ausreichend vielen Feedbackdaten lässt sich ein Vorschlagesystem wie in [HKV08] durch kollaboratives Filtern aufbauen. Um bei diesem datengetriebenen Entwicklungsprozess möglichst schnell Feedback zu verschiedenen Arten von Stellen zu erhalten, könnten die angezeigten Ergebnisse mit thematisch abweichenden Stellen angereichert werden.

5.1 Datengetriebener Entwicklungsprozess

Um überhaupt evaluieren zu können, wie gut Stellen dem Nutzer vorgeschlagen werden, muss eine initiale Lösung bereits so gut funktionieren, dass die Nutzer bereit sind, diese zu nutzen, und erste positive Treffer erzielen können. Damit können Paare von Stellenanzeigen und anonymisierten Nutzerprofilen erstellt werden, erweitert um den jeweiligen Hinweis vom Nutzer, ob es sich um ein relevantes Ergebnis handelt oder nicht.

Mit diesen Daten lassen sich nicht nur die Stellschrauben des IR-Systems direkt optimieren. Vor allem kollaboratives Filtern wird, mit seinem einfachen Vektormodell und der damit verbundenen geringen Notwendigkeit einer definierten Ähnlichkeitsfunktion, erst mit großer Feedbackzahl qualitativ hochwertig.

5.1.1 Überblick über die Rohdatengrundlage von Stellenanzeigen

Die Datengrundlage der Entwicklung und Evaluation von IR-Systemen zum Vorschlagen von Jobs soll ein Datensatz von 167084 Stellenanzeigen, bereitgestellt von cesar, dienen. In diesem Datensatz im XML-Format werden für jede Stelle folgende (durch cesar extrahierte) Informationen bereitgestellt:

- URL zum Originaldokument (HTML)
- Kurzbeschreibung (extrahiert)
- Ort (extrahiert)
- Firma (extrahiert)
- Anzeigenvolltext (extrahiert)

Verteilung der Tokens

Abbildung 5.1 stellt analog dieselben Beobachtungen wie die Abbildung 4.2 für den Rohdatensatz dar. Bei den Tokens zeichnet sich ein gleichmäßigeres Verhältnis zwischen sichtbaren und allen Tokens ab.

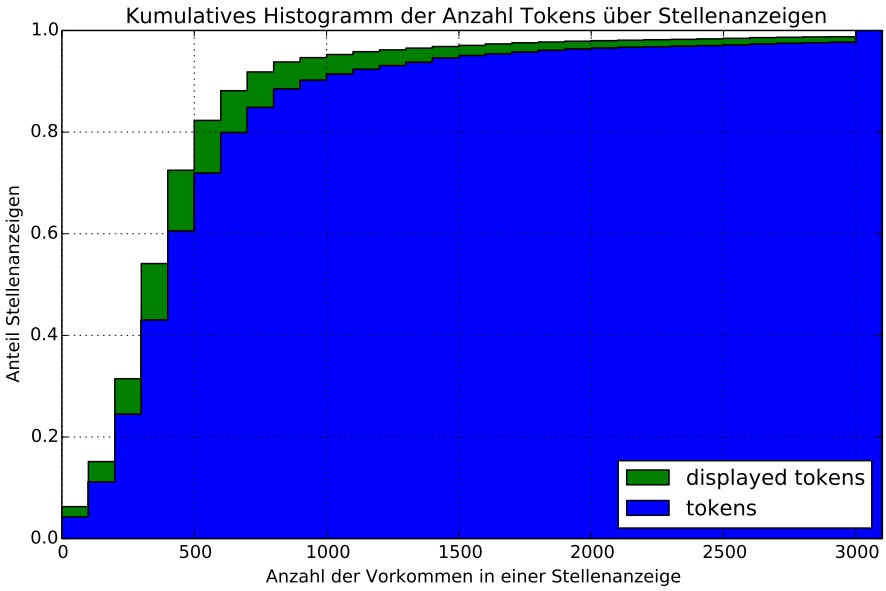

Abbildung 5.1.: Kumulatives Histogramm mit konstanter Schrittgröße 100 und unendlicher Größe des letzten Intervalls. In dem von cesar bereitgestellten Datensatz kommen Stellen mit vielen nicht angezeigten Tokens seltener als wie in Abbildung 4.2 dargestellt vor.

Vorkommen der Talente

Durchsucht man den Volltext der 167084 Stellenanzeigen nach exakt[1] diesen Talenten (Stemming[2] und Kleinschreibung wird jedoch angewandt), werden bereits 117003 Anzeigen gefunden. Ca. 70% der Stellen enthalten demzufolge mindestens eines dieser Talente. Tabelle 5.1 listet die häufigsten Talente auf. „Teamfähigkeit“ führt mit relativ großem Abstand und ist auch das am häufigsten getaggte Talent.

5.2 Erster naiver Prototyp

Um bereits während der Entwicklung des Stellen-Parsers den datengetriebenen Entwicklungsprozess zum Matching zu starten, wird mit den Stellen von cesar direkt[3] ein einfaches Suchsystem mit Haystack[4] und Solr[5] aufgebaut. Beide werden mit den Standard-Konfigurationen von Haystack in Betrieb genommen. Einzig der Solr-Stemmer und die Solr-Stoppwörter werden auf Deutsch hinzugefügt. Außerdem wird die Standardverknüpfung von Queryteilen auf `OR` gesetzt.

Ein Query wird nun aus zwei veroderten Teilen erzeugt:

- Standard Haystack AutoQuery[6], erstellt aus den Eingaben eines optionalen Suchfeldes, sucht nur im Angebotsvolltext
- Standard Solr ExtendedDisMax mit allen Einträgen aus dem Curriculum Vitae (CV), außer den in der Vergangenheit besuchten Standorten von Schule, Ausbildung, . . . , (q.alt) in allen vier Feldern (qf) gesucht, Firmen nur als Phrasen, qs = 10

Abbildung 5.2 zeigt einen Jobvorschlag für den Autor und bittet um explizites Feedback. Aber auch implizites Feedback, wie das Betrachten des Volltextes oder das Anklicken der URL, werden als subtile Interessensbekundung protokolliert.

[1] „Phrase Query“ gemäß [MRS08] in doppelten Anführungszeichen.
[2] Deutscher Snowball-Stemmer (`http://snowball.tartarus.org/algorithms/german/stemmer.html`).
[3] Es sei daran erinnert, dass hier zumindest auch die Slots Stellenbezeichnung, Ort und Firma vorhanden sind und separat durchsucht werden können.
[4] `http://haystacksearch.org/`
[5] `http://lucene.apache.org/solr/`
[6] `http://django-haystack.readthedocs.org/en/latest/inputtypes.html`

Häufigste Talente	
Talent	Anteil
Teamfähigkeit	17,3%
Flexibilität	12,2%
Engagement	9,9%
Projektmanagement	7,6%
Koordination	6,6%
Eigeninitiative	6,6%
Belastbarkeit	6,5%
Organisation	6,3%
Kundenorientierung	5,1%
Zuverlässigkeit	5,0%
Kreativität	4,4%
Durchsetzungsvermögen	4,4%
Präsentation	3,3%
Verhandlungsgeschick	3,0%
Verantwortungsbewusstsein	2,9%
Leistungsbereitschaft	2,6%
Networking	2,3%
Überzeugungskraft	1,5%
Selbstständigkeit	1,4%
Serviceorientierung	1,3%
Analytisches Denken	1,1%
Ergebnisorientierung	1,0%
Offenheit	1,0%

Tabelle 5.1.: Die häufigsten Talente, die in mindestens 1% der Stellen vorkommen.

Abbildung 5.2.: Jobvorschlag mit der Bitte um explizites Feedback.

5.3 IR anhand der extrahierten Informationen

Im Folgenden wird gezeigt, wie der erste Prototyp mit den extrahierten Feldern verbessert wird. Es folgt zunächst eine Übersicht über den komplett automatisch getaggten Datensatz. Anschließend wird erklärt, wie die zusätzlichen Informationen verwendet werden.

5.3.1 Überblick über den geparsten Korpus

Alle Anzeigen des cesar-Datensatzes wurden anhand ihrer URL heruntergeladen. Sämtliche Dokumente wurden anschließend von einem CRF – trainiert auf allen getaggten Stellen der Parsing-Evaluation aus Abschnitt 4.3.2 – getaggt. Abbildung 5.3 stellt analog dieselben Beobachtungen wie die Abbildung 4.5 für den automatisch getaggten Datensatz dar. Von den acht wichtigsten Labels werden nun deutlich mehr nie vergeben. 20% der Stellen erhalten

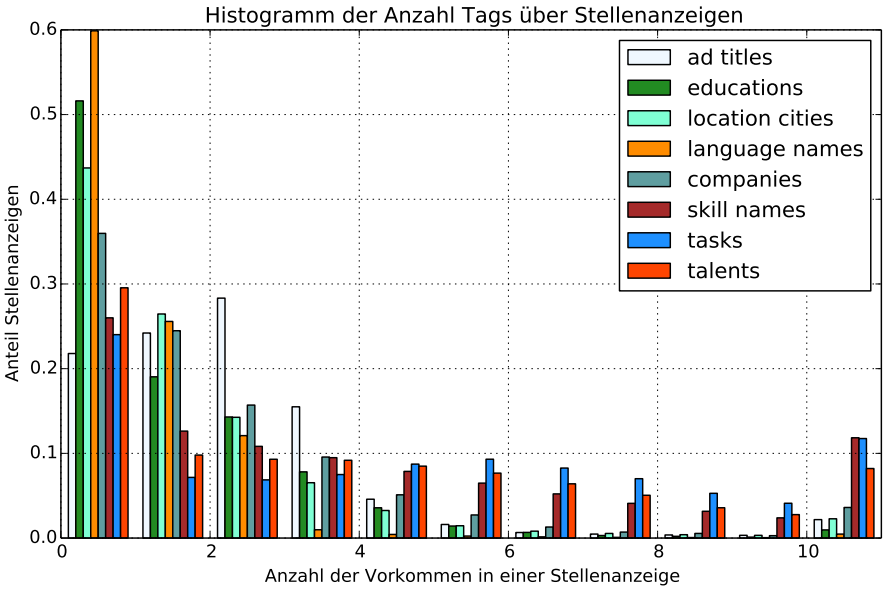

Abbildung 5.3.: Histogramm mit konstanter Schrittgröße 1 und unendlicher Größe des letzten Intervalls. Überblick über die Verteilung der wichtigsten Tags im automatisch getaggten Korpus von cesar.

keine Stellenbezeichnung, was sicherlich zum Großteil einem niedrigen Recall zu schulden ist. Es ist jedoch auch davon auszugehen, dass bei manchen Stellen das Crawling der HTML-Daten misslungen ist.

5.3.2 Normalisierung der extrahierten Informationen

Laut Abschnitt 2.2 ist die Normalisierung Teil der IE und sollte damit auch als Teil des Parsers betrachtet werden. Da es hierbei jedoch auch um die Entwicklung der zu durchsuchenden Dokumente geht und sich die Normalisierung aus den genormten Feldern des EuropassCV und der Talente heraus motiviert, wird sie in diesem Teil der Arbeit vorgestellt. Eine Expansion des Querys in verschiedene Schreibweisen der genormten CV-Felder wäre ebenfalls denkbar. Dies könnte jedoch zu ungewollten Überlappungen führen, weshalb von dieser Vorgehensweise abgesehen wird.

Die Zuweisung zwischen den standardisierten Einträgen einer Norm und deren verschiedenen Schreibweisen soll für die häufigsten Vorkommen von Schreibweisen unterstützt durch „fuzzy matching“ manuell vorgenommen werden. Es ist davon auszugehen, dass extrahierte Schreibweisen mehreren Normierungen zugewiesen werden können müssen.

Für diese Arbeit sollen Anforderungen an die Talente und gesprochenen Sprachen eines Bewerbers normalisiert werden. Dabei wird die Ähnlichkeit zum Zipfschen Gesetz betrachtet. Es wird in [MRS08] so definiert, dass die Häufigkeit cf_i des i-t häufigsten Terms (Rang i) proportional zum Kehrwert seines Rangs ist:

$$cf_i \propto \frac{1}{i}. \tag{5.1}$$

Diese Formel lässt sich in das äquivalente „power law“

$$cf_i = ci^k. \tag{5.2}$$

mit den Konstanten c und k umformulieren. Logarithmiert man beide Seiten, wird klar, dass auf logarithmischen Skalen stets eine Gerade entsteht. Abbildung 5.4 zeigt dies empirisch an den automatisch getaggten Talenten des cesar-Datensatzes. Die Messwerte scheinen sich gut durch eine Gerade annähern zu lassen.

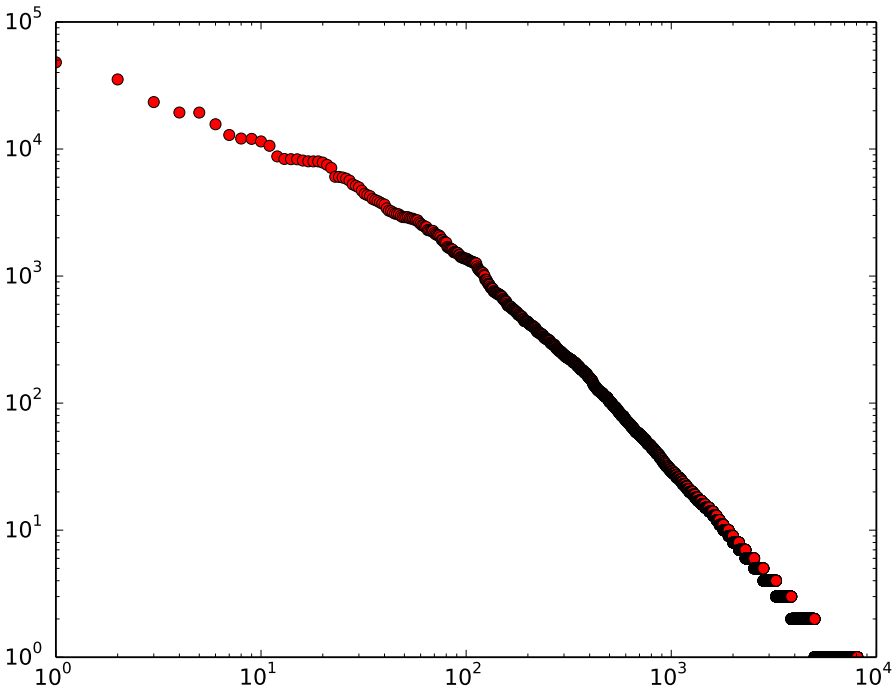

Abbildung 5.4.: Sortierte Häufigkeiten von getaggten Talenten auf logarithmischen Achsen. Die hohe Ähnlichkeit zu einer Geraden lässt näherungsweise ein „power law“ vermuten.

Da Häufigkeiten, die der Gleichung (5.1) folgen, mit steigendem Rang sehr schnell fallen, erscheint es als sinnvoll, zunächst die häufigsten Vorkommen zu normalisieren, um eine möglichst große Abdeckung zu erzielen. Insgesamt wurden 111686 Sprachen und 723612 Talente getaggt. Diese bestehen jedoch aus nur 202 bzw. 8074 einmaligen Zeichensequenzen. Werden nun die 60 häufigsten Sprachen bzw. 300 häufigsten Talente normalisiert, werden bereits 111466 (~99,8%) bzw. 633897 (~87,6%) der Vorkommen abgedeckt.

Tabelle 5.2 zeigt das einfache Datenbankschema in Form von Spaltennamen und gibt einige Beispiele an. Viele Zuweisungen sind relativ eindeutig. Manche sind jedoch komplexer und es ist nicht immer einfach, getaggten Talenten passende aus der Norm zuzuordnen. Dies ist sogar bei den Sprachen der Fall, da die Stellen meist nicht zwischen den spezielleren Sprachdetails unterscheiden. Bei den Sprachen wird der Standard von Django[7] verwendet, welcher sich wiederum am Standard ISO 639[8] orientiert.

5.3.3 Umsetzung

Tabelle 5.3 beschreibt, wie aus den einzelnen Angaben im Nutzer-CV jeweils ein ExtendedDisMax generiert wird, der wiederum mit allen anderen mit *OR* verknüpft wird. Es wird versucht, auf den jeweils zum CV passenden Feldern zu suchen. Wegen der Beschränkung auf maximal ein Label pro Token, werden hierbei Kompromisse eingegangen. So wird beispielsweise mit den Fähigkeiten eines Nutzers auch in der Aufgabenbeschreibung gesucht.

[7] `https://github.com/django/django/blob/stable/1.5.x/django/conf/global_settings.py`
[8] `http://www.iso.org/iso/language_codes`

Label	Getaggte Spanne	Normalisierung
Talent	„teamorientiertes“	Teamfähigkeit
Talent	„Teamorientierung“	Teamfähigkeit
Talent	„partnerschaftlichen“	Einfühlungsvermögen
Talent	„partnerschaftlichen“	Diplomatisches Geschick
Talent	„partnerschaftlichen“	Liebevolles Wesen
Talent	„partnerschaftlichen“	Loyalität
Talent	„partnerschaftlichen“	Emotionale Intelligenz
Sprache	„Deutsche“	Deutsch
Sprache	„Deutschkentnisse“	Deutsch
Sprache	„Chinesisch“	Vereinfachtes Chinesisch
Sprache	„Chinesisch“	Traditionelles Chinesisch

Tabelle 5.2.: Beispiele zur Normalisierung von extrahierten Informationen. Nur als gesamte Tripel sind die Zeilen eindeutig zu halten.

Nutzereingaben = q.alt	P. E.	Durchsuchte Felder qf = pf
Talente (Text)	✗	Extrahierte Talente
Talente (Codes)	✗	Normalisierte extrahierte Talente
Sprachen (Text)	✗	Extrahierte Sprachen
Sprachen (Codes)	✗	Normalisierte extrahierte Sprachen
Schulabschlüsse	✗	education_level, Volltext
Hochschularten	✗	education_level, Volltext
Studienrichtungen	✗	education, Volltext
Studienabschlüsse	✗	education_level, Volltext
Arbeitsverhältnisse	✗	position_type, Volltext
Wohnort	✗	location_*, Volltext
Suchfeld	✗	ad_title, company, location_*, task, education, talent, skill_*, position_type, language_name, education_level, Volltext
Fähigkeiten	✓	skill_name, skill_activity, task, talent, ad_title, Volltext
Gelernte Berufe	✓	education, ad_title, skill_name, skill_activity, Volltext
Studienfächer	✓	education, ad_title, Volltext
Berufserfahrungen	✓	task, skill_name, skill_activity, ad_title, Volltext
Firmen & Schulen	✓	Nur als Phrasen, company, Volltext

Tabelle 5.3.: Pro Zeile ein ExtendedDisMax wie in Abschnitt 3.3.2 unter Solr gezeigt. Die Parameter qs und pf werden jeweils auf 10 gesetzt. Die Spalte „Pro Eingabe“ gibt an, ob ein ExtendedDisMax für jede einzelne Angabe erstellt wird oder nur einer für alle Angaben mit Leerzeichen aneinander gehängt.

5.4 Evaluation

5.4.1 Metriken zur Bewertung der IR-Systeme

Da es unmöglich ist, zu bestimmen, wie viele Stellenanzeigen für eine Suchanfrage relevant sind, wird ein Evaluationsmaß benötigt, das ohne Recall auskommt. In [MRS08] werden in Kapitel 8.4 zwei solche Maße vorgestellt, nämlich „Precision@k“ und „Normalized Discounted Cumulative Gain (NDCG)“.

Ersteres ist die Anzahl der relevanten Dokumente unter den ersten k Ergebnisse geteilt durch k für jede Suchanfrage. Anschließend wird der Mittelwert über alle Queries berechnet. Laut [MRS08] ist dieses Maß jedoch nicht sehr stabil bei Vergleichen von Systemen.

Im zweiten Fall sei $\mathcal{Q}$ die Menge der getätigten Suchanfragen, k die Anzahl der besten, zurückgelieferten Vorschläge und $R(j,d)$ die Relevanz des d-t besten Resultats für Query j. Dann ist der NDCG definiert als

$$ndcg(\mathcal{Q},k) = \frac{1}{|\mathcal{Q}|}\sum_{j\in\mathcal{Q}} Z_{kj}\sum_{m=1}^{k}\frac{2^{R(j,m)}-1}{\log_2(1+m)}, \tag{5.3}$$

wobei Z_k als Maximum der inneren Summe das Maß zwischen 0 und 1 normalisiert. Die innere Summe wird gleich 0 gesetzt, wenn sich kein relevantes Dokument in den ersten k Ergebnissen befindet. Befindet sich jedoch mindestens ein relevantes Dokument ($R > 0$) in der Ergebnismenge und sind die entsprechenden Relevanzwerte absteigend sortiert, wird der Maximalwert von 1 erreicht.

Die beiden Maße komplementieren sich sehr gut, da Precision@k besonderen Wert darauf legt, möglichst viele relevante Ergebnisse in den ersten k Treffern vorzufinden, während NDCG eher auf deren Reihenfolge achtet und bereits für nur ein einziges relevantes Dokument an erster Stelle den Maximalwert von 1 vergibt.

5.4.2 Erster naiver Prototyp

Für die bislang 179 getätigten Suchanfragen $\mathcal{Q}_{179}$ und jeweils 5 bewerteten Stellen (Relevanz für „JA“ = 1, „VIELLEICHT“ = 0.5 und „NEIN“ = 0) ergibt sich ein noch verbesserungsbedürftiger, jedoch noch nicht sehr aussagekräftiger Wert für $ndcq(\mathcal{Q}_{179},5) = 0.4159$. Die weiteren Messwerte finden sich in Tabelle 5.4. Für die Nutzer, die auch auf Anfrage kein Feedback abgegeben haben, wurde nach bestem Wissen manuell bewertet. Dieser Anteil liegt bei ca. 37%.

k	NDCG@k	Precision@k
1	0.2690	0.3017
2	0.2826	0.2765
3	0.3271	0.2793
4	0.3610	0.2737
5	0.4159	0.2849

Tabelle 5.4.: NDCG@k und Precision@k für den naiven Prototyp und 179 Queries bei jeweils 5 bewerteten Ergebnissen.

5.4.3 Unterstützt durch extrahierte Informationen

Das Experiment aus Abschnitt 5.4.2 wird mit der Umsetzung mit den extrahierten Feldern aus Abschnitt 5.3.3 wiederholt. Bei schnellen Vorabtests zeigt sich, dass nun vor allem Dokumente zurückgeliefert werden, die besonders viele Stellenangebote enthalten. Hierzu zählen vor allem Dokumente von Stellenbörsen, die bis zu 25 Stellen auf einer Seite abbilden. Dieses Problem wird temporär versucht zu lösen, indem zu große Dokumente übersprungen werden. Die konkreten Grenzen liegen hier bei < 8000 Tokens und < 10 Stellentitel / Referenznummern oder bei < 800 Tokens. Wegen der neuen, anderen Suchergebnisse werden die Nutzer manuell um Feedback gebeten. Geben sie keines ab, wird erneut nach bestem Wissen manuell bewertet. Dieser Anteil beträgt hier ca. 33%. Die Ergebnisse finden sich in Tabelle 5.5.

k	NDCG@k	Precision@k
1	0.3447	0.4134
2	0.3593	0.3352
3	0.3696	0.2849
4	0.4168	0.2779
5	0.4785	0.2860

Tabelle 5.5.: NDCG@k und Precision@k für die Umsetzung mit den extrahierten Feldern und 179 Queries bei jeweils 5 bewerteten Ergebnissen.

5.4.4 Vergleichende Analyse

Mit der Suche auf den extrahierten Feldern kann die Precision@k besonders für kleine k verbessert werden. In den ersten 1–2 Ergebnissen finden sich nun also häufiger gute Vorschläge. Interessanterweise ist dies für $k \geq 3$ eher nicht der Fall. Der NDCG@k ist jedoch für jeden Wert von k größer. Dies zeigt, dass sich für größere k zumindest die Reihenfolge verbessert. Wegen der geringen Stichprobenzahl können noch keine signifikanten Aussagen getroffen werden. Es zeichnet sich jedoch ein Trend zur Verbesserung ab, obwohl viele Angebote wegen zu häufiger Stellen in einem Dokument heraus gefiltert werden.

6 Fazit und Ausblick

Fazit und Ausblick sollen für die beiden Schwerpunkte der Arbeit getrennt gegeben werden.

6.1 Fazit

Im Folgenden werden die Erkenntnisse dieser Arbeit für das Parsing und Matching getrennt zusammengefasst.

6.1.1 Parsing

Das eingangs erklärte Ziel, dem Computer beizubringen, Stellenanzeigen besser zu verstehen, kann mit dem CRF zufriedenstellend erreicht werden. Die Evaluation auf den manuell getaggten Stellen zeigen bereits gute Ergebnisse. Die visuellen Auffälligkeiten als Features zeigen einen positiven Einfluss, sodass über weitere optische Merkmale nachgedacht werden sollte. Mit Angaben über Höhe, Breite und Position eines jeden Textknoten könnte ein heuristischer Wert geschätzt werden, wie stark ein Token im kognitiven Fokus des Betrachters liegt.

6.1.2 Matching

Die beste Extraktion der wichtigen Informationen ist nahezu wertlos, wenn diese nicht auch dazu verwendet werden können, passende Stellen- oder Bewerbervorschläge zu erhalten. Zur erleichterten Vergleichbarkeit werden die vorgestellten Systeme möglichst nah an ihrer Standardkonfiguration betrieben. In der Evaluation aus Abschnitt 5.4 wird empirisch die Tendenz aufgezeigt, dass mit relativ leicht zu formulierenden Queries auf den extrahierten Feldern die zurückgelieferten Ergebnisse sowohl in Qualität als auch in Reihung verbessert werden können. Es tritt jedoch das Problem auf, dass Dokumente mit mehreren Stellenanzeigen zu gut platziert werden, da diese Anforderungen und Angaben enthalten, die auf verschiedene Personen zutreffen können.

6.2 Ausblick

Das vorgestellte System befindet sich stets in der Weiterentwicklung. Neben der Verbesserung und Erweiterung der bisherigen Ansätze und Taggingdaten, sind weitere, darüber hinausgehende Ideen zu verfolgen.

6.2.1 Parsing

Beim Tagging von Stellen kann der niedrige Recall datengetrieben weiter verbessert werden. Weitere Features und deren Auswirkungen sind zu beobachten. Weiteres manuelles Tagging sollte ebenfalls zur steten Verbesserung der F-Werte führen. Vor allem für den tatsächlichen Betrieb des Parsers müssen heterogene Stellenanzeigen aus anderen Gebieten manuell getaggt werden. In der Studie von [CDPW02] wird untersucht, wie ein bereits existierendes IE-System mit den Annotatoren zusammenarbeiten kann. Auch andere Sprachen in Stellenanzeigen sollen in Zukunft beachtet werden.

Die Aufhebung der Beschränkung auf nicht überlappende Labels wird auch in Betracht gezogen. Das resultierende Multi-Label-Problem könnte durch mehrere CRFs gelöst werden. Diese sollten jedoch nicht unabhängig voneinander sein. Hier gilt es zu untersuchen, welches Modell welche Labels übernimmt und welche Labels sich wiederum als brauchbare Features für die folgenden Modelle eignen. Diese Abhängigkeiten werden in der Arbeit von [GM05] untersucht und ausgenutzt.

Zusammengehörende Felder werden bislang nur heuristisch miteinander verknüpft. Welches Niveau jedoch tatsächlich zu welchem Skill gehört, kann in WebAnno ebenfalls annotiert werden. Diese Relationen könnten anschließend ebenfalls gelernt und evaluiert werden. Besonders interessant aber wären die Auswirkungen auf die Matching-Qualität.

In [SM06] werden „Skip-Chain“ CRFs für Dokumente vorgeschlagen, in denen die selben zu extrahierenden Informationen mehrfach vorkommen. Dies ist bei Stellenanzeigen tatsächlich öfter der Fall. Dieses Modell erweitert das linear verkettete CRF um Faktorknoten zwischen identischen Beobachtungen und kann so beispielsweise besser mehrere Vorkommen desselben Seminarsprechers extrahieren.

6.2.2 Matching

Besonders wertvoll wäre Feedback durch ausschreibende Firmen und Vermittler aus bereits vollzogenen Einstellungen. Mit dem datengetriebenen Entwicklungsprozess wird ein System zur Gewinnung immer neuer Klickdaten verbessert, um damit in Zukunft kollaboratives Filtern einzusetzen. Außerdem kann mit mehr Feedback die Evaluation stabilisiert werden, um signifikante Aussagen treffen zu können. Um den Zeitaufwand bei der manuellen Normalisierung zu verteilen, soll Crowdsourcing zum Einsatz kommen. In Zukunft soll auch der Suchprozess umgekehrt werden, sodass ausgehend von einer Stellenanzeige passende Bewerber gefunden werden. Ein bis zum Schluss verborgenes, großes Problem bleiben Dokumente mit mehreren Stellenanzeigen. Hierfür können Textsegmentierungsverfahren wie z. B. von [RB12] herangezogen werden.

Literaturverzeichnis

[AHL89] M. B. Arthur, D. T. Hall und B. S. Lawrence: *Handbook of career theory*. Cambridge University Press, Cambridge, England; New York, 1989.
Erreichbar unter: http://dx.doi.org/10.1017/CBO9780511625459.

[ANA10] KONCEPT ANALYTICS: *Global Recruitment Market Report*, 2010.
Erreichbar unter: http://www.catenon.com/inversores/pdf/Global-Recruitment-Market-Report-2010.pdf [zuletzt besucht 24.09.2013].

[AS01] R. Arnold und I. Schüßler: *Entwicklung des Kompetenzbegriffs und seine Bedeutung für die Berufsbildung und für die Berufsbildungsforschung*. In: G. Franke (Herausgeber): *Komplexität und Kompetenz. Ausgewählte Fragen der Kompetenzforschung*, Seiten 52–74. Bertelsmann, W, Bonn, 2001.

[AS02] S. Ahmad und R. G. Schroeder: *The importance of recruitment and selection process for sustainability of total quality management*. In: *International Journal of Quality & Reliability Management*, Band 19, Seiten 540–550. MCB UP Ltd, 2002.
Erreichbar unter: http://dx.doi.org/10.1108/02656710210427511, doi:10.1108/02656710210427511.

[BH06] J. P. Briscoe und D. T. Hall: *The interplay of boundaryless and protean careers: Combinations and implications*. In: *Journal of Vocational Behavior*, Band 69, Seiten 4–18. Elsevier Inc., 2006.
Erreichbar unter: http://www.sciencedirect.com/science/article/pii/S0001879105001065, doi:http://dx.doi.org/10.1016/j.jvb.2005.09.002.

[Bie06] C. Biemann: *Unsupervised part-of-speech tagging employing efficient graph clustering*. In: *Proceedings of the 21st International Conference on computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, COLING ACL '06, Seiten 7–12, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=1557856.1557859.

[Bis07] C. M. Bishop: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, Secaucus, NJ, USA, 1st ed. 2006. Corr. 2nd printing Auflage, October 2007.
Erreichbar unter: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0387310738.

[BK99] J. N. Baron und D. M. Kreps: *Strategic Human Resources: Frameworks for General Managers*. John Wiley, New York, 1999.
Erreichbar unter: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471072532.html.

[BYRN99] Ricardo A. Baeza-Yates und Berthier Ribeiro-Neto: *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[CDPW02] F. Ciravegna, A. Dingli, D. Petrelli und Y. Wilks: *User-System Cooperation in Document Annotation Based on Information Extraction*. In: *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, EKAW '02, Seiten 122–137, London, UK, 2002. Springer-Verlag.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=645362.650873.

[Cir00] F. Ciravegna: *Learning to Tag for Information Extraction from Text*. In: *Proceedings of the ECAI Workshop on Machine Learning for Information Extraction*, Berlin, Germany, 2000.
Erreichbar unter: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.33.1770&rep=rep1&type=pdf.

[Cir01a] F. Ciravegna: *Adaptive information extraction from text by rule induction and generalisation*. In: *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2*, IJCAI'01, Seiten 1251–1256, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=1642194.1642261.

[Cir01b] F. Ciravegna: *$(LP)^2$, an Adaptive Algorithm for Information Extraction from Web-related Texts*. In: *Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, Seattle, 2001.
Erreichbar unter: http://eprints.aktors.org/120/01/Atem01.pdf.

[CM99] M. E. Califf und R. J. Mooney: *Relational learning of pattern-match rules for information extraction*. In: *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, AAAI '99/IAAI '99, Seiten 328–334, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=315149.315318.

[CM03] P. T. Costa und R. R. McCrae: *Personality in Adulthood: A Five-factor Theory Perspective*. Guilford Press, New York, 2003.
Erreichbar unter: http://www.amazon.de/Personality-Adulthood-Five-Factor-Theory-Perspective/dp/1593852606.

[DO12] R. Dridan und S. Oepen: *Tokenization: returning to a long solved problem a survey, contrastive experiment, recommendations, and toolkit*. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, Seiten 378–382, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=2390665.2390750.

[DSD12] T. De Smedt und W. Daelemans: *Pattern for Python*. Journal of Machine Learning Research, 13(1):2063–2067, Juni 2012.
Erreichbar unter: http://www.clips.ua.ac.be/pages/pattern.

[EB05] R. Enggruber und C. Bleck: *Modelle der Kompetenzfeststellung im beschäftigungs- und bildungstheoretischen Diskurs – unter besonderer Berücksichtigung des Gender Mainstreaming*. Technischer Bericht, Institut für regionale Innovation und Sozialforschung (IRIS e.V.), Dresden, 2005.
Erreichbar unter: www.equal-sachsen-sozialwirtschaft.de/download/Modelle_gesamt.pdf.

[EvR03] J. Erpenbeck und L. von Rosenstiel: *Handbuch Kompetenzmessung*. Schäffer-Poeschel Verlag, Stuttgart, 2003.
Erreichbar unter: http://www.amazon.de/Handbuch-Kompetenzmessung/dp/3791021060.

[fA13] Bundesagentur für Arbeit: *Die HR-BA-XML-Schnittstelle der Bundesagentur für Arbeit*, 2013.
Erreichbar unter: http://www.arbeitsagentur.de/zentraler-Content/A04-Vermittlung/A045-Dritte/Publikation/White-paper.pdf [zuletzt besucht 21.09.2013].

[FK00] D. Freitag und N. Kushmerick: *Boosted Wrapper Induction*. In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, Seiten 577–583, Menlo Park, CA, USA, 2000. AAAI Press.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=647288.723413.

[FM00] D. Freitag und A. McCallum: *Information Extraction with HMM Structures Learned by Stochastic Optimization*. In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, Seiten 584–589, Menlo Park, CA, USA, 2000. AAAI Press.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=647288.723414.

[FR92] E. A. Fleishman und M. E. Reilly: *Handbook of Human Abilities. Definitions, Measurements, and Job Task Requirements*. Consulting Psychologists Press, Palo Alto, CA, 1992.

[Fre98] D. Freitag: *Multistrategy Learning for Information Extraction*. In: *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, Seiten 161–169, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=645527.657302.

[GC04] D. Guest und N. Conway: *Employee Well-being and the Psychological Contract*. CIPD, London, 2004.

[Glo13] S. Glotzbach: *Silbentrennung durch Crowdsourcing und Möglichkeiten Userinterfaces bei Crowdflower anzupassen*. In: *Text Anlytics: Crowdsourcing*, Darmstadt, Germany, 2013. TU Darmstadt.
Erreichbar unter: http://www.ukp.tu-darmstadt.de/fileadmin/user_upload/Group_UKP/teaching/TA2012/SG_paper-final.pdf.

[GM05] N. Ghamrawi und A. McCallum: *Collective multi-label classification*. In: *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, Seiten 195–200, New York, NY, USA, 2005. ACM.
Erreichbar unter: http://doi.acm.org/10.1145/1099554.1099591, doi:10.1145/1099554.1099591.

[HKV08] Y. Hu, Y. Koren und C. Volinsky: *Collaborative Filtering for Implicit Feedback Datasets*. In: *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, Seiten 263–272, Washington, DC, USA, 2008. IEEE Computer Society.
Erreichbar unter: http://dx.doi.org/10.1109/ICDM.2008.22, doi:10.1109/ICDM.2008.22.

[HLLP13] W. Hong, L. Li, T. Li und W. Pan: *iHR: an online recruiting system for Xiamen Talent Service Center*. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '13, Seiten 1177–1185, New York, NY, USA, 2013. ACM.
Erreichbar unter: http://doi.acm.org/10.1145/2487575.2488199, doi:10.1145/2487575.2488199.

[HMTK12] M. Hasan, A. Mueen, V. Tsotras und E. Keogh: *Diversifying query results on semi-structured data*. In: *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, Seiten 2099–2103, New York, NY, USA, 2012. ACM.
Erreichbar unter: http://doi.acm.org/10.1145/2396761.2398581, doi:10.1145/2396761.2398581.

[JSZ06] R. Jin, L. Si und C. Zhai: *A study of mixture models for collaborative filtering*. Inf. Retr., 9(3):357–382, Juni 2006.
Erreichbar unter: http://dx.doi.org/10.1007/s10791-006-4651-1, doi:10.1007/s10791-006-4651-1.

[KP09] O. Koppel und A. Plünnecke: *Fachkräftemangel in Deutschland: Bildungsökonomische Analyse, politische Handlungsempfehlungen, Wachstums- und Fiskaleffekte*. Deutscher Instituts-Verlag, Köln, 2009.
Erreichbar unter: http://www.iwkoeln.de/de/studien/iw-analysen/beitrag/62521.

[KWD97] N. Kushmerick, D. S. Weld und R. Doorenbos: *Wrapper Induction for Information Extraction*. In: *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, Seiten 729–735, Nagoya, Aichi, Japan, 1997. Morgan-Kaufmann.
Erreichbar unter: http://citeseer.uark.edu:8080/citeseerx/viewdoc/summary?doi=10.1.1.7.849.

[LBC+10] R. Loth, D. Battistelli, F. Chaumartin, H. de Mazancourt, J. Minel und A. Vinckx: *Linguistic information extraction for job ads (SIRE project)*. In: *Adaptivity, Personalization and Fusion of Heterogeneous Information*, RIAO '10, Seiten 222–224, Paris, France, France, 2010. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=1937055.1937114.

[LCY08] R. Levering, M. Cutler und L. Yu: *Using Visual Features for Fine-Grained Genre Classification of Web Pages*. In: *Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, HICSS '08, Seiten 131–, Washington, DC, USA, 2008. IEEE Computer Society.
Erreichbar unter: http://dx.doi.org/10.1109/HICSS.2008.488, doi:10.1109/HICSS.2008.488.

[LCY10] T. Lavergne, O. Cappé und F. Yvon: *Practical Very Large Scale CRFs*. In: *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, Seiten 504–513, Uppsala, Sweden, July 2010. Association for Computational Linguistics.
Erreichbar unter: http://www.aclweb.org/anthology/P10-1052.

[LMP01] J. D. Lafferty, A. McCallum und F. C. N. Pereira: *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. In: *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, Seiten 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=645530.655813.

[MB05] R. J. Mooney und R. Bunescu: *Mining knowledge from text using information extraction*. SIGKDD Explor. Newsl., 7(1):3–10, Juni 2005.
Erreichbar unter: http://doi.acm.org/10.1145/1089815.1089817, doi:10.1145/1089815.1089817.

[MBSJ09] M. Mintz, S. Bills, R. Snow und D. Jurafsky: *Distant supervision for relation extraction without labeled data*. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, Seiten 1003–1011, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=1690219.1690287.

[MC10] A. Mejer und K. Crammer: *Confidence in structured-prediction using confidence-weighted models*. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, Seiten 971–981, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=1870658.1870753.

[McC05] A. McCallum: *Information Extraction: Distilling Structured Data from Unstructured Text*. Queue, 3(9):48–57, November 2005.
Erreichbar unter: http://doi.acm.org/10.1145/1105664.1105679, doi:10.1145/1105664.1105679.

[MN03] R. J. Mooney und U. Y. Nahm: *Text Mining with Information Extraction*. In: W. Daelemans, T. du Plessis, C. Snyman und L. Teck (Herausgeber): *Multilingualism and Electronic Language Management: Proceedings of the 4th International MIDP Colloquium*, Seiten 141–160. Van Schaik: South Africa, Bloemfontein, South Africa, September 2003.
Erreichbar unter: http://www.cs.utexas.edu/users/ai-lab/?nahm:midp03.

[MRS08] C. D. Manning, P. Raghavan und H. Schütze: *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
Erreichbar unter: http://nlp.stanford.edu/IR-book/.

[Nah04] U. Y. Nahm: *Text mining with information extraction*. Dissertation, The University of Texas at Austin, Austin, TX, USA, 2004. AAI3143436.
Erreichbar unter: http://www.cs.utexas.edu/users/ml/papers/discotex-dissertation-04.pdf.

[NM00] U. Y. Nahm und R. J. Mooney: *A Mutually Beneficial Integration of Data Mining and Information Extraction*. In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000*, Seiten 627–632, Austin, TX, USA, July 2000. AAAI Press.
Erreichbar unter: http://www.cs.utexas.edu/users/ai-lab/?nahm:aaai00.

[Oka07] N. Okazaki: *CRFsuite: a fast implementation of Conditional Random Fields (CRFs)*, 2007.
Erreichbar unter: http://www.chokkan.org/software/crfsuite/.

[Rab90] L. R. Rabiner: *A tutorial on hidden Markov models and selected applications in speech recognition*. In: Alex Waibel und Kai-Fu Lee (Herausgeber): *Readings in speech recognition*, Seiten 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=108235.108253.

[Ras06] D. Rastetter: *Kompetenzmodelle und die Subjektivierung von Arbeit*. In: G. Schreyögg und P. Conrad (Herausgeber): *Management von Kompetenz*, Seiten 163–199. Gabler, Wiesbaden, 2006.
Erreichbar unter: http://dx.doi.org/10.1007/978-3-8349-9300-7_5, doi:10.1007/978-3-8349-9300-7_5.

[RB12] M. Riedl und C. Biemann: *Text Segmentation with Topic Models*. JLCL, 27(1):47–69, 2012.
Erreichbar unter: http://www.jlcl.org/2012_Heft1/jlcl2012-1-3.pdf.

[Rid99] H. G. Ridder: *Personalwirtschaftslehre*. W. Kohlhammer Verlag, Stuttgart, 1999.
Erreichbar unter: http://www.amazon.de/Personalwirtschaftslehre-Hans-Gerd-Ridder/dp/3170192957.

[Sch78] E. H. Schein: *Career dynamics: matching individual and organizational needs*. Addison-Wesley Pub. Co., Reading, MA, USA, 1978.

[Sch99] R. E. Schapire: *A brief introduction to boosting*. In: *Proceedings of the 16th international joint conference on Artificial intelligence - Volume 2*, IJCAI'99, Seiten 1401–1406, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.
Erreichbar unter: http://dl.acm.org/citation.cfm?id=1624312.1624417.

[SCM99] S. Soderland, C. Cardie und R. Mooney: *Learning Information Extraction Rules for Semi-structured and Free Text*. In: *Machine Learning*, Band 34, Seiten 233–272, Hingham, MA, USA, 1999. Kluwer Academic Publishers.
Erreichbar unter: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.8809.

[SKKR01] B. Sarwar, G. Karypis, J. Konstan und J. Riedl: *Item-based collaborative filtering recommendation algorithms*. In: *Proceedings of the 10th international conference on World Wide Web*, WWW '01, Seiten 285–295, New York, NY, USA, 2001. ACM.
Erreichbar unter: http://doi.acm.org/10.1145/371920.372071, doi:10.1145/371920.372071.

[SM06] C. Sutton und A. Mccallum: *An introduction to conditional random fields for relational learning*. In: L. Getoor und B. Taskar (Herausgeber): *Introduction to Statistical Relational Learning*. The MIT Press, MA, USA, 2006.
Erreichbar unter: http://www.cs.umass.edu/~{}mccallum/papers/crf-tutorial.pdf.

[SM12] C. Sutton und A. McCallum: *An Introduction to Conditional Random Fields*. Foundations and Trends in Machine Learning, 4(4):267–373, 2012.
Erreichbar unter: http://dblp.uni-trier.de/db/journals/ftml/ftml4.html#SuttonM12.

[SS99] K. Sonntag und N. Schaper: *Förderung beruflicher Handlungskompetenz*. In: K. Sonntag (Herausgeber): *Personalentwicklung in Organisationen*, Seiten 211–244. Hogrefe-Verlag, Göttingen, 1999.
Erreichbar unter: http://www.amazon.de/Personalentwicklung-Organisationen-Karlheinz-Sonntag/dp/3801712125.

[SSR04] K. Sonntag und C. Schmidt-Rathjens: *Kompetenzmodelle als Erfolgsfaktoren*. Personalführung, 10:18–26, 2004.

[Wei11] T. Weitzel: *Recruiting Trends*. In: *Symposium für Personaler*, Band 9, Seite 2, Frankfurt, 2011.
Erreichbar unter: http://de.amiando.com/eventResources/4/Y/gUarLBBiNnNgUu/Vortrag_Tim_Weitzel.pdf.

[Wil45] F. Wilcoxon: *Individual Comparisons by Ranking Methods*. Biometrics Bulletin, 1(6):80–83, 1945.
Erreichbar unter: http://dx.doi.org/10.2307/3001968, doi:10.2307/3001968.

[WW07] D. H. Widyantoro1 und Y. Wibisono: *Information Extraction for E-Job Marketplace*. In: *Proceedings of the 4th International Conference TSSA*, Bandung, Indonesien, 2007.
Erreichbar unter: http://fpmipa.upi.edu/staff/yudi/tssa-2007-dwi-yudi-information-extraction-fo
pdf.

[XWD+12] C. Xiong, T. Wang, W. Ding, Y. Shen und T. Liu: *Relational click prediction for sponsored search*. In: *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, Seiten 493–502, New York, NY, USA, 2012. ACM.
Erreichbar unter: http://doi.acm.org/10.1145/2124295.2124355, doi:10.1145/2124295.2124355.

[YGdCB13] S. M. Yimam, I. Gurevych, R. E. de Castilho und C. Biemann: *WebAnno: A Flexible,Web-based and Visually Supported System for Distributed Annotations*. In: *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, Seiten 1–6, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.
Erreichbar unter: https://code.google.com/p/webanno/.

Anhang

A URLs in Fußnoten

Sämtliche URLs in Fußnoten wurden zuletzt am 23.09.2013 besucht.

B Taggingrichtlinien

Tabelle B.1 erweitert die Taggingrichtlinien aus Tabelle 4.3 um Regelauszüge und Beispiele der verbleibenden Labels.

Label	**Richtline (Auszug)**	**Beispiel**
Contact email	Email-Adresse markieren	„tobias[at]kroenke[punkt]de“ = contact_email
Contact fax	Inkl. Ländercode und „+“	„+49(123)4567890“ = contact_fax
Contact name	Inkl. Anrede	„Herr Tobias Kroenke“ = contact_name
Contact position	Position, die Kontaktperson innehält	„Herr Tobias Kroenke (Recruiter)“ →„Recruiter“ = contact_position
Contact tel	Inkl. Ländercode und „+“	„+49(123)4567890“ = contact_tel
Contact url	Inkl. Protokoll	„Weitere interessante Stellenangebote finden Sie unter https://mecruiting.de“ →„https://mecruiting.de“ = contact_url
Education level	Ausbildungsniveaus	„Abgeschlossenes Studium“, „Ausbildung“, „Hochschulstudium“ = education_level
End date	Bis wann die Stelle zu auszuüben ist	„Projektende September 2013“ →„September 2013“ = end_date
Level	• Niveau / Ausprägung einer Eigenschaft • Muss in Verbindung mit skill_name, talent oder language_name stehen	„Fundierte Berufserfahrung im Bereich der Schüttguttechnik“ →„Fundierte Berufserfahrung“ = level →„Schüttguttechnik“ = skill_name
Location address	Adresszeile	„Luisenplatz 3“ = location_address
Location country	Land	„Deutschland“ = location_country
Location state	Bundesland	„Hessen“ = location_state
Location region	Meist Regionen	„weltweit“ = location_region
Location zip	PLZ	„64283“ = PLZ
Necessity	Notwendigkeit einer Eigenschaft	„von Vorteil“, „wünschenswert“ = necessity
Position type	Art & Umfang der zu besetzenden Position	„Reisebereitschaft“, „Vollzeit“ = position_type
Reference	Referenznummer	„ABC1234567890“ = reference
Salary	Gehalt	„8€ / h“ = salery
Skill activity	Wie eine Fähigkeit angewandt wird	„Erfahrung im Umgang mit SAP“ →„Erfahrung“ = level →„Umgang“ = skill_activity →„SAP“ = skill_name
Start date	Ab wann die Stelle zu besetzen ist	„Ab sofort“ = start_date

Tabelle B.1.: Auszug aus den Taggingrichtlinien der verbleibenden Labels mit Beispielen.