EXTRAKTION GEOGRAPHISCHER ENTITÄTEN ZUR SUCHE NUTZERGENERIERTER INHALTE FÜR NACHRICHTENEREIGNISSE

DISSERTATION

Zur Erlangung des akademischen Grades Doktoringenieur (Dr.-Ing.)

Vorgelegt an der
Technischen Universität Dresden,
Fakultät Informatik

Eingereicht von

DIPL.-MEDIENINF. PHILIPP KATZ

Geboren am 15. Oktober 1982 in Heilbronn-Neckargartach

Gutachter:
Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill, TU Dresden
Prof. Dr.-Ing. Martin Gaedke, TU Chemnitz
Sen.-Prof. Dr.-Ing. habil. Uwe Petersohn, TU Dresden

Tag der Verteidigung: 22. Oktober 2014
Dresden, im Oktober 2014

DANKSAGUNG

Diese Dissertation entstand während meiner Zeit am Lehrstuhl Rechnernetze der Technischen Universität Dresden. Professor Schill danke ich für die Betreuung der Arbeit, vor allem für die richtige Kombination aus regelmäßigen Deadlines, gewinnbringenden Kommentaren und der eingeräumten Geduld, mein Forschungsthema zu finden. Ebenfalls danke ich Professor Petersohn und Professor Gaedke für die hilfreichen Anmerkungen.

Bei sämtlichen Kollegen des Lehrstuhls möchte ich mich für die angenehme Zeit bedanken. Über die letzten Jahre sind hier viele wertvolle Freundschaften entstanden. Dank Marius Feldmann bin ich im Jahre 2009 als Student mit dem Lehrstuhl Rechnernetze in Kontakt gekommen. Seinem unermüdlichen Einsatz ist es zu verdanken, dass ich in zwei Forschungsprojekten unterkam, die meine Dissertation erst ermöglichten. Außerdem stand er mir während meiner kompletten Dissertation tatkräftig und mit wertvollen Ratschlägen zur Seite. Der regelmäßige Austausch mit der Internet-Information-Retrieval-Gruppe hat mir ungemein geholfen, fokussiert und in schwierigen Zeiten weiter am Ball zu bleiben. Zu nennen sind hier insbesondere David Urbansky, Klemens Muthmann, Miranda Ackerman und Daniel Esser. Die Arbeit an unseren gemeinsamen Projekten hat mir viel Spaß gemacht und ich hoffe auf eine weitere enge Zusammenarbeit mit Euch in der Zukunft!

Abschließend gilt mein allergrößter Dank meiner Familie, insbesondere meinen Eltern, die mich auf meinem Lebensweg fortwährend unterstützt haben, mir dabei immer große Freiheiten bei meinen Entscheidungen ließen und Verständnis dafür zeigten, dass ich mich während der Dissertationsphase immer mal wieder rarmachen musste.

ZUSAMMENFASSUNG

Der Einfluss sogenannter nutzergenerierter Inhalte im Web hat in den letzten Jahren stetig zugenommen. Auf Plattformen wie Blogs, sozialen Netzwerken oder Medienportalen werden durch Anwender kontinuierlich Textnachrichten, Bilder oder Videos publiziert. Auch Inhalte, die aktuelle gesellschaftliche Ereignisse, wie beispielsweise den Euromaidan in Kiew dokumentieren, werden durch diese Plattformen verbreitet. Nutzergenerierte Inhalte bieten folglich das Potential, zusätzliche Hintergrundinformationen über Ereignisse direkt vom Ort des Geschehens zu liefern.

Diese Arbeit verfolgt die Vision einer Nachrichtenplattform, die unter Verwendung von Methoden des Information Retrievals und der Informationsextraktion Nachrichtenereignisse erkennt, diese automatisiert mit relevanten nutzergenerierten Inhalten anreichert und dem Leser präsentiert.

Zur Suche nutzergenerierter Inhalte kommen in dieser Arbeit maßgeblich geographische Entitäten, also Ortsbezeichnungen zum Einsatz. Für die Extraktion dieser Entitäten aus gegebenen Nachrichtendokumenten stellt die Arbeit verschiedene neue Methoden vor. Die Entitäten werden genutzt, um zielgerichtete Suchanfragen zu erzeugen. Es wird gezeigt, dass sich eine geounterstützte Suche für das Auffinden nutzergenerierter Inhalte besser eignet als eine konventionelle schlüsselwortbasierte Suche.

INHALTSVERZEICHNIS

1 EINLEITUNG

In den letzten Jahren hat sich das Web zu einem immer dynamischeren und interaktiveren Medium entwickelt. Im Zuge des Web 2.0 gewannen nutzergenerierte Inhalte (engl. „User-Generated Content"), die durch Nutzer auf verschiedenen Plattformen wie Blogs, sozialen Netzwerken oder Medienportalen publiziert werden, wachsende Bedeutung. Aufgrund der steigenden Verbreitung mobiler Endgeräte wie Smartphones oder Tablets in Kombination mit fast allgegenwärtigen Möglichkeiten des mobilen Internetzugangs wird eine täglich wachsende Menge an Textnachrichten, Bildern und Videos erzeugt und auf einer Vielzahl von Plattformen veröffentlicht. Statistiken von Twitter[1] aus dem Jahre 2011 beziffern die durchschnittliche Anzahl täglich versendeter Tweets auf 140 Mio. mit Spitzenwerten von fast 7.000 Tweets pro Sekunde[2]. Auf der Plattform Instagram[3] werden aktuell durchschnittlich jeden Tag 55 Mio. Bilder hochgeladen[4], auf Flickr[5] sind es täglich über 1,5 Mio.[6]

Im Zuge der Berichterstattung zu aktuellen Ereignissen können solche Inhalte von großem Interesse sein, da sie hochaktuelle Informationen aus erster Hand liefern. Abbildung 1.1 zeigt beispielsweise ein Foto[7] auf Instagram, welches unmittelbar nach dem Flugzeugunglück des Asiana Airlines Flug 214 am 6. Juli 2013 auf dem Flughafen in San Francisco aufgenommen wurde.

1.1 MOTIVATION

Die Beteiligung des Bürgers an journalistischer Berichterstattung, auch als „Graswurzel-Journalismus" oder „partizipativer Journalismus" bezeichnet, existierte bereits vor der Ära des Internets (S. Allan und Thorsen, 2009). Durch dessen zunehmende Verbreitung erlebte die Idee jedoch einen spürbaren Bedeutungsgewinn (Jurrat, 2011). Einschneidende gesellschaftliche und politische Ereignisse in jüngerer Zeit, wie beispielsweise der arabische Frühling, die Proteste im Gezi-Park in Istanbul oder der Euromaidan in Kiew, wurden maßgeblich durch soziale Medien und Plattformen begleitet.

1 https://twitter.com
2 https://blog.twitter.com/2011/numbers
3 http://instagram.com
4 http://instagram.com/press/
5 https://www.flickr.com
6 http://www.flickr.com/photos/franckmichel/6855169886/
7 http://instagram.com/p/bb3enbIiy6/

Abbildung 1.1: Beispielbild von Instagram, welches den Flugzeugunfall des Asiana Airlines Flugs 214 am 6. Juli 2013 auf dem Flughafen in San Francisco zeigt

Nutzergenerierte Inhalte bieten das Potential, zusätzlich zu der Berichterstattung auf klassischen Kanälen, einen alternativen Blick aus anderen Perspektiven auf Geschehnisse zu erlauben, der von offiziellen Medien – bewusst oder unbewusst – nicht geliefert werden.

Der wachsende Einfluss nutzergenerierter Inhalte macht sich vor allem bei jüngeren Internetnutzern bemerkbar. Eine aktuelle Studie von Crowdtap (2014) zeigt, dass diese nutzergenerierte Inhalte vertrauenswürdiger und einprägsamer empfinden als Informationen aus klassischen Medien.

Wollen Anwender für ein gegebenes Ereignis gezielt nach nutzergenerierten Inhalten suchen, so muss dies zum Status quo jedoch maßgeblich manuell geschehen. Quellen wie Twitter, Flickr, YouTube oder Instagram müssen unter Verwendung passender Suchbegriffe gezielt nach den gewünschten Informationen durchsucht werden. Die dort gefundenen Resultate stehen jeweils für sich isoliert und verteilt über die verschiedenen Plattformen. Die thematische Relevanz der Ergebnisse muss durch den Anwender selbst erkannt und beurteilt werden.

Nachrichtenquellen wie CNN kombinieren bereits nutzergenerierte Inhalte mit ihrer eigenen Berichterstattung. In manchen Artikeln und besonders auf der Unterplattform iReport[8] werden Nutzerbeiträge von Quellen wie Twitter, YouTube[9] oder Flickr zusammengetragen. Die Web-Plattform Storify[10] zielt ausschließlich darauf ab, Nutzern die Möglichkeit zu geben, Inhalte auf verschiedenen Social-Media-Seiten zu suchen, zu „Stories“ zusammenzufassen und gesammelt zu präsentieren. Diese Plattform wird gerade auch im nachrichtenrelevanten Bereich genutzt, indem Anwender oder

8 http://ireport.cnn.com
9 https://www.youtube.com
10 http://storify.com

auch Nachrichtenagenturen eigene und fremde Inhalte miteinander verknüpfen. Die Seite „Breaking News“[11] ist ein Startup-Projekt innerhalb des NBC Networks, dessen Ziel es ist, eine neuartige Echtzeitplattform für Nachrichten zu schaffen. Diese Plattform wurde zunächst hauptsächlich auf Basis von Twitter aufgebaut und wird manuell von Journalisten und einer Gruppe ausgewählter Partner gepflegt, die Beiträge beispielsweise in Form von Tweets, Bildern oder Links beisteuern können. In sämtlichen dieser Beispiele werden die nutzergenerierten Inhalte jedoch auf manuellem Wege zusammengesucht.

Diese Arbeit setzt sich zum Ziel, unter Verwendung geographischer Entitäten, die aus Nachrichtentexten gegebener Ereignisse extrahiert werden, eine zielgerichtete Suche nach multimedialen nutzergenerierten Inhalten mit hoher Zuverlässigkeit vorzunehmen. Die Vision ist eine Nachrichtenplattform, die ähnlich wie Google News[12], Columbia NewsBlaster[13] (McKeown et al., 2002), NewsStand[14] (Teitler et al., 2008) oder dem Europe Media Monitor[15] (Steinberger, Pouliquen und van der Goot, 2009) Inhalte etablierter Nachrichtenquellen aggregiert. Die aggregierten Nachrichten sollen jedoch automatisiert um nutzergenerierte Inhalte in Form von Texten, Bildern und Videos angereichert werden, um Lesern einen Mehrwert durch zusätzliche Hintergrundinformationen und alternative Sichtweisen auf Ereignisse zu bieten. Für diese Vision liefert die vorliegende Arbeit essentielle Bausteine.

Der im Journalismus übliche 5W1H-Stil (Flint, 1917) besagt, dass Nachrichtenmeldungen so strukturiert sein sollen, dass die sechs Fragen nach dem „Wer“, „Was“, „Wo“, „Wann“, „Warum“ und „Wie“ (engl. „Who“, „What“, „Where“, „When“, „Why“ und „How“) beantwortet werden. Eine Reihe von Forschungsinitiativen wie die MUC (Message Understanding Conference) und die ACE (Automated Content Extraction) haben untersucht, wie diese Informationen automatisiert aus Artikeln extrahiert werden können (Doddington et al., 2004; Grishman und Sundheim, 1996). Ortsbeschreibungen decken zwar nur eine dieser sechs Fragen ab, ihre Relevanz zeigt jedoch die folgende Betrachtung: Über den RSS-Feed von CNN[16] wurden 105 Nachrichtenartikel akquiriert und mittels AlchemyAPI[17] Entitäten der Typen „Person“, „Organisation“ und „Lokation“ aus den Texten extrahiert[18]. Mehrfachnennungen innerhalb eines Artikels wurden ignoriert, womit 2.214 Entitäten verblieben. Innerhalb dieser Menge stellen die Entitäten vom Typ „Lokation“ mit über 45 % den deutlich überwiegenden Anteil dar, Vorkommen vom Typ „Person“ machen nur 32 % aus, auf die Kategorie „Organisation“ entfallen 22 %. Anders ausgedrückt enthält jeder Nachrichtenartikel durchschnittlich fast zehn

11 http://www.breakingnews.com
12 https://news.google.com
13 http://newsblaster.cs.columbia.edu
14 http://newsstand.umiacs.umd.edu/web/
15 http://emm.newsbrief.eu/overview.html
16 http://rss.cnn.com/rss/edition.rss; Datum der durchgeführten Betrachtung: 30.05.2014
17 http://www.alchemyapi.com/api/entity/
18 Alchemy deckt über hundert, sehr fein strukturierte Entitätentypen ab, die manuell auf die drei genannten Kategorien abgebildet wurden. Spezielle Typen, wie „Anatomy“, „Automobile“, „Anniversary“ etc., die nur einen geringen Anteil ausmachen und die in keine der Kategorien passten, wurden verworfen.

unterschiedliche Ortsbezeichnungen. Die hier vorgestellte Nachrichtenplattform wird deshalb für die Suche nutzergenerierter Inhalte maßgeblich von geographischer Semantik Gebrauch machen. Im Verlauf dieser Arbeit wird exploriert, wie geographische Informationen aus Nachrichten extrahiert und wie mit diesen Informationen die Suche verbessert werden kann.

1.2 DEFINITIONEN

Ereignis In den vorherigen Abschnitten wurde mehrfach der Term „Ereignis" erwähnt, ohne diesen bisher näher zu definieren. Die hier verwendete Definition basiert auf jener, die im Bereich Topic Detection and Tracking (TDT) verwendet wird (J. Allan, 2002). Cieri, Graff, Liberman, Martey und Strassel (2000) bezeichnen im TDT-Kontext ein Ereignis als eine spezifische Sache, die zu einer spezifischen Zeit und an einem spezifischen Ort passiert, mitsamt notwendigen Vorbedingungen und Konsequenzen. Im Falle des eingangs erwähnten Flugzeugunglücks wird beispielsweise die missglückte Landung und die daraus resultierende Bruchlandung, die Rettungseinsätze und die Auswirkungen auf die Abfertigung am Flughafen demselben Ereignis zugeordnet.

Diese Arbeit schränkt den Begriff ein, indem gefordert wird, dass das jeweilige Ereignis eine sichtbare Veränderung am Ort des Geschehens auslöst und dabei ein internationales Interesse hervorruft. Diese Kriterien gelten als erfüllt, wenn die unmittelbar mit dem Ereignis zusammenhängenden Auswirkungen fotografisch dokumentiert und durch Menschen mit dem Ereignis in Zusammenhang gebracht werden können. Dies ist beispielsweise bei Naturkatastrophen wie Erdbeben oder Unfällen, Massenprotesten und -aufständen wie dem arabischen Frühling oder bedeutenden organisierten Veranstaltungen wie der Olympiade der Fall. Nicht als Ereignis betrachtet werden in dieser Arbeit hingegen politische oder diplomatische Ansprachen, Vereinbarungen, Verhandlungen und Interviews, Berichte über polizeiliche Ermittlungen, Entwicklungen auf dem Finanzmarkt oder die Verabschiedung von Gesetzen, da hier das Kriterium einer „sichtbaren Veränderung" in der realen Welt nicht unmittelbar erfüllt ist.

Geographische Entität Sang und Meulder (2003) definieren den Begriff „Named Entity" als Phrase (innerhalb eines Texts), die Namen von Personen, Organisationen oder Lokationen enthält. Als „geographische Entitäten" werden in dieser Arbeit somit Named Entities mit Ortsbezug bezeichnet, wie beispielsweise „Canberra", „France", „Hyde Park" oder „Bellagio". Ortsnamen sind jedoch in den wenigsten Fällen eindeutig, so existieren beispielsweise mehrere Orte mit dem Namen „Hyde Park" auf der Welt. Das Problem der eindeutigen Referenzierung von Entities wird als Disambiguierung bezeichnet und im weiteren Verlauf der Arbeit detailliert betrachtet.

1.3 FOKUS UND ABGRENZUNG

In Abschnitt 1.1 wurde die Vision einer Nachrichtenplattform mit nutzergenerierten Inhalten skizziert. Diese Vision wird innerhalb der Arbeit aus Sicht der Forschungsfelder der Informationsextrakti-

on und des Information Retrievals adressiert. Eine Reihe von weiteren Fragen und Forschungsschwerpunkten entstehen in diesem Zusammenhang auch auf journalistischer, gesellschaftlicher und wirtschaftlicher Ebene, die selbstredend den Rahmen dieser Arbeit sprengen. Auch innerhalb der beiden fokussierten Felder können jedoch nicht alle potentiellen Fragestellungen beantwortet werden, weshalb nachfolgend die getroffenen Einschränkungen erläutert werden.

Fokus Als wesentliche Eigenschaft zur Beschreibung von Nachrichtenereignissen wird in dieser Arbeit der Aspekt „Ort" betrachtet. Wie in der Motivation gezeigt wurde, stellen Entitäten des Typs „Ort" einen signifikanten Bestandteil von Nachrichtenartikeln dar. Ein bedeutender Teil dieser Arbeit ist deshalb der Extraktion von Lokationsdaten aus Textdokumenten und der Beurteilung des geographischen Hauptfokus von Textdokumenten gewidmet. Letzterer bezeichnet den repräsentativsten Ort eines Dokuments. Für beide Aspekte werden unterschiedliche Methoden vorgestellt und miteinander verglichen. Extrahierte Lokationsdaten werden für eine zielgerichtete Suche nach nutzergenerierten Inhalten verwendet.

Für die Suche wird auf offizielle Schnittstellen der betrachteten Plattformen zurückgegriffen, also kein Crawling oder Scraping vorgenommen. Die beschriebenen Methoden bedienen sich durchweg des Text Minings. Dies bedeutet, dass visuelle Eigenschaften von Bildern oder Videos in dieser Arbeit keine Betrachtung finden, sondern ausschließlich Texte, Bild-/Videotitel, Beschreibungen oder Tags berücksichtigt werden.

Abgrenzung Während das vorliegende Manuskript in deutscher Sprache verfasst ist, wurden die hier vorgestellten Methoden und Algorithmen für die englische Sprache entwickelt, optimiert und dementsprechend evaluiert. Ein weiterer – im vorliegenden Umfeld zweifellos bedeutender – Aspekt ist die Beurteilung der Glaubwürdigkeit berücksichtigter Quellen. Konzepte für eine Glaubwürdigkeitsbeurteilung betrachteter Inhalte werden im Rahmen dieser Arbeit nicht detailliert. Zu guter Letzt wird der Aspekt der Benutzerschnittstelle von den Betrachtungen in dieser Arbeit ausgeschlossen. Für die Akzeptanz des beschriebenen Gesamtsystems ist eine intuitive Schnittstelle essentiell, diese Betrachtung liegt jedoch ebenfalls außerhalb des Fokus.

1.4 FORSCHUNGSFRAGEN UND HYPOTHESEN

Ziel der Arbeit ist die Beantwortung der nachfolgenden vier Forschungsfragen. Aus jeder Forschungsfrage wird jeweils eine Hypothese abgeleitet, die im Verlauf der Arbeit verifiziert wird.

Welche Methoden eignen sich, um aus unstrukturierten Texten geographische Entitäten zu extrahieren?
Nachrichtentexte enthalten eine Vielzahl geographischer Entitäten, die Informationen über den oder die Orte des Geschehens liefern. Geographische Entitäten sind mithin wichtige Attribute zur Beschreibung und Einordnung von Nachrichtenmeldungen.

Erste Hypothese: Mittels Mechanismen des Machine Learnings kann ein Extraktionsmechanismus geschaffen werden, der geographische Entitäten aus englischen Texten auf Koordinaten abbildet. Dieser Mechanismus erreicht im Vergleich zu State-of-the-Art-Systemen eine bessere Extraktionsqualität bezüglich üblicher Standardmaße[19] im Forschungsfeld der Informationsextraktion.

Welche Methoden eignen sich, um den geographischen Fokus unstrukturierter Texte zu bestimmen?
In Nachrichtentexten finden sich in der Regel mehrere geographische Entitäten, beispielsweise „Ukraine", „Moscow", „Russia", „Kiew", „Europe" „Majdan Nesaleschnosti"[20]. Der Ort des Geschehens des jeweiligen Ereignisses, also die Frage nach dem „Where" kann in der Regel durch eine dieser Entitäten (im erwähnten Beispiel „Majdan Nesaleschnosti") wiedergegeben werden.
Zweite Hypothese: Die aus einem Text extrahierten geographischen Entitäten können mittels Machine Learning und einer kleinen Menge von Trainingsdaten so gerankt werden, dass der Fokus präziser bestimmt werden kann als mit heuristischen Regeln. Existierende Verfahren der Textklassifikation zur Fokusbestimmung erzielen nur bei beträchtlichen Mengen von Trainingsdaten, die mehrere Größenordnungen über den notwendigen Trainingsmengen für das vorgestellte Machine-Learning-basierte Verfahren liegen, vergleichbare Resultate.

Wie können große Mengen ereignisrelevanter Informationen in Quellen für nutzergenerierte Inhalte gefunden werden?
Durch die wachsende Verbreitung von REST-Paradigmen (Representational State Transfer; Fielding, 2000) bietet mittlerweile quasi jede Plattform für nutzergenerierte Inhalte offiziell publizierte Schnittstellen an, die eine Suche nach Inhalten unter Verwendung unterschiedlicher Kriterien erlauben. So können Entwickler beispielsweise über die API[21] der Plattform Flickr neben einer reinen Textsuche auf Facetten wie zeitliche Intervalle oder geographische Koordinaten und Regionen zurückgreifen. Bei einem gegebenen Nachrichtenereignis müssen jedoch passende Suchanfragen verwendet werden, um einen hohen Recall bei akzeptabler Precision zu erzielen.
Dritte Hypothese: Bei der Verwendung von Anfragen mit lokationsspezifischen Bestandteilen (also Ortsnamen und geographischen Koordinatenpaaren, welche unter Verwendung der vorangehend beschriebenen Mechanismen extrahiert wurden) kann eine hohe Anzahl von Treffern gefunden werden und eine bessere Precision als bei reiner Suche mit Schlüsselwörtern oder Wortgruppen erzielt werden.

Wie kann für die gefundenen Informationen eine hohe Precision im Hinblick auf die Relevanz für das Ereignis erreicht werden?
Bei der vorangehend beschriebenen Suche nach nutzergenerierten Inhalten lag das Ziel darin, einen

19 Die genaue Definition der verwendeten Evaluierungsmaße „Precision" und „Recall" folgt im Rahmen der Grundlagen in Abschnitt 2.4.
20 Der „Platz der Unabhängigkeit" in Kiew, der in den Jahren 2013 und 2014 das Zentrum der Euromaidan-Proteste war.
21 `http://www.flickr.com/services/api/`

hohen Recall an Ergebnissen zu erzielen. Die Precision, das heißt die Relevanz der einzelnen Resultate, war hier nur zweitrangig. Aufgrund dessen ist davon auszugehen, dass die Ergebnismengen insgesamt viele irrelevante Bestandteile enthalten, sodass eine nachträgliche Filterung notwendig ist.
Vierte Hypothese: Durch die Anwendung von Klassifikationsmechanismen, die mittels Machine Learning und Trainingsdaten trainiert wurden, kann der Anteil irrelevanter Inhalte für ein gegebenes Nachrichtenereignis stark reduziert werden, ohne dabei irrtümlich viele tatsächlich relevante Treffer auszufiltern.

1.5 ARCHITEKTUR DES GESAMTSYSTEMS

Wie bereits in den Forschungsfragen angedeutet, nutzt das hier vorgestellte Gesamtsystem maßgeblich Mechanismen der geographischen Informationsextraktion, um damit gezielt nach nutzergenerierten Inhalten für Nachrichtenereignisse zu suchen. Das dabei beschriebene Gesamtsystem wird nachfolgend „NewsSeecr“ genannt.

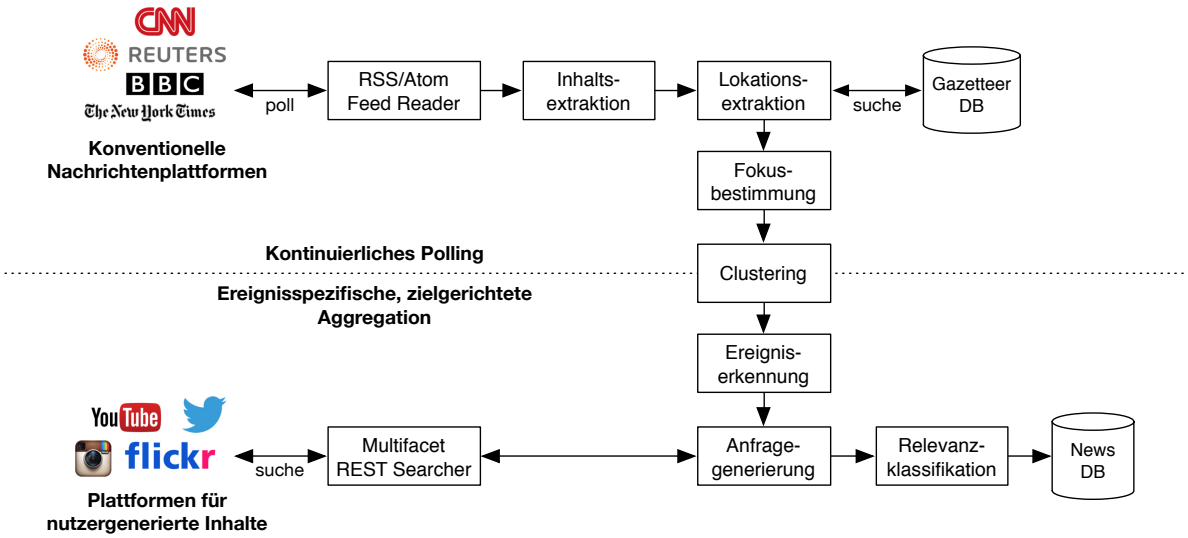


Abbildung 1.2: Architektur des NewsSeecr-Systems mit beteiligten Komponenten und Verarbeitungsablauf

Abbildung 1.2 zeigt die Architektur des Gesamtsystems mit relevanten Komponenten und dem zugehörigen Verarbeitungsfluss. Die durch das System verarbeiteten externen Datenquellen teilen sich in **Konventionelle Nachrichtenplattformen** und **Plattformen für nutzergenerierte Inhalte** auf. Erstere werden mittels Web-Feeds abgefragt, für letztere findet eine zielgerichtete Suche mit ereignisspezifischen Anfragen statt. Für die Abfrage der Feed-Quellen (Komponente **RSS/Atom Feed Reader** und **Inhaltsextraktion**) kommen vorhandene Konzepte zum Einsatz, die unter Mitarbeit des Autors beispielsweise bereits in Reichert et al. (2011) Verwendung fanden. Sie sind Bestandteil der Softwarebibliothek Palladian (Urbansky, Muthmann, Katz und Reichert, 2012) verfügbar und werden

deshalb im Rahmen dieser Arbeit nicht näher thematisiert. Die im Verarbeitungsablauf nächsten Schritte der **Lokationsextraktion** und **Fokusbestimmung** bilden den Schwerpunkt der vorliegenden Arbeit. Unter Verwendung der in den Kapiteln 3 und 4 beschriebenen Konzepte werden aus den aggregierten Nachrichtentexten die in Abschnitt 1.2 definierten geographischen Entitäten extrahiert, die von nachfolgenden Verarbeitungsschritten genutzt werden. Für diese Verarbeitungsschritte wird auf geographisches Wissen zurückgegriffen, welches dem System in Form der **Gazetteer-Datenbank** bereitgestellt wird.

Aufgabe des **Clusterings** im nächsten Schritt ist, die aggregierten Nachrichtendokumente im Hinblick auf Ereignisse zu gruppieren. Berichten beispielsweise die Quellen CNN, BBC und Reuters über das Ereignis „Erdbeben auf Bohol", ist das Ziel, die entsprechenden Berichte zu einem Ereignis-Cluster zusammenzufügen. Die nachgelagerte **Ereigniserkennung** filtert jene Inhalte und Cluster, die nicht der in Abschnitt 1.2 spezifizierten Definition eines Ereignisses entsprechen, um unnötige Anfragen, für die keine der erwünschten Resultate zu erwarten sind, in den späteren Verarbeitungsschritten zu vermeiden. Die Komponente **Anfragegenerierung** erzeugt für ausgewählte Cluster zielgerichtete Suchanfragen, mit denen eine fokussierte Datenaggregation von Quellen mit nutzergenerierten Inhalten, wie YouTube, Twitter, Instagram oder Flickr vorgenommen wird. Die Komponente **Multifacet REST Searcher** bildet dazu die Anfragen auf die spezifischen REST-APIs der betrachteten Quellen ab und nimmt ein Parsing der Ergebnisdaten auf das verwendete interne Datenformat vor. Die entsprechenden Komponenten für die multifacettierte Suche wurden, mit durchaus nicht geringem Aufwand, im Zuge dieser Arbeit implementiert und sind innerhalb von Palladian verfügbar. Da die damit verbundenen Herausforderungen jedoch primär im implementierungsnahen Bereich liegen, werden diese im Rahmen der Arbeit nicht eingehend detailliert.

Die abschließende **Relevanzklassifikation** adressiert das im Zusammenhang mit der vierten Forschungsfrage erläuterte Problem, dass aggregierte Inhalte viele irrelevante Beiträge enthalten, indem diese klassifiziert und gefiltert werden. Die Komponenten Clustering, Ereigniserkennung, Anfragegenerierung und Relevanzklassifikation werden in Kapitel 5 thematisiert. Die **News-Datenbank** enthält die Resultate des hier beschriebenen Arbeitsablaufs. Sie bildet den Anknüpfpunkt für Systeme, die auf den vorgestellten Konzepten aufbauen und die aggregierten Daten, beispielsweise in Form einer Webanwendung oder Smartphone-/Tablet-App, dem Anwender präsentieren.

1.6 ZUSAMMENFASSUNG UND AUFBAU

In Kapitel 2 werden relevante Grundlagen erläutert. Dies umfasst geographische Maße und Distanzen, verwendete Methoden des Machine Learnings und Maße zu Evaluierung im Information Retrieval. Leser, die bereits über entsprechendes Hintergrundwissen verfügen, können diesen Abschnitt überspringen. Im weiteren Verlauf der Arbeit werden jeweils Verweise zurück auf die Grundlagen gegeben.

Der weitere Aufbau folgt der Struktur der beschriebenen Forschungsfragen und Thesen. In Kapitel 3 wird das Problem der Lokationsextraktion aus unstrukturierten Texten beschrieben, welches durch die erste Forschungsfrage motiviert wurde. Nach einem Überblick über State-of-the-Art-Systeme und Datensets für Evaluierungszwecke werden zwei im Rahmen der Arbeit konzipierte Verfahren zur Lokationsextraktion vorgestellt. Abschließend erfolgt ein ausführlicher Vergleich mit State-of-the-Art-Systemen.

Kapitel 4 knüpft an die Thematik des vorangegangenen Kapitels an und widmet sich der geographischen Fokusbestimmung von Nachrichten. Hier wird Bezug auf die zweite Forschungsfrage genommen. Das Kapitel stellt eine Reihe unterschiedlicher Strategien für die Fokusbestimmung vor. Unter anderem eine Variante, die mittels Machine Learning unterschiedliche Heuristiken miteinander kombiniert. Des Weiteren wird eine neue Methode präsentiert, die State-of-the-Art-Strategien, welche zur Fokusbestimmung ein Raster verwenden, deutlich verbessert. Die insgesamt zehn vorgestellten Strategien werden ausführlich unter Verwendung von Datensets unterschiedlichen Umfangs evaluiert und verglichen.

Im letzten Kapitel 5 werden die dritte und vierte Forschungsfrage adressiert. Es wird beschrieben, wie mittels Clustering relevante Nachrichtenereignisse erkannt werden können. Anschließend werden im Hinblick auf die dritte Forschungsfrage Strategien für die zielgerichtete Suche nutzergenerierter Inhalte für gegebene Ereignisse vorgestellt. Diese Strategien legen die Priorität auf einen hohen Recall. Um die Precision der Ergebnisse zu verbessern, wird im Anschluss die vierte Forschungsfrage thematisiert.

Schlussendlich greift Kapitel 6 die Forschungsfragen dieser Arbeit auf und liefert eine Zusammenfassung der gewonnenen Erkenntnisse.

2 GRUNDLAGEN

Dieses Kapitel gibt einen kurzen Überblick über wichtige Konzepte, die in den nachfolgenden Kapiteln zum Einsatz kommen. Dies umfasst zunächst die Repräsentation von Orten anhand geographischer Koordinaten und daraus hervorgehende Berechnungsmethoden, die vor allem für Kapitel 3 und 4 relevant sind. In Abschnitt 2.2 werden Repräsentationen für Textdokumente vorgestellt, die aus dem „Bag-of-Words"-Modell hervorgehen und an verschiedenen Stellen in dieser Arbeit genutzt werden. Klassifikationsprobleme im Rahmen dieser Arbeit werden mittels Machine Learning adressiert. Die hierzu verwendeten Decision Trees und Random Forests werden in Abschnitt 2.3 eingeführt. Entsprechend der Gepflogenheiten im Information Retrieval und der Informationsextraktion erfolgt eine Bewertung der hier vorgestellten Methoden anhand etablierter Evaluierungsmaße. Die hier verwendeten Maße werden grundlegend in Abschnitt 2.4 eingeführt und später im Verlauf der Arbeit genauer auf die individuellen Problemstellungen adaptiert.

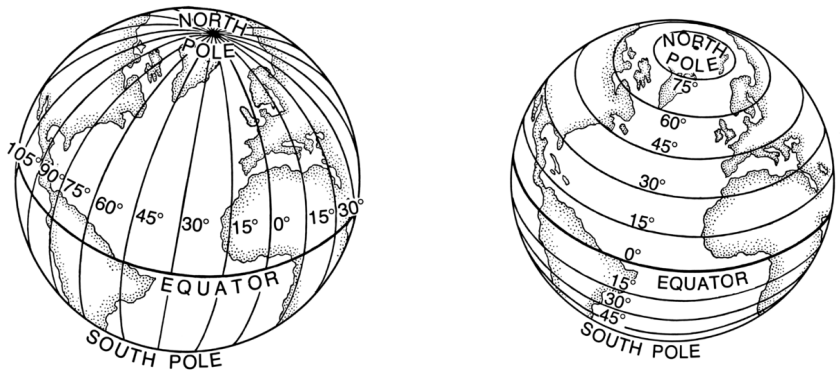
Leser ohne Vorkenntnisse, die direkt in die Thematik einsteigen wollen, können dieses Kapitel überspringen und bei Unklarheiten die vorhandenen Verweise auf die entsprechenden Abschnitte nutzen. Für jene Leser hingegen, die noch tiefer in die hier besprochene Thematik einsteigen möchten, werden zahlreiche Referenzen auf relevante Grundlagenliteratur gegeben.

2.1 GEOGRAPHISCHE GRUNDLAGEN

Nachfolgend wird die in dieser Arbeit verwendete Repräsentation für Orte anhand von Längen- und Breitenkoordinaten, die Ermittlung von Entfernungen zwischen zwei Orten, sowie die Berechnung von Mittelpunkten erläutert.

2.1.1 KOORDINATEN

Punkte auf der Erdoberfläche können mittels geographischer Koordinaten, bestehend aus Breitengrad (Latitude) und Längengrad (Longitude) angegeben werden. Dazu wird der Erdball in 180 Breiten- und 360 Längengrade geteilt (siehe Abbildung 2.1). Per Definition entspricht die geographische Breite von 0° dabei dem Äquator. Die Länge von 0°, der sogenannte Nullmeridian, verläuft durch die Königliche Sternwarte in Greenwich bei London.



(a) Längengrade (b) Breitengrade

Abbildung 2.1: Darstellung der Längen- und Breitengrade der Erde (Pearson Scott Foresman, 2007a, 2007b)

Werden Orte anhand ihrer Koordinaten beschrieben, so wird zuerst der Breiten-, dann der Längengrad angegeben. Der Breitengrad wird dabei entweder mit „Nord“ oder „Süd“ qualifiziert bzw. als positiver (nördliche Breite) oder negativer Wert (südliche Breite) im Bereich von 0° bis 90° angegeben. Entsprechend wird der Längengrad entweder mit dem Zusatz „West“ oder „Ost“ bzw. als negativer oder positiver Wert (entsprechend westlicher oder östlicher Länge) von 0° bis 180° definiert. Verwendet werden auch die Symbole λ für die Länge und ϕ für die Breite.

2.1.2 DISTANZEN

Die Distanzbestimmung zwischen zwei per Koordinaten angegebenen Orten auf der Weltkugel kann über den Großkreis erfolgen, bei dem ein Teilstück, eine sogenannte „Orthodrome“, die kürzeste Strecke zwischen diesen Punkten darstellt. Ein Großkreis ist ein größtmöglicher Kreis auf einer Kugeloberfläche, dessen Mittelpunkt mit dem Kugelmittelpunkt zusammenfällt und der diese somit in zwei gleich große Halbkugeln teilt. Die in Formel 2.1 abgebildete Haversine-Funktion (Sinnott, 1984) berechnet diese Distanz für die zwei Punkte P_1 und P_2 mit den Längen λ_1 und λ_2 und den Breiten ϕ_1 und ϕ_2 und einem angenommenen Erdradius r. Für r wird typischerweise ein Wert von 6.371 Kilometern verwendet. Da die Erde keine perfekte Kugel ist, ermittelt die Formel nur eine Näherung für die Distanz. Die Genaugkeit der Distanzberechnung reicht jedoch für Anwendungen in dieser Arbeit völlig aus[22]. Für die programmatische Berechnung müssen die Koordinaten zusätzlich vorher vom Grad- ins Bogenmaß umgerechnet werden.

$$\text{distance}(P_1, P_2) = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_1 - \phi_2}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \tag{2.1}$$

22 Eine genauere Berechnung erlaubt die Formel von Vincenty (1975).

2.1.3 GEOGRAPHISCHER MITTELPUNKT

Die Berechnung des geographischen Mittelpunkts mehrerer Orte kann über den Schwerpunkt erfolgen. Die Ermittlung einfacher Durchschnittswerte aus Länge und Breite ist hingegen, aufgrund der „Unterbrechung" der Längengrade gegenüber dem Nullmeridian, nicht zielführend. Intutitiv kann diese Vorgehensweise so beschrieben werden: An jedem bei der Schwerpunktbestimmung zu berücksichtigenden Ort wird ein gleich schweres Gewicht auf einem Globus befestigt. Anschließend kann dieser Globus so lange frei rotieren, bis der schwerste Punkt nach unten zeigt. Dieser Punkt ist der geographische Mittelpunkt (GeoMidpoint.com, 2007). Zur Berechnung werden die Koordinaten aus Längen- und Breitengraden zuerst in dreidimensionale kartesische Koordinaten umgewandelt (Formel 2.2).

$$(x,y,z) = (\cos(\phi)\cos(\lambda), \cos(\phi)\sin(\lambda), \sin(\phi)) \tag{2.2}$$

Anschließend wird der Durchschnitt aus den x-, y- und z-Werten berechnet und die ermittelte Durchschnittskoordinate zurück in Längen- und Breitengrade transformiert (Formel 2.3).

$$(\phi, \lambda) = (\text{atan2}(y, x), \text{atan2}(z, \sqrt{x^2 + y^2})) \tag{2.3}$$

2.1.4 GEOGRAPHISCHER MEDIAN

Anders als der Schwerpunkt in Abschnitt 2.1.3 bezeichnet der geographische Median jenen Punkt, der die Summe der Distanzen zu sämtlichen anderen betrachteten Orten minimiert. Er ist ein Spezialfall des geometrischen Mittelpunkts[23], dessen Bestimmung als Fermat–Weber-Problem (Weber, 1909) bezeichnet wird. Im Gegensatz zur geographischen Mittelpunktbestimmung existiert hierfür keine einfache Berechnungsvorschrift, vielmehr muss die Berechnung iterativ erfolgen. Für die Ermittlung des geometrischen Medians wird der von GeoMidpoint.com (2007) beschriebene Algorithmus verwendet, der zunächst jeweils für den Schwerpunkt (Abschnitt 2.1.3) und sämtliche betrachteten Koordinaten die minimale summierte Distanz berechnet und davon ausgehend schrittweise für jeweils acht umgebende Punkte mit einer festgelegten Distanz prüft, ob diese die summierten Distanzen reduzieren. Abbildung 2.2 zeigt zur Verdeutlichung den Schwerpunkt und den geometrischen Median für eine Menge von drei Koordinaten.

2.2 DOKUMENTMODELLE

Um ein Textdokument anhand eines Vektors zu repräsentieren, muss dieses zunächst in Merkmale überführt werden, die anschließend gewichtet werden können. Dazu kann eine Tokenisierung vorgenommen werden, die den Text in Wörter (Tokens) segmentiert. Anschließend können Stoppwörter entfernt werden, die für sich keine Bedeutung tragen. Mittels Stemming lassen sich Wortvarianten

23 In der Literatur wird der geometrische Median auch als „Zentrum minimaler Distanz" oder „Medianzentrum" bzw. „center of minimum distance" oder „median center" bezeichnet.

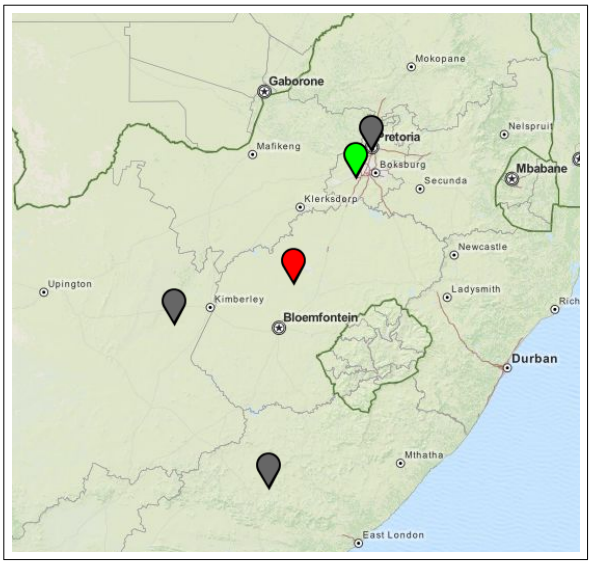


Abbildung 2.2: Geographischer Mittelpunt (rot) und geographischer Median (grün) für die gegebenen grauen Koordinaten (Kartenmaterial von MapQuest, 2013)

des selben Wortstamms auf eine gemeinsame Grundform reduzieren (Hooper und Paice, 2005; Porter, 2001). Um Merkmale mit mehr Kontext zu erhalten, können mehrere Wörter zu Wort-n-Grammen kombiniert werden. Dabei werden unter Verwendung eines Schiebefensterprinzips jeweils Sequenzen aus $n \geq 2$ Wörtern zu einer Einheit kombiniert. Für den Satz „San Francisco is nice“ entstehen beispielsweise die 2-Gramme „San Francisco“, „Francisco is“ und „is nice“. Zeichen-n-Gramme dagegen sind Sequenzen mit jeweils fester Zeichenlänge. Für das genannte Beispiel entstehen die Zeichen-5-Gramme „San F“, „an Fr“, „n Fra“ etc.

Die einfachste Form zur Merkmalsgewichtung ist das boolesche bzw. „Set-of-Words“-Modell, bei dem binäre Vektoren die Präsenz von Termen im Dokument anzeigen. Ein „Bag-of-Words“-Modell dagegen enthält die Vorkommenshäufigkeit der einzelnen Terme (Termfrequenz, nachfolgend TF) im Dokument und geht davon aus, dass häufig vorkommende Terme eine hohe Relevanz besitzen. Um Unterschiede zwischen unterschiedlich langen Dokumenten auszugleichen, ist es üblich, die Termfrequenzen der Terme t mit der maximalen Termfrequenz im Dokument d zu normalisieren (siehe Formel 2.4).

$$\mathrm{tf}(t,d) = \frac{\mathrm{count}(t,d)}{\max\{\mathrm{count}(u,d) : u \in d\}} \tag{2.4}$$

Einen weiteren Indikator für die Wichtigkeit eines Terms liefert die Häufigkeit seines Vorkommens in anderen betrachteten Dokumenten. Terme, die in vielen Dokumenten auftreten, sind weniger charakteristisch als seltener vorkommende Terme. Die in Formel 2.5 abgebildete inverse Dokumentfrequenz (nachfolgend IDF) gibt den zur Dämpfung logarithmierten, inversen Anteil von

Dokumenten innerhalb eines Korpus D an, die einen Term t enthalten.

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{2.5}$$

Die Termfrequenz und die inverse Dokumentfrequenz (nachfolgend TF-IDF) können als Produkt miteinander kombiniert werden (Jones, 1972), womit sich eine Relevanzgewichtung in Abhängigkeit von der Vorkommenshäufigkeit im Dokument d und der Häufigkeit im Korpus D ergibt (Formel 2.6).

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d)\ \mathrm{idf}(t,D) \tag{2.6}$$

2.3 MACHINE LEARNING ZUR KLASSIFIKATION

Mit Machine Learning wird die Fähigkeit eines künstlichen Systems bezeichnet, Wissen aus vorhandenen Daten zu generieren. Unterschieden wird in die Kategorien „überwachtes Lernen“ (supervised learning) und „unüberwachtes Lernen“ (unsupervised learning). In dieser Arbeit sind vor allem Klassifikationsprobleme von Relevanz bei denen eine Zuordnung von Objekten zu Kategorien unter Zuhilfenahme von Kriterien (Attributen oder Features) vorgenommen wird. Klassifikationsprobleme können mittels überwachtem Lernen adressiert werden. Dazu sind explizite Trainingsbeispiele notwendig, aus denen mittels Lernalgorithmen eine Hypothese, auch als Modell bezeichnet, kreiert wird. Mit dieser sollen möglichst korrekte Klassifikationsentscheidungen für die Trainingsdaten und – insbesondere – neue, unbekannte Daten (Testdaten) getroffen werden können.

Klassifikationsalgorithmen können danach kategorisiert werden, ob sie auf die Trainingsdaten „überanpassen“ oder „unteranpassen“. Überanpassung (overfitting oder „high variance“) bezeichnet die Eigenschaft, dass eine Hypothese zu stark an die zugrunde liegenden Daten angepasst ist. Dies bedeutet, dass gute Klassifikationsresultate auf den gelernten Daten erzielt werden, die Klassifikation jedoch schlecht generalisiert. Somit werden auf ungesehenen Testdaten deutlich schlechtere Klassifikationsergebnisse erzielt. Auf der anderen Seite steht die Unteranpassung (underfitting oder „high bias“). Hierbei gibt die Hypothese nicht ausreichend die Eigenschaften der zugrundeliegenden Trainingsdaten wieder.

Die nachfolgende Tabelle 2.1 zeigt das „Play Tennis“-Datenset (Quinlan, 1986), ein verbreitetes Beispieldatenset im Machine-Learning-Bereich, bei dem auf Basis von Wetterbedingungen die Entscheidung getroffen werden soll, ob eine Person Tennis spielen will. Das Datenset besteht aus vier Features (Outlook, Temperature, Humidity, Windy) und einer binären Klassenzugehörigkeit (Class), die die negative (N) bzw. positive (P) Entscheidung repräsentiert.

	#	Outlook	Temperature	Humidity	Windy	Class
Training	1	sunny	hot	high	false	N
	2	sunny	hot	high	true	N
	3	overcast	hot	high	false	P
	4	rain	mild	high	false	P
	5	rain	cool	normal	false	P
	6	rain	cool	normal	true	N
	7	overcast	cool	normal	true	P
Test	8	sunny	mild	high	false	N
	9	sunny	cool	normal	false	P
	10	rain	mild	normal	false	N
	11	sunny	mild	normal	true	P
	12	overcast	mild	high	true	P
	13	overcast	hot	normal	false	P
	14	rain	mild	high	true	N

Tabelle 2.1: „Play Tennis“-Dataset (Quinlan, 1986)

Bei Betrachtung der Trainingsdaten in Tabelle 2.1 könnte beispielsweise die nachfolgende Hypothese zur Klassifikation für Klasse „P“ abgeleitet werden:

$$
\begin{aligned}
&(\mathit{Outlook} = \mathit{overcast} \wedge \mathit{Temperature} = \mathit{hot} \wedge \mathit{Humidity} = \mathit{high} \wedge \neg \mathit{Windy}) \\
\vee\ &(\mathit{Outlook} = \mathit{rain} \wedge \mathit{Temperature} = \mathit{mild} \wedge \mathit{Humidity} = \mathit{high} \wedge \mathit{Windy}) \\
\vee\ &(\mathit{Outlook} = \mathit{rain} \wedge \mathit{Temperature} = \mathit{cool} \wedge \mathit{Humidity} = \mathit{normal} \wedge \neg \mathit{Windy}) \\
\vee\ &(\mathit{Outlook} = \mathit{overcast} \wedge \mathit{Temperature} = \mathit{cool} \wedge \mathit{Humidity} = \mathit{normal} \wedge \mathit{Windy})
\end{aligned}
$$

Diese Hypothese zeigt eine Überanpassung an die betrachteten Daten. Während die Klasse für die zum „Training“ benutzten Einträge 1 bis 7 mit den gezeigten Regeln durchweg korrekt bestimmt wird, werden bei den ungesehenen Daten 8 bis 14 die tatsächlich mit „P“ bezeichneten Einträge irrtümlich als „N“ klassifiziert.

Im Gegensatz dazu ist die nachfolgende Hypothese bei Betrachtung der Trainingsbeispiele 1 bis 7 unterangepasst. Hier werden zwei der positiven Einträge (# 3 und # 7) fälschlicherweise als „N“ klassifiziert.

$$\mathit{Outlook} = \mathit{rain} \wedge \neg \mathit{Windy}$$

Die Modelle unterschiedlicher Lernverfahren können zwei Kategorien zugeordnet werden: Generativen und diskriminativen. Vereinfacht gesagt modellieren erstere zur Klassifikation die Eigenschaften der zugrunde liegenden Daten, wohingegen letztere lediglich Kriterien zur Unterscheidung modellieren. Ein Beispiel[24] verdeutlicht dies: Um die Sprache eines Texts zu klassifizieren, können entweder sämtliche vorkommenden Sprachen gelernt werden, um mit diesem Wissen die Sprache zu

24 http://stackoverflow.com/questions/879432/what-is-the-difference-between-a-generative-and-discriminative-algorithm#answer-7762497

bestimmen (generativ) oder es werden lediglich die charakteristischen Unterschiede in den linguistischen Modellen gelernt (diskriminativ). Ng und Jordan (2002) stellen fest, dass mit diskriminativen Verfahren in der Regel bessere Klassifikationsergebnisse erreicht werden – entgegen der womöglich naheliegenden Intuition im Hinblick auf die Mächtigkeit generativer Modelle.

Beispiele für Klassifikatoren mit generativen Modellen sind Naïve-Bayes-Klassifikatoren oder Hidden-Markov-Modelle (HMM). Diskriminative Klassifikatoren sind die logistische Regression, Support Vector Machines (SVM), Conditional Random Fields (CRF), künstliche neuronale Netze oder die nachfolgend vorgestellten, in dieser Arbeit verwendeten Decision Trees. Einen ausführlichen Überblick über unterschiedliche Klassifikationsverfahren geben Mitchell (1997) und Manning, Raghavan und Schütze (2009).

2.3.1 NAÏVE BAYES

Naïve-Bayes-Klassfikatoren machen sich das Bayes-Theorem (Bayes, 1763) aus Formel 2.7 zunutze. Mit dem Theorem kann die bedingte Wahrscheinlichkeit $P(A \mid B)$ ermittelt werden, die angibt dass ein Ereignis A unter der Bedingung von B eintritt. Sie lässt sich durch die A-priori-Wahrscheinlichkeit $P(A)$ für das Ereignis A, die bedingte Wahrscheinlichkeit $P(B \mid A)$, dass bei gegebenem Ereignis A das Ereignis B auftritt und die A-priori-Wahrscheinlichkeit $P(B)$ für Ereignis B berechnen.

$$P(A \mid B) = \frac{P(A) \cdot P(B \mid A)}{P(B)} \tag{2.7}$$

Die Bezeichnung „naïv“ rührt daher, dass für die Klassifikation mit mehreren Eigenschaften die Annahme getroffen wird, dass diese voneinander unabhängig sind. Die Grundidee zur Bestimmung der Wahrscheinlichkeit für eine Klasse c bei gegebenen Features F ist in Formel 2.8 dargestellt.

$$P(c \mid F) = P(c) \prod_{f \in F} P(f \mid c) \tag{2.8}$$

Sowohl für die Trainings- als auch für die Klassifikationsphase weisen Naïve-Bayes-Klassfikatoren eine lineare Komplexität auf. Zum Training müssen lediglich Wahrscheinlichkeiten berechnet werden. Aus diesem Grund sind Naïve-Bayes-Klassfikatoren bei richtiger Adaption gut für die Textkategorisierung geeignet, wo sie bezüglich Klassifikationsgüte mit deutlich komplexeren Verfahren wie SVMs konkurrieren können. Die in dieser Arbeit verwendeten Maßnahmen zur Adaption von Naïve-Bayes-Klassfikatoren auf Textklassifikationsprobleme beschreiben Rennie, Shih, Teevan und Karger (2003).

2.3.2 DECISION TREES

Decision Trees (zu deutsch „Entscheidungsbäume“) gehören zu den weitverbreitetsten Methoden für Klassifikationsaufgaben und werden auch im Bereich der Informationsextraktion häufig eingesetzt.

Sie teilen komplexe Entscheidungen in einfach verständliche, schrittweise angewendete Regeln auf und bilden Eingabedaten auf diskrete Zielklassen ab. Abbildung 2.3 zeigt einen Decision Tree für das Datenset „Play Tennis“ aus Tabelle 2.1. An jedem Feature (repräsentiert durch die Knoten „Outlook“, „Humidity“ und „Windy“) erfolgt eine Entscheidung auf Basis der in abgerundeten Rechtecken abgebildeten Zweige (zum Beispiel „sunny“, „overcast“ oder „rain“ auf erster Ebene). Zur Klassifikation von Daten wird der Baum schrittweise von oben nach unten durchlaufen, bis ein Blattknoten des Baumes erreicht ist, der eine Klasse repräsentiert („P“ oder „N“ im Kreis).

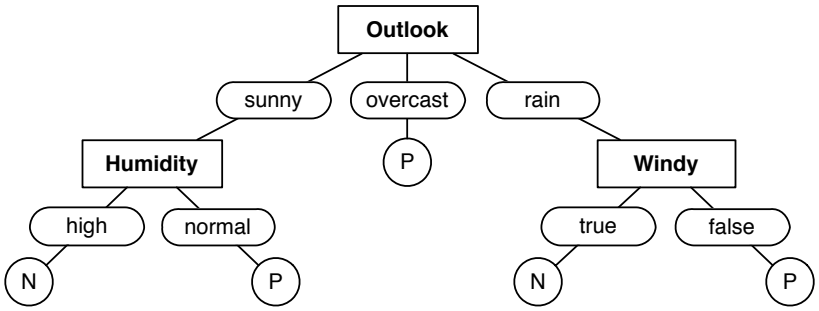


Abbildung 2.3: Decision Tree für das „Play Tennis“-Datenset (siehe Tabelle 2.1, nach Quinlan, 1986) mit den Features „Outlook“, „Humidity“ und „Windy“ und den Klassen „N“ und „P“ als Ergebnis der Klassifikation

Algorithmen zum Lernen von Decision Trees sind der in Breiman, Friedman, Stone und Olshen (1984) beschriebene CART oder der in Quinlan (1986) beschriebene ID3. Der Nachfolger C4.5 (Quinlan, 1993) verbessert ID3, beispielsweise indem auch kontinuierliche Features unterstützt werden (also beispielsweise Gradzahlen für die in Tabelle 2.1 angegebene Temperatur) oder mit fehlenden Attribute umgegangen werden kann. Mittlerweile existiert vom gleichen Autor mit C5.0 eine weiter verbesserte und kommerziell vermarktete Implementierung[25].

Die nachfolgenden Erklärungen zum Decision Tree Learning stützen sich auf den ursprünglichen Algorithmus ID3. Die zentralen und relevanten Aspekte sollen hier wiedergegeben werden. Für nähere Details sei auf Quinlan (1993) und Mitchell (1997) verwiesen.

ID3 baut einen Entscheidungsbaum rekursiv „top-down“ auf. Dazu wird am Wurzelknoten zunächst für sämtliche Trainingsinstanzen bestimmt, mit welchem der vorhandenen Attribute die beste Klassenseparation erzielt wird. Diese Quantifizierung erfolgt mittels des in Formel 2.9 angegebenen Information-Gain-Maß. Information Gain bezeichnet die Entropiereduktion, die bei Separation der Instanzen S unter einem der Attribute aus A stattfindet.

$$\mathrm{IG}(S, A) = \mathrm{H}(S) - \sum_{v \in \mathrm{values}(A)} \frac{|S_v|}{|S|} \mathrm{H}(S_v) \tag{2.9}$$

25 http://www.rulequest.com/products.html

Die Entropie wird wie in Formel 2.10 auf Basis der Klassenzugehörigkeit der Instanzen ermittelt. p_i gibt den Anteil von Instanzen, die zu Klasse i gehören, an.

$$\mathrm{H}(S) = \sum_{i=1}^{c} -p_i \log_2 p_i \tag{2.10}$$

Das Attribut mit dem höchsten Information Gain wird jeweils ausgewählt und mit diesem ein neuer Knoten erzeugt. Anschließend werden sämtliche Instanzen entsprechend ihrer Attribute auf die dem Knoten anschließenden Zweige verteilt und das ausgewählte Attribut aus der verbleibenden Attributmenge entfernt. Die Prozedur wird solange für die jeweils an die Zweige verteilten Instanzen wiederholt, bis sämtliche Attribute berücksichtigt wurden. Als Blattknoten wird jene Klasse mit dem höchsten Anteil gewählt. Befindet sich an einem Zweig eine homogene Menge aus einer Klasse, stoppt die Rekursion für diesen. Der beschriebene Lernalgorithmus ist „greedy", das heißt, es erfolgt kein Backtracking, bei dem einmal erzeugte Knoten revidiert werden.

Die Vorteile von Decision Trees im Vergleich zu anderen Klassifikatoren, insbesondere SVMs, liegen einerseits darin, dass die Daten keinen Vorverarbeitungsschritten wie beispielsweise der Normalisierung unterworfen werden müssen. Ferner erzielen sie auch bei nicht-optimalen Trainingsdaten, beispielsweise mit sehr ungleicher Klassenverteilung oder teilweise fehlerhaften Klassenzuweisungen gute Ergebnisse. Ein weiterer Vorteil besteht darin, dass aufwendige Optimierungsschritte, wie sie beispielsweise bei SVMs in Bezug auf den verwendeten Kernel und dessen Parameter notwendig sind, wegfallen (Chen, 2011). Decision Trees sind für den Menschen gut verständlich und die Relevanz einzelner Attribute kann unmittelbar aus der Position im Baum abgelesen werden.

Als problematisch erweist sich, dass Decision-Tree-Modelle stark overfitten. Diese Problematik kann durch einen nachgelagerten Pruning-Schritt entschärft werden. Hierbei wird die Komplexität eines Decision Trees durch Abschneiden von Zweigen reduziert, um eine bessere Generalisierbarkeit zu erreichen. Für das „Reduced Error Pruning" (Elomaa und Kääriäinen, 2001; Quinlan, 1987) wird ein getrenntes Trainings- und Validierungsset genutzt. Von dem gelernten Baum werden schrittweise einzelne Zweige entfernt und die Anzahl der Fehlklassifikationen mittels Validierungsset überprüft. Dieser Vorgang wird so lange fortgesetzt, wie die Anzahl der Fehlklassifikationen abnimmt.

2.3.3 RANDOM FORESTS

Eine weitere Möglichkeit, ein Overfitting zu vermeiden, stellen Ensemble-basierte Methoden dar, bei denen mehrere Modelle gelernt und kombiniert werden. Einen Überblick geben Opitz und Maclin (1999) und Dietterich (2000). Beim „Bagging" (Bootstrap Aggregation) werden aus einer Trainingsmenge mehrere Teilmengen erzeugt, so genannte „Bootstrap Samples" (Breiman, 1996). Mit jeder Teilmenge kann anschließend ein Decision Tree trainiert werden.

Die von Breiman (2001) beschriebenen Random Forests kombinieren Bagging mit einer Random-Subspace-Methode (Ho, 1998), bei der an jedem Knoten nur eine Teilmenge der tatsächlich vor-

handenen Attribute berücksichtigt werden (bei p Attributen üblicherweise $\sqrt{p}$ oder $\log_2 p$). Ein anschließendes Pruning der Bäume findet hier nicht statt.

Zur Klassifikation werden die Daten mit jedem Decision Tree klassifiziert und mittels Mehrheitsentscheid die wahrscheinlichste Klasse ermittelt. Die Häufigkeiten der vorhergesagten Klassen können genutzt werden, um Aussagen zur Konfidenz der Klassifikation zu treffen (wenn beispielsweise sieben von zehn Bäumen eines Random Forests die Klasse „P“ angeben, entspricht dies einer Konfidenz von 0,7). Die Konfidenzwerte können genutzt werden, um bei binären Klassenzugehörigkeiten eine Polarisierung der Klassifikationsergebnisse zu erreichen. So kann anstatt des üblichen Schwellwerts von 0,5 beispielsweise ein Wert von 0,2 für die Konfidenz für „P“ gewählt werden, um – bei Inkaufnahme von Fehlklassifikationen – mehr Daten als „P“ zu klassifizieren.

2.3.4 FEATURE-SELEKTION

Klassifikationsaufgaben mittels Machine Learning geht ein Feature Engineering voraus, bei dem deskriptive Features zur Klassifikation beschrieben werden. Nicht immer sind alle Features tatsächlich notwendig. Möglicherweise existieren mehrere voneinander abhängige oder irrelevante Features oder sogar solche, die für die Klassifikation kontraproduktiv sind. Mit der Feature-Selektion soll eine Teilmenge der ursprünglichen Feature-Menge bestimmt werden, mit dem annähernd gleich gute oder sogar bessere Klassifikationsergebnisse erzielt werden (siehe Abschnitt 2.4 für mögliche Maße). Durch Feature-Selektion kann dem Problem des Overfittings vorgebeugt und der Klassifikationsaufwand reduziert werden, da weniger Features berechnet werden müssen.

Guyon und Elisseeff (2003) geben einen Überblick verschiedener Verfahren. In dieser Arbeit wird die „Backward Feature Elimination“ angewendet. Diese beginnt mit der vollständigen Menge aus n Features. Es folgen n Iterationen, wobei in jeder Iteration jedes verbleibende Feature einmal entfernt, unter Verwendung eines Trainingssets ein Modell trainiert und mittels Validierungsset getestet wird. Am Ende jeder Iteration wird jenes Feature eliminiert, bei dessen Wegnahme die besten Klassifikationsergebnisse erzielt wurden (im Hinblick auf Accuracy oder F1-Maß, siehe Abschnitt 2.4).Die Teilmenge kann einerseits aus den Top-k Features bestehen, das heißt, es werden die zuletzt eliminierten k Features selektiert. Andererseits können, ebenfalls beginnend beim zuletzt eliminierten Feature, so viele ausgewählt werden, dass eine festgelegte Klassifikationsqualität erreicht wird.

Ähnlich wie eine „Forward Selection“, die jedoch mit einer leeren Feature-Menge startet und schrittweise die besten Features selektiert, ist die Backward Feature Elimination ein Greedy-Algorithmus. Getroffene Eliminationsentscheidungen werden in späteren Iterationen nicht revidiert. Die Forward Selection wählt jedoch unter Umständen ein gutes Feature, für das eine bessere Featurekombination aus zwei für sich alleine schwächeren Features existiert (Guyon und Elisseeff, 2003). Bei beiden Verfahren besteht theoretisch die Gefahr, dass nur lokale Optima bezüglich Featurekombinati-

on bestimmt werden. Im Gegensatz zu einer „Brute Force“ Feature Selection jedoch, bei der $n!$ Kombinationen untersucht werden müssen, wird der Suchaufwand auf $n\,(n+1)\,/\,2$ reduziert.

2.4 EVALUIERUNG IM INFORMATION RETRIEVAL

Nachfolgend werden die verwendeten Evaluierungsmaße vorgestellt, die im Information Retrieval und für Klassifikationsaufgaben genutzt werden. Diese umfassen Precision (Präzision), Recall (Trefferquote), F-Maß sowie Accuracy.

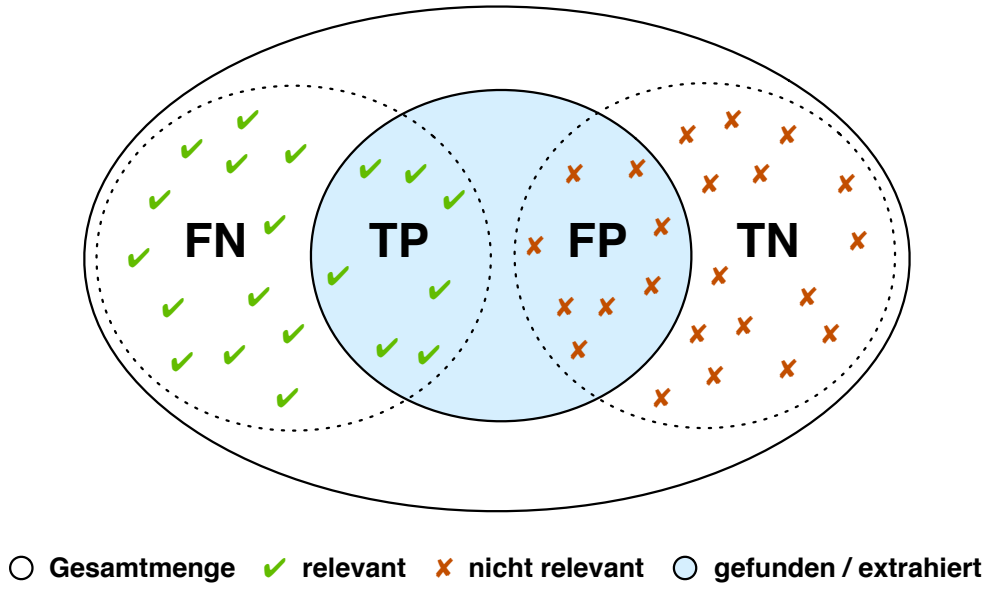


Abbildung 2.4: Dokumentmengen TP (true positive), TN (true negative), FP (false positive), FN (false negative) und deren Bedeutung

Werden Objekte oder Dokumente klassifiziert oder gesucht, so kann eine Evaluierung der Ergebnisse vorgenommen werden, indem die Gesamtmenge in *tatsächlich* relevante und als durch das evaluierte System als relevant *klassifizierte* Elemente gegliedert wird. Durch die Schnittmengen aus relevanten, irrelevanten und gefundenen bzw. extrahierten Elementen ergeben sich die vier in Abbildung 2.4 dargestellten disjunkten Mengen „true positive“ (nachfolgend TP), „true negative“ (TN), „false positive“ (FP) und „false negative“ (FN). TP sind dabei jene Elemente, die tatsächlich relevant sind und durch das System ebenso klassifiziert wurden. TN bezeichnet jene Elemente, die nicht relevant sind und durch das System korrekterweise nicht gefunden bzw. als irrelevant klassifiziert wurden. FP bezeichnet die Menge der Elemente, die durch das System fälschlicherweise als relevant, FN dementsprechend jene, die durch das System irrtümlicherweise als irrelevant klassifiziert oder nicht gefunden wurden.

Die Precision gibt den Anteil korrekter klassifizierter Resultate innerhalb der Ergebnismenge an und berechnet sich wie in Formel 2.11 angegeben.

$$Precision = \frac{|relevant \wedge gefunden|}{|gefunden|} = \frac{|TP|}{|TP| + |FP|} \tag{2.11}$$

Der Recall ermittelt sich wie in Formel 2.12 gezeigt. Er gibt den Anteil korrekt klassifizierter Ergebnisse an den tatsächlich relevanten Elementen an. Die Werte für Precision und Recall liegen im Intervall $[0, 1]$, wobei höhere Werte als besser anzusehen sind. In dieser Arbeit werden die Maße prozentual angegeben.

$$Recall = \frac{|relevant \wedge gefunden|}{|relevant|} = \frac{|TP|}{|TP| + |FN|} \tag{2.12}$$

Precision und Recall sollten immer gemeinsam betrachtet werden. Insbesondere ist eine hohe Precision bei einem minimalem Recall oder umgekehrt in der Regel wertlos. Um eine Betrachtung anhand eines einzigen Werts zu erlauben, wird das F-Maß genutzt, welches das harmonische Mittel aus Precision und Recall darstellt (Rijsbergen, 1979).

$$F\beta = \frac{(1 + \beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \tag{2.13}$$

Mit $\beta \in [0, \infty]$ kann eine Gewichtung von Precision oder Recall vorgenommen werden. Üblicherweise wird das F1-Maß mit $\beta = 1$ verwendet, bei dem Precision und Recall gleich gewichtet werden. Somit vereinfacht sich die Formel wie nachfolgend angegeben (Formel 2.14).

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{2.14}$$

Ein weiteres Evaluierungsmaß, welches hauptsächlich bei Klassifikationsaufgaben eingesetzt wird, ist die Accuracy. Hier wird, wie in Formel 2.15 angegeben, die Anzahl korrekt klassifizierter Elemente in Bezug zur Gesamtmenge gesetzt.

$$Accuracy = \frac{|TP| + |TN|}{|FN| + |TP| + |FP| + |TN|} \tag{2.15}$$

3 LOKATIONSEXTRAKTION

Wie in Abschnitt 1.1 motiviert, haben geographische Entitäten eine große Bedeutung bei der Beschreibung von Nachrichtentexten. Die Betrachtungen zeigten, dass bei den betrachteten Nachrichtentexten jeder im Durchschnitt zehn unterschiedliche Ortsreferenzen enthielt. Aber auch in anderen Domänen sind geographische Entitäten von Bedeutung. Bereits ältere Untersuchungen im Rahmen von Suchmaschinen zeigten, dass 18,6 % der Suchanfragen geographische Terme beinhalten (Sanderson und Kohler, 2004). Neuere Statistiken gehen sogar davon aus, dass 30 bis 40 % der vorgenommenen Suchanfragen bei Google einen lokalen Bezug haben (Parsons, 2012). Das Problem der korrekten Extraktion von Ortsnamen und die Ermittlung zugehöriger Orte ist somit nicht nur im hier betrachteten Nachrichtenumfeld von hoher Relevanz.

Nach einer grundsätzlichen Einordnung und Vorstellung verwandter Arbeiten in Abschnitt 3.2 werden in Abschnitt 3.3 Evaluierungsverfahren für die Lokationsextraktion vorgestellt. In Abschnitt 3.4 folgt ein Überblick über Datensets, welche zur Evaluierung genutzt werden. Die Ergebnisse der dort durchgeführten Recherche haben gezeigt, dass keine frei verfügbaren Datensets von guter Qualität für Evaluierungszwecke existieren, weswegen in Abschnitt 3.4.2 ein während dieser Arbeit erstelles Datenset, welches die Grundlage für spätere Evaluierungen bildet, vorgestellt wird. In Abschnitt 3.5 wird die verwendete Terminologie eingeführt.

Anschließend stellt das Kapitel zwei neue Verfahren für die Lokationsextraktion aus unstrukturierten Texten vor: Einen heuristischen Ansatz in Abschnitt 3.8 und in Abschnitt 3.9 einen Ansatz, welcher auf Machine Learning zurückgreift. Die Abschnitte 3.7 und 3.10 beschreiben beiden Ansätzen gemeinsame Vor- und Nachverarbeitungsschritte. Die Umsetzung der beiden Verfahren und die vorgenommenen Optimierungen werden in Kapitel 3.11 besprochen. Abschließend erfolgt in Kapitel 3.12 ein detaillierter Vergleich der hier präsentierten Methoden mit verschiedenen State-of-the-Art-Mechanismen zur Extraktion von Lokationen.

3.1 PROBLEMSTELLUNG

Die Lokationsextraktion besteht im Wesentlichen aus zwei Schritten wie in Abbildung 3.1 dargestellt. Im Schritt der Named Entity Recognition werden Lokationsvorkommen, sogenannte Toponyme, wie im Beispiel „Canberra“ und „Australia“ aus dem Text extrahiert. Je nach Extraktionsansatz können

diese Kandidaten mit unterschiedlicher Konfidenz als Lokationen begriffen werden. Die Referenz zu tatsächlichen Orten im Sinne von Koordinaten oder eideutigen Identifikatoren wird im nächsten Schritt, der Toponymdisambiguierung, hergestellt.

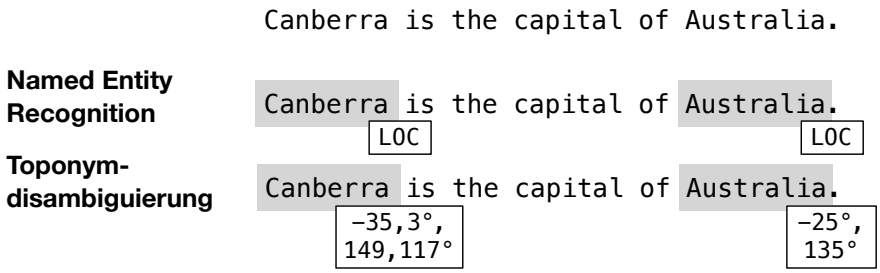


Abbildung 3.1: Schematische Darstellung der Named Entity Recognition und der Toponymdisambiguierung

Named Entity Recognition Named Entity Recognition ist ein etablierter Bereich im Natural Language Processing und der Informationsextraktion. Erste Verfahren wurden zu Beginn der 90er Jahre entwickelt. Vor allem im Umfeld der MUC (Message Understanding Conference) setzte sich die Erkenntnis durch, dass zur systematischen Informationsextraktion aus unstrukturierten Texten eine Differenzierung in spezifische Informationseinheiten wie Personen-, Organisations- und Ortsnamen oder numerische Fragmente wie Zeit-/Datums-, Geld- oder Prozentangaben vorgenommen werden muss. Im Laufe der Zeit wurden verschiedene Strategien zur Named Entity Recognition umgesetzt und evaluiert. Grundsätzlich kann hier eine Evolution von ursprünglich regelbasierten, manuell konfigurierten Systemen zu Ansätzen, welche auf das maschinelle Lernen zurückgreifen, festgestellt werden. Einen grundsätzlichen Überblick liefern Nadeau und Sekine (2007) und Ratinov und Roth (2009).

Der Prozess der Named Entity Recognition wird in folgende Schritte unterteilt: „Delimitation“ bzw. „Recognition“ und „Type Detection“ bzw. „Classification“. In der Delimitation-Phase werden zunächst Kandidaten aus dem Text extrahiert. Anschließend erfolgt die Klassifizierung der extrahierten Kandidaten, wobei diese entweder definierten Kategorien wie „Person“ oder „Lokation“ zugeordnet werden oder als Nicht-Entität klassifiziert und aus der Kandidatenliste entfernt werden.

Im Kontext der Lokationsextraktion identifizieren Amitay, Har'El, Sivan und Soffer (2004) das Problem der „Geo/Non-Geo“-Ambiguitäten. Durch die Named Entity Recognition können einerseits irrtümlich Entitäten anderen Typs als Lokation erkannt werden (z. B. im Satz „Georgia went to Australia“ in dem „Georgia“ korrekterweise als „Person“ klassifiziert werden müsste). Andererseits besteht die Möglichkeit, dass fälschlicherweise Nicht-Entitäten als Lokationen extrahiert werden (z. B. das Land Türkei aus dem Satz „Turkey tastes delicious“).

Toponymdisambiguierung Während durch die Named Entity Recognition ein Vorkommen wie „Canberra“ im Text erkannt und als „Lokation“ oder „Stadt“ klassifiziert werden kann, dient die Toponymdisambiguierung –im Englischen auch als „Grounding“ bezeichnet – dazu, mehrdeutige Lokationsvorkommen auf eine eindeutige Instanz eines Ortes abzubilden. Sollte sich bei der Toponymdisambiguierung zeigen, dass für eine als Ort klassifizierte Entität nur unplausible Abbildungen möglich sind (etwa, weil keine der möglichen Lokationen im Umkreis anderer Lokationen im Text liegt), kann diese auch im Schritt der Toponymdisambiguierung als Nicht-Lokation verworfen werden. Die Referenzierung der Orte erfolgt Anhand einer Datenbasis mit Orten, einem sogenannten Gazetteer.

Mehrdeutigkeiten, von Amitay et al. (2004) als „Geo/Geo“-Ambiguitäten bezeichnet, stellen ein inhärentes Problem bei der Toponymdisambiguierung dar. Smith und Crane (2001) zufolge existieren für 57,1 % der Ortsnamen in Nord- und Zentralamerika mehrere tatsächliche Orte. Bei Betrachtung der gesamten Welt kann davon ausgegangen werden, dass diese Zahl noch weitaus höher liegt. Beispielsweise listet die englischsprachige Wikipedia 26 Lokationen auf, die „Paris“ im Namen haben[26].

3.2 VERWANDTE ARBEITEN

Der nachfolgende Überblick stellt in Abschnitt 3.2.1 verwandte Arbeiten aus dem wissenschaftlichen Umfeld anhand der verwendeten Methoden vor und grenzt die eigene von den existierenden Arbeiten ab. Abschließend erfolgt in den Abschnitten 3.2.2 und 3.2.3 eine Darstellung von Web-APIs und Softwarebibliotheken zur Lokationsextraktion.

3.2.1 METHODEN ZUR EXTRAKTION UND DISAMBIGUIERUNG

Dieser Abschnitt gibt einen kompakten Überblick über verschiedene Heuristiken und Methoden, die zur Lokationsextraktion und -disambiguierung eingesetzt werden.

Räumliche Distanzen Üblicherweise beziehen sich Ortsnennungen in Texten auf gleiche Regionen, weshalb räumliche Distanzen (siehe Abschnitt 2.1.2) Hinweise bei der Disambiguierung geben können. Bei einem Text mit den Toponymen „Cambridge“ und „London“ erscheint zum Beispiel die Annahme plausibel, dass sich beide auf die Orte in England beziehen, die nur 78 km voneinander entfernt liegen, wohingegen beispielsweise die Distanz zwischen London in Ontario und Cambridge in Neuseeland mehrere tausend Kilometer beträgt.

Ausgenutzt wird dies von Smith und Crane (2001), die eine Toponymdisambiguierung für Inhalte einer digitalen Bibliothek historischer Daten durchführen. Nach der Identifikation von Eigennamen folgt eine Filterung unwahrscheinlicher Lokationskandidaten. Die Kandidatenmenge wird reduziert,

26 http://en.wikipedia.org/wiki/Paris_(disambiguation)

indem die Zentroidlokation ermittelt und grobe „Ausreißer“ entfernt werden, die überdurchschnittlich weit vom Zentroid entfernt liegen. In Zong, Wu, Sun, Lim und Goh (2005) werden mehrdeutige Toponymkandidaten disambiguiert, indem jeweils jene Entsprechung mit der geringsten Distanz zu bereits disambiguierten Kandidaten selektiert wird.

Leidner, Sinclair und Webber (2003) treffen für den vorgestellten LSW03-Algorithmus maßgeblich zwei Annahmen. Einerseits erfolgt die Annahme des „Single Sense per Discourse“ (Gale, Church und Yarowsky, 1992), andererseits wird eine „Spatial Minimality“-Heuristik angewendet: Für jede potentielle Disambiguierung, das heißt, jede mögliche Abbildungskombination von Ortsname auf Lokation, wird ein Flächeninhalt berechnet und jene mit der kleinsten Fläche als Ergebnis ausgewählt.

Hierarchie Neben räumlichen Distanzen können auch hierarchische Eigenschaften zur Disambiguierung ausgenutzt werden. Hiermit sind hierarchische Beziehungen zwischen Orten gemeint, die Teil-von-Relationen ausdrücken. Wird innerhalb eines Texts beispielsweise der Bundesstaat „Texas“ genannt, so ist dies ein Hinweis dafür, dass der dort erwähnte Ort „Paris“ nicht die Hauptstadt von Frankreich sondern die Stadt im Osten von Texas bezeichnet.

Li, Srihari, Niu und Li (2002, 2003) verwenden hierarchische Eigenschaften kombiniert mit einem Optimierungsalgorithmus. Mehrdeutige Toponyme werden in eine Graphstruktur mit ungerichteten gewichteten Kanten überführt. Das Kantengewicht determiniert sich durch eine Reihe von Regeln. Der mittels Kruskal-Algorithmus (Kruskal, 1956) ermittelte Maximum Weight Spanning Tree repräsentiert das Ergebnis disambiguierter Toponyme.

Buscaldi und Rosso (2008a) greifen auf die WordNet-Ontologie (Miller, 1995) zurück. Die WordNet-Synsets werden anhand eines „Conceptual Density“-Maßes gerankt, das angibt, wie stark Holonyme im Umfeld des zu disambiguierenden Worts vertreten sind. Holonyme sind Wörter, die in einer Teil-von-Beziehung zu anderen Wörtern stehen. Beispielsweise wäre „North America“ ein Holonym von „USA“. Buscaldi und Rosso (2008a) merken jedoch an, dass WordNet bezüglich der Quantität zur Verfügung stehender Daten nicht mit anderen Gazetteer-Quellen konkurrieren kann. Beispielsweise beinhaltet WordNet nach Angaben der Autoren drei Einträge für den Ort „Springfield“, wohingegen der GNIS-Gazetteer[27] 129 Einträge liefert. Da das zur Evaluierung verwendete Datenset „GeoSemCor“ (siehe Abschnitt 3.4.1) vorrangig häufig vorkommende Orte enthält, ist davon auszugehen, dass die präsentierten Evaluierungsergebnisse deutlich in positive Richtung verfälscht sind. In Buscaldi und Rosso (2008b) wird ein Vergleich zu einem Ansatz gezogen, der die Disambiguierung über Distanzen vom Mittelpunkt (siehe Abschnitt 2.1.3) vornimmt. Mit dem WordNet-basierten Verfahren werden dabei bessere Erebnisse erzielt (85 % F1-Maß). Beide Ansätze sind jedoch einer Baseline, die schlicht die erste Bedeutung eines mehrdeutigen Toponyms auswählt, deutlich unterlegen (94,2 % F1-Maß).

27 http://earth-info.nga.mil/gns/html/index.html

Von da Graça Martins (2008) wird der Hierarchielevel von Lokationen ausgewertet. Im Falle von Mehrdeutigkeiten werden jene Lokationen präferiert, die an oberster Stelle der Hierarchie stehen und somit größere Regionen repräsentieren. Desweiteren werden dicht besiedelte Orte bevorzugt, mutmaßlich also solche mit mehr Kindlokationen, allerdings ist die Arbeit an dieser Stelle nicht explizit.

Einwohnerzahlen Die jeweilige Anzahl von Einwohnern in Orten gleichen Namens liefert einen möglichen Anhaltspunkt dafür, welcher Ort das plausibelste Disambiguierungsresultat darstellt. Diese Information wird üblicherweise durch die Gazetteer-Datenbank zur Verfügung gestellt. Enthält ein Text das Toponym „Dresden", so ist – zumindest unter Ausschluss anderer Indikatoren – die Wahrscheinlichkeit deutlich höher, dass tatsächlich Dresden in Sachsen mit über 500.000 Einwohnern gemeint ist als das Dorf Dresden in Muskington County, Ohio mit lediglich 1.423 Bewohnern[28]. Eine Reihe von Arbeiten, beispielsweise Rauch, Bukatin und Baker (2003), werten daher Einwohnerzahlen während der Disambiguierung aus.

Lokationstypen Verschiedene Typen von Lokationen können bei der Disambiguierung unterschiedlich behandelt beziehungsweise bestimmte Typen präferiert werden. Soll beispielsweise ein Toponym mit dem Namen „China" disambiguiert werden, so liegt die Annahme nahe, dass hier tatsächlich das Land und nicht etwa der Ort China in Kennebec County im Bundesstaat Maine der USA gemeint ist. Leidner et al. (2003) beispielsweise lösen Toponyme grundsätzlich als Land auf, sofern eine solche Entsprechung existiert. Andogah, Bouma, Nerbonne und Koster (2008) treffen die Annahme, dass innerhalb eines Texts vornehmlich gleiche Arten von Lokationen erwähnt werden. Kommen unter den bereits disambiguierten Lokationen beispielsweise hauptsächlich Städte vor, so werden uneindeutige Lokationen ebenfalls auf Kandidaten vom Typ „Stadt" abgebildet.

Lokale Einschränkung Mit dem Vorwissen, dass ein Nachrichtenartikel beispielsweise von einer simbabwischen Quelle stammt, kann eine lokale Einschränkung bei der Disambiguierung vorgenommen werden, indem sämtliche Ortskandidaten, die gemäß Hierarchie nicht innerhalb von Simbabwe liegen, ausgeschlossen werden. Diese Annahme nutzen Pouliquen et al. (2006).

Andogah et al. (2008) bestimmen unter Verwendung eines Dokumentindex und Information-Retrieval-Methoden zuerst einen geographischen Fokus eines Dokuments (siehe Abschnitt 4.1 im nachfolgenden Kapitel), der anhand von Kontinenten, Ländern oder Provinzen repräsentiert wird. Im Anschluss werden sämtliche Lokationskandidaten eliminiert, die außerhalb des Fokus liegen.

Kontexte und Muster Ein Termkontext bezeichnet die Menge von Termen, die innerhalb des Texts im Umfeld des betrachteten Toponyms vorkommen, beispielsweise maximal zwei Terme vor und hinter dem Toponym. Termkontexte können im Rahmen der Named Entity Recognition oftmals

28 Quelle: `http://de.wikipedia.org/wiki/Dresden_(Ohio)`

Hinweise auf den Typ der betrachteten Entität liefern. Rauch et al. (2003) werten Termkontexte aus, um Geo/Non-Geo-Ambiguitäten zu erkennen. So ist beispielsweise das Vorkommen „city“ innerhalb des Termkontexts ein positiver Indikator für eine Lokation, „Mr.“ hingegen ein negativer. Zong et al. (2005) nutzen Termkontexte für die Typklassifizierung. Für mehrdeutige Lokationsnamen wird versucht, eine regelbasierte Eingrenzung der Kandidatenmenge durch Kontextindikatoren vorzunehmen. So können beispielsweise bei „*state* of California“ oder „Rio Grande *County*“ jeweils jene Kandidaten eliminiert werden, die nicht vom Typ „State“ oder „County“ sind.

Ähnlich der Termkontexte können Muster eingesetzt werden, die Regeln für die Auflösung häufiger Sequenzen von Toponymen bestimmten Typs darstellen. Ein gängiges und einfaches Muster innerhalb von Texten ist eine explizite Disambiguierung eines Stadtnamens, indem nachfolgend explizit der Bundesstaat angegeben wird, wie beispielsweise bei „Chicago, IL“. Hier werden also hierarchische Eigenschaften ausgewertet. Eingesetzt wird dies beispielsweise von Smith und Crane (2001) oder Amitay et al. (2004). Muster können auch weitaus komplexer und spezifischer definiert werden, wie bei da Graça Martins (2008), die Regeln für Aufzählungen wie „cities such as *A*, *B*, and *C*“ definieren.

Korpora Ergänzend zu Gazetteer-Datenbanken können Textkorpora bei der Disambiguierung hilfreich sein. Sogenannte „Bootstrapping“-Methoden schlagen beispielsweise Rauch et al. (2003) vor. Eine große Menge ungelabelter Trainingsdaten wird genutzt, um den verwendeten Mechanismus zur Konfidenzbestimmung vorab zu trainieren. Häufig vorkommende Name-Toponym-Auflösungen werden gelernt und bei der Anwendung auf neuen Texten genutzt. Smith und Mann (2003) trainieren unter Verwendung von Textfeatures Klassifikatoren mit bereits disambiguierten Trainingsdaten (also Texten, in denen z. B. „Nashville, Tenn.“ explizit angegeben ist). Hier wird folglich der gesamte Dokumentinhalt als Kontext betrachtet. Peng, He und Mao (2006) werten Wahrscheinlichkeiten gemeinsamen Auftretens aus. Bei Betrachtung großer Dokumentkollektionen kann auf diese Weise zum Beispiel festgestellt werden, dass in Dokumenten „Washington, D. C.“ häufig auch „President Bush“ vorkommt. Die notwendigen Trainingsdaten wurden auf automatische Weise unter Beschränkung auf Toponymvorkommen gewonnen, die eindeutig aufgelöst werden können.

Innerhalb des Web-a-Where-Systems adressieren Amitay et al. (2004) die Problematik der Geo/Non-Geo-Disambiguierung durch Verwendung eines Korpus mit über 1 Mio. Seiten. Lokationsnamen, die im Korpus entweder selten als groß geschriebene Eigennamen auftauchen (wie z. B. „Humble“ in Texas) oder solche, deren Auftrittshäufigkeit disproportional zur Einwohnerzahl ist, werden als Nicht-Lokationen betrachtet, sofern nicht explizite gegenteilige Hinweise im Text gegeben sind.

Garbin und Mani (2005) verwenden auf einem Textkorpus gelernte Regeln zur Typklassifikation von Toponymen unter Verwendung des RIPPER-Klassifikators (Cohen, 1996). Die Klassifikation erfolgt lediglich in die drei Kategorien „civil“ (Region, Staat oder Land), „ppl“ (Stadt) und „cap“ (Hauptstadt).

Lieberman, Samet und Sankaranarayanan (2010) kombinieren innerhalb des NewsStand-Systems (Teitler et al., 2008) eine Reihe von Heuristiken mit zwei spezifischen „Lexicons". Ein „Global Lexicon" repräsentiert Toponyme von allgemeiner Bekanntheit, ein „Local Lexicon" hingegen enthält lokale Toponyminformationen, die in der Regel nur einer kleinen regionalen Zielgruppe bekannt sind. Die Verwendung des Local Lexicons bringt eine beachtliche Verbesserung des F1-Maßes bei der Disambiguierung von 64,5 % auf 88,5 % bei der Verwendung des stark lokal geprägten LGL-Datensets (siehe Abschnitt 3.4.1).

Anstelle der Nutzung lokaler Korpora greift das GeoScope-System von Qin, Xiao, Fang, Xie und Zhang (2010) auf Web-Suchmaschinen zurück, um repräsentative Kontextterme zur Disambiguierung zu ermitteln. Soll beispielsweise ein mehrdeutiges Toponym mit dem Namen „Gary" aufgelöst werden, für das Orte in den Bundesstaaten Minnesota und Indiana existieren, werden die Suchanfragen „Gary, MN" und „Gary, IN" vorgenommen. Zwischen den Texten der gefundenen Resultate und den Termkontexten wird die Kosinusähnlichkeit ermittelt und jene Entsprechung mit der höchsten Ähnlichkeit gewählt.

Verschiedene Arbeiten verwenden Informationen aus der Wikipedia. Overell und Rüger (2008) erzeugen ein Modell mit Wahrscheinlichkeiten gemeinsamen Auftretens von Lokationen, das für die Disambiguierung genutzt wird. Roberts, Bejan und Harabagiu (2010) nutzen ereignisspezifische Entitäten. Dazu wird eine bestehende Geo-Ontologie mit Entitäten vom Typ „Person" und „Organisation" verknüpft, die aus der Wikipedia gewonnen werden.

Kombinationen Eine Zusammenfassung der vorangehend vorgestellten verwandten Arbeiten findet sich in Tabelle 3.1. Wie sich zeigt, kombinieren die einzelnen Arbeiten in der Regel jeweils mehrere der vorgestellten Heuristiken, die in der Regel schrittweise angewendet werden. Vielfach eingesetzte Methoden zur Disambiguierung bedienen sich räumlicher Distanzen und hierarchischen Relationen. Teilweise verwenden die Ansätze zusätzliche Vorverarbeitungsregeln, so eliminiert beispielsweise Pouliquen et al. (2006) zunächst Entitäten, die als Teil von Personennamen im Text vorkommen und wendet eine Geo-Stoppwortliste an, um typische Falschextraktionen zu vermeiden. Beispielsweise existiert ein Ort mit dem Namen „And", der im Iran liegt, die Wahrscheinlichkeit einer Falsch-Positiv-Extraktion ist hier jedoch hoch, weshalb durch die Stoppwortliste von Pouliquen et al. (2006) die Extraktion solcher Orte komplett unterbunden wird.

Machine Learning Von der Möglichkeit, verschiedene Heuristiken als Features aufzufassen und mittels Machine Learning zu kombinieren, machen bisher nur Lieberman und Samet (2012) Gebrauch. Deren Arbeit stellt sogenannte „Adaptive Context Features" vor, die für jedes zu disambiguierende Toponym unter Verwendung einer festgelegten „Fenstergröße" aus dessen Textumfeld ermittelt werden. Ferner wird für den Kontext eine Tiefe festgelegt, die angibt, wie viele Kandidaten pro Toponym bei der Disambiguierung betrachtet werden. Durch die Wahl dieser Parameter kann direkt Einfluss auf die Verarbeitungsgeschwindigkeit des Systems Einfluss genommen werden. Insgesamt wurden durch Lieberman und Samet (2012) sieben Features extrahiert, wovon sich zwei auf

den adaptiven Kontext beziehen. Eines wird aus dem „Local Lexicon“ (Lieberman et al., 2010), die restlichen aus dem Gazetteer extrahiert.

Autor und Jahr	Räuml. Distanz	Hierarchie	Einwohnerzahl	Typ der Lokation	Lokale Einschr.	Kontexte und Muster	Korpus	Machine Learning
Smith und Crane (2001)	✓							
Li et al. (2002, 2003)		✓						
Rauch et al. (2003)	✓		✓			✓		
Smith und Mann (2003)							✓	
Leidner et al. (2003)	✓			✓				
Amitay et al. (2004)		✓	✓			✓	✓	
Zong et al. (2005)	✓	✓		✓		✓		
Garbin und Mani (2005)						✓		
Pouliquen et al. (2006)	✓	✓		✓	✓			
Peng et al. (2006)		✓	✓				✓	
Buscaldi und Rosso (2008a)		✓						
Buscaldi und Rosso (2008b)	✓							
da Graça Martins (2008)		✓	✓			✓		
Overell und Rüger (2008)							✓	
Andogah et al. (2008)		✓	✓	✓	✓		✓	
Qin et al. (2010)		✓				✓	✓	
Roberts et al. (2010)	✓	✓	✓					
Lieberman et al. (2010)	✓	✓				✓	✓	
Lieberman und Samet (2012)	✓	✓	✓			✓	✓	✓

Tabelle 3.1: Überblick über verwendete Methoden verwandter Arbeiten zur Lokationsextraktion und -disambiguierung (chronologisch sortiert)

3.2.2 WEB-APIS

Neben den im vorangegangenen Abschnitt vorgestellten akademischen Arbeiten zur Lokationsextraktion existieren eine Reihe von State-of-the-Art Web-APIs[29], die die Extraktion von Lokationen entweder als Teil eines Named-Entity-Recognition-Systems vornehmen oder sich ausschließlich auf die Extraktion von Lokationen beschränken. Die einzelnen APIs werden nachfolgend kurz vorgestellt und sofern verfügbar in den abschließenden Vergleich einbezogen. Bis auf Unlock sind sämtliche der erwähnten APIs kommerzielle Lösungen, die als „Black Boxes“ angesehen werden müssen. Über Interna der verwendeten Algorithmen und Datenbestände machen die Anbieter keine Angaben.

29 Application Programming Interface

Yahoo! BOSS Geo Services Yahoo[30] bietet mit der PlaceSpotter API einen Dienst zur Lokationsextraktion und -disambiguierung aus unstrukturiertem Text. Der verwendete Gazetteer von Yahoo umfasst laut eigenen Angaben sechs Millionen Einträge. Neben Englisch wird eine Reihe weiterer Sprachen unterstützt. Yahoo erlaubt 2.000 unentgeltliche Aufrufe des Dienstes pro Tag für nicht-kommerzielle Zwecke.

Unlock Unlock[31] stellt eine Web-API für den Edinburgh Geoparser[32] (Alex und Grover, 2010; Grover, Tobin, Byrne und Woollard, 2009; Tobin, Grover, Byrne, Reid und Walsh, 2010) zur Verfügung. Unlock arbeitet asynchron; die zu verarbeitenden Text-Dokumente müssen in einem ersten Schritt an den Webservice übermittelt werden. Die Verarbeitung nimmt dann einige Zeit in Anspruch; für ein in den nachfolgenden Abschnitten verwendetes Datenset aus ca. 150 Textdokumenten benötigte der Dienst rund eine halbe Stunde. Eine Typisierung der extrahierten Entitäten nimmt Unlock nicht vor.

OpenCalais OpenCalais[33] offeriert in seiner Metadata-API einen Allzweck-Named-Entity-Recognizer, der auch die Lokationstypen „continent“, „city“, „country“, „facility“, „naturalfeature“, „region“ und „provinceorstate“ beinhaltet. Für die meisten der extrahierten Lokationen stellt der Dienst auch die Koordinaten zur Verfügung.

AlchemyAPI Ähnlich OpenCalais erlaubt die AlchemyAPI[34] unter anderem die Extraktion geographischer Entitäten. Erkannt werden die Lokationstypen „city“, „country“, „facility“, „geographicfeature“, „region“ und „stateorcounty“. Obwohl die Webseite angibt, Koordinaten für Lokationen zu extrahieren, waren diese bei durchgeführten Tests nicht vorhanden.

Extractiv Extractiv[35] ist ein Service zur Named Entity Recognition in unstrukturiertem Text. Der Dienst hat mittlerweile offiziell seinen Betrieb eingestellt, die API war jedoch zum Zeitpunkt, als diese Arbeit entstand, weiterhin verfügbar[36]. Abgedeckt wird ein breites Spektrum von Entitätstypen, darunter auch 34 Typen mit Lokationsbezug. Nicht alle der extrahierten Lokationen werden in geographische Koordinaten disambiguiert.

MetaCarta MetaCarta[37] bietet mit „GeoTag“ eine Funktion zur Toponymextraktion und -disambiguierung für Textinhalte. Laut Angaben auf der Webseite steht ein Demo-Zugriff für Entwickler

30 http://developer.yahoo.com/boss/geo/
31 http://unlock.edina.ac.uk/home/
32 http://www.ltg.ed.ac.uk/clusters/Edinburgh_Geoparser
33 http://www.opencalais.com/documentation/calais-web-service-api/api-metadata/entity-index-and-definitions
34 http://www.alchemyapi.com/api/entity/
35 http://extractiv.com
36 Seit Anfang 2014 ist die ursprüngliche Webseite nicht mehr erreichbar.
37 http://www.metacarta.com

zur Verfügung, der Registrierung und anschließende Accountaktivierung funktioniert jedoch nicht. Folglich wird MetaCarta in der nachfolgenden Evaluierung nicht weiter betrachtet.

3.2.3 SOFTWAREBIBLIOTHEKEN

Neben den vorangehend genannten Web-APIs wurden im Rahmen der Recherche zwei Softwarebibliotheken für die Lokationsextraktion ausfindig gemacht.

CLAVIN Berico Technologies[38] bietet mit CLAVIN (Cartographic Location And Vicinity INdexer) ein Java-basiertes Open-Source-Paket für Geotagging und -parsing an. Für die Auflösung von Lokationen werden Kontextinformationen des Texts ausgewertet. Das Paket nutzt den Maximum-Entropy-basierten Named Entity Recognizer von Apache OpenNLP[39] für die Extraktion von Lokationskandidaten und den GeoNames-Gazetteer.

Fieldspring Fieldspring[40] ist ein Projekt des Computational Linguistic Labs der University of Texas zur Disambiguierung von Toponymen in Texten. Eine Dokumentation, die die notwendigen Schritte zur Inbetriebnahme des Systems erklärt, existiert jedoch nicht. Im Gegensatz zu CLAVIN wird Fieldspring somit im Rahmen des abschließenden Vergleichs nicht betrachtet.

3.2.4 ZUSAMMENFASSUNG VERWANDTER ARBEITEN

In den vorhergehenden Abschnitten wurden verwandte Arbeiten aus dem wissenschaftlichen Umfeld vorgestellt. In Abschnitt 3.2.1 wurde gezeigt, dass eine Reihe bewährter Methoden und Heuristiken existiert, die bereits auf unterschiedlichste Weise miteinander kombiniert wurden (Tabelle 3.1). Diese Kombinationen können gewissermaßen als Metaheuristiken angesehen werden. Hier zeigen sich jedoch auch Grenzen. Leidner (2007) vergleicht seine heuristische LSW03-Methode mit einer „Maximum Population"-Baseline, die er im Rahmen seiner Evaluierungen qualitativ nicht durchweg zu übertreffen vermag.

Keine der verwandten Arbeiten hat bis dato einen direkten Vergleich zwischen einem heuristisch und Machine-Learning-getriebenen Ansatz vorgenommen. Diese Arbeit stellt im Nachfolgenden jeweils einen Vertreter beider Kategorien vor und kann somit einen direkten Vergleich ziehen. Lieberman und Samet (2012) liefern erste Impule, heuristische Indikatoren zur Lokationsdisambiguierung mittels Machine Learning zu kombinieren. Die Autoren beschränken sich jedoch auf eine vergleichsweise kleine Menge von sieben Features. Im Rahmen des hier konzipierten Machine-Learning-basierten Verfahrens wird eine große Menge neuer Features für die Disambiguierung vorgestellt.

38 http://clavin.bericotechnologies.com
39 http://opennlp.apache.org
40 https://github.com/utcompling/fieldspring

3.3 EVALUIERUNGSANSÄTZE

Die nachfolgenden Abschnitte stellen die Evaluierungsmethoden vor, die im weiteren Verlauf dieses Kapitels verwendet werden. Die Evaluierung bewertet einerseits die Entity-Erkennung und -klassifikation, andererseits die korrekte Extraktion der Geo-Koordinaten. Beide Bewertungen nutzen die in Abschnitt 2.4 vorgestellten Maße Precision und Recall als Grundlage. Die Mittelung erfolgt nach Micro-Average-Kriterium, das heißt, dass die Werte für Precision und Recall auf Annotationsbasis und nicht auf Dokumentbasis gemittelt werden (Manning et al., 2009).

3.3.1 EVALUIERUNG VON NAMED ENTITY RECOGNITION

Die Resultate von Named Entity Recognizern werden üblicherweise mit manuell annotierten Referenzdaten verglichen. Der nachfolgende Beispieltext könnte beispielsweise einer manuellen Annotation entspringen (Nadeau, 2007):

```
Unlike <PER>Robert</PER>, <PER>John Briggs Jr</PER> contacted <ORG>Wonderful
Stockbrockers Inc</ORG> in <LOC>New York</LOC> and instructed them to sell all his
shares in <ORG>Acme</ORG>.
```

Der gleiche Text, welcher durch ein hypothetisches Named-Entity-Recognition-System annotiert wurde, sieht folgendermaßen aus:

```
<LOC>Unlike</LOC> Robert, <ORG>John Briggs Jr</ORG> contacted Wonderful
<ORG>Stockbrockers</ORG> Inc <PER>in New York</PER> and instructed them to sell all his
shares in <ORG>Acme</ORG>.
```

Durch die Verarbeitungsschritte „Delimitation“ und „Classification“ können jeweils unterschiedliche Fehler auftreten. Bei der Delimitation können einerseits irrtümlich falsche Kandidaten extrahiert oder tatsächlich korrekte Entitäten übersehen werden. Desweiteren besteht die Möglichkeit, dass die Start- bzw. Endpunkte der Kandidaten falsch bestimmt werden. In der Classification-Phase kann Kandidaten der falsche Typ zugeordnet bzw. dieser irrtümlich als Nicht-Entität klassifiziert werden. Tabelle 3.2 zeigt alle möglichen Fehlerfälle anhand des Beispieltexts von oben.

MUC Die in dieser Arbeit verwendete MUC-Methode quantifiziert Ergebnisse in den zwei Dimensionen „Typ“ und „Text“. Erste gibt an, ob der Typ korrekt klassifiziert wurde, zweite, ob die Start- und Endpositionen korrekt ermittelt wurden. Fehler, die in nur einer Dimension auftreten, werden als „Teilerfolg“ gewertet. Aus den möglichen Kombinationen ergeben sich pro Entität die in Tabelle 3.2 aufgelisteten sechs mögliche Ergebnistypen.

Sowohl für die Dimension „Typ“ als auch für die Dimension „Text“ wird jeweils die Anzahl korrekter Resultate (Correct), die Anzahl der vom Named-Entity-Recognition-System zurückgegebenen Resultate (Actual) und die Anzahl möglicher Entitäten im Ergebnis (Possible) ermittelt. Precision und Recall werden über beide Dimensionen ermittelt, wobei jeweils die Anzahlen addiert werden.

Typ	Referenz	Ergebnis	Beschreibung
–	`<ORG>Acme</ORG>`	`<ORG>Acme</ORG>`	Extraktion und Klassifikation korrekt
1	`On`	`<LOC>On</LOC>`	Entität markiert, an der sich tatsächlich keine befindet
2	`<PER>Robert</PER>`	`Robert`	Tatsächlich vorhandene Entität wurde nicht markiert
3	`<PER>John Briggs Jr</PER>`	`<ORG>John Briggs Jr</ORG>`	Entität korrekt markiert, aber falsches Label zugewiesen
4	`<ORG>Wonderful Stockbrockers Inc</ORG>`	`<ORG>Stockbrockers</ORG>`	Entität erkannt, Start- und/oder Endposition falsch
5	`<LOC>New York</LOC>`	`<DATE>in New York</DATE>`	Start- und/oder Endpositionen sowie Label falsch

Tabelle 3.2: Fehlerfälle bei Named Entity Recognition (Nadeau, 2007)

Im obigen Beispiel wurde zwei mal der richtige Typ und zwei mal die richtigen Start-/Endpositionen bestimmt. Somit ergibt sich die nachfolgende Kalkulation:

Correct = 2 Type + 2 Text = 4
Actual = 5 Type + 5 Text = 10
Possible = 5 Type + 5 Text = 10

Die Precision wird aus den Anzahlen für „Correct“ und „Actual“, der Recall aus den Anzahlen für „Correct“ und „Possible“ ermittelt (siehe Abschnitt 2.4).

Precision = # *Correct* / # *Actual* = 40 %
Recall = # *Correct* / # *Possible* = 40 %

3.3.2 EVALUIERUNG VON GEO-EXTRAKTION

Die vorangehend beschriebenen Evaluierungsmethoden beziehen sich auf allgemeine Named Entity Recognizer, nehmen jedoch keinen Bezug zu geographischen Daten. Wie bereits eingangs erläutert, liegt eine wesentliche Herausforderung in der korrekten Disambiguierung extrahierter Lokationen. Der nachfolgende Beispielsatz enthält mehrere Lokationen:

```
Paris, Texas is a city located 98 miles northeast of Dallas in Lamar County, Texas, in
the United States. The Sam Bell Maxey House is a historic house in Paris.
```

Trotz korrekter Erkennung und Klassifizierung von „Paris“ als Lokation wird die Stadt möglicherweise mit der Hauptstadt von Frankreich assoziiert. Zur Evaluierung muss also zusätzlich geprüft werden, ob tatsächlich die korrekte Instanz der Lokation ermittelt wurde. Aufgrund der geographischen Domäne liegt die Verwendung geographischer Distanzen für die Geo-Extraktion nahe. Leidner (2007) beispielsweise schlägt die Verwendung einer RMSE[41]-gewichteten Distanz, genannt „Root Mean Squard Distance“ vor, um größere Abweichungen stärker zu bestrafen als geringe. Eine einfachere Variante, wie sie beispielsweise von Lieberman und Samet (2012) verwendet wird, definiert einen Distanz-Schwellwert, unter dem die Extraktion als korrekt betrachtet wird. Somit können Precision, Recall und F1-Maß ermittelt werden. Lieberman und Samet (2012) evaluieren die Disambiguierung jedoch isoliert und ermitteln die Precision nur für bereits korrekt erkannte Toponyme, was durchweg zu hohen Precision-Werten führt.

Die Verwendung von Distanzen eignet sich für Lokationen, welche sich durch eine geringe Fläche auszeichnen und gut durch ihren Mittelpunkt repräsentiert werden können. Bei Objekten mit größeren Ausmaßen, wie Ländern, Kontinenten oder anderen geographischen Einheiten wie Flüssen oder Bergen ist eine Repräsentation als einzelner Punkt jedoch nicht möglich. Sind die Ausprägungen in den Referenzdaten durch Konturen gegeben, kann geprüft werden, ob sich erkannte Punkte innerhalb definierter Bereiche befinden. Solche Datenbestände sind jedoch unüblich. Eine weitere Variante ist eine rein symbolische Evaluierung, die die Übereinstimmung anhand von Lokationsidentifikatoren überprüft. Diese Vorgehensweise setzt jedoch einheitliche Datenbestände zwischen Evaluierungsset und verwendeten Extraktionsmechanismen voraus.

Precision-Geo, Recall-Geo Die Evaluierung in dieser Arbeit findet distanzbasiert mit Verwendung eines Schwellwerts statt und ermittelt darauf aufbauend Precision, Recall und F1. Die verwendeten Evaluierungsmaße werden nachfolgend mit „Precision-Geo“, „Recall-Geo“ und „F1-Geo“ bezeichnet. Im Gegensatz zu der Vorgehensweise von Lieberman und Samet (2012) werden jedoch sämtliche annotierten Toponyme in die Evaluierung einbezogen, um realistischere Qualitätsaussagen bezüglich des Gesamtsystems treffen zu können.

Tabelle 3.3 zeigt ein hypothetisches Ergebnis für den anfangs erwähnten Beispielsatz. Erkannt wurden hier fünf der sechs Lokationen, für die jeweils die Distanz zwischen Referenz- und extrahierter Lokation ermittelt wurde. „Paris“ wurde erkannt, jedoch fälschlicherweise als Hauptstadt von Frankreich disambiguiert. Für „Texas“ und „United States“ zeigt sich das Problem unterschiedlicher Datenbestände; bei Betrachtung der Distanzen zwischen Ergebnis- und Referenzkoordinate kann zwar von einer korrekten Extraktion ausgegangen werden, allerdings müssten hier die Schwellwerte entsprechend hoch gelegt werden, um entsprechende Abweichungen als korrekt zu bewerten. Dieses Problem kann umgangen werden, indem pro Lokationstyp spezifische Schwellwerte definiert werden (für Länder würde somit ein deutlich höherer Schwellwert gewählt). Da die Wahl geeigneter Schwellwerte jedoch zusätzlichen Aufwand mit sich bringt und zumindest bei Ländern Ambiguitäten ausgeschlossen

41 Root-mean-square error

Name	Typ	Referenz	Ergebnis	Distanz (km)
Paris	CITY ✓	(33,6625, –95,5477)	(48,8534, 2,3488)	7.782,67
Texas	STATE	(31, –100)	(31,2504, –99,2506)	76,58
Dallas	CITY ✓	(32,7758, –96,7967)	(32,7758, –96,7967)	0
Lamar County	UNIT	(33,67, –95,57)	(33,6668, –95,5836)	1,31
Texas	STATE	(31, –100)	(31,2504, –99,2506)	76,58
United States	COUNTRY	(39,76, –98,5)	(37,0902, –95,7129)	383,46
Sam Bell Maxey House	POI ✓	(33,6539, –95,555)	*missed*	–
Paris	CITY ✓	(33,6625, –95,5477)	(48,8534, 2,3488)	7.782,67

Tabelle 3.3: Beispiel für Geo-basierte Evaluierung (mit ✓ gekennzeichnete Einträge werden berücksichtigt)

werden können, wird in dieser Arbeit die metrische Evaluierung auf Lokationen vom Typ `CITY` und `POI` beschränkt (eine Definition der Typen folgt in Abschnitt 3.4.2). Falsch gesetzte Start- bzw. Endpositionen oder inkorrekte Typzuweisungen werden hier nicht als Fehler betrachtet, das heißt zum Beispiel, dass die extrahierte Lokation `<LOCATION>New York</LOCATION>` bei einer Referenzannotation `<CITY>New York City</CITY>` als korrekt gewertet wird, soweit die Distanz unter dem Schwellwert liegt.

Bei einem Distanzschwellwert von *maxDistance* = 100 km ergeben sich die nachfolgenden Mengen und die angegebenen Werte für Precision-Geo und Recall-Geo:

 # Correct = $|\{$ `Dallas` $\}|$ = 1
 # Actual = $|\{$ `Paris`, `Dallas`, `Paris` $\}|$ = 3
 # Possible = $|\{$ `Paris`, `Dallas`, `Sam Bell Maxey House`, `Paris` $\}|$ = 4
Precision-Geo = *# Correct* / *# Actual* = 33,33 %
Recall-Geo = *# Correct* / *# Possible* = 25 %

3.4 DATENSETS

Nachfolgend werden relevante Datensätze für die Evaluierung der Extraktion geographischer Informationen aus unstrukturierten Texten vorgestellt. Datensätze, welche für die Evaluierung allgemeiner Named-Entity-Recognition-Systeme gedacht sind und keine geographischen Daten enthalten, werden hierbei nicht berücksichtigt. Dies schließt beispielsweise das CoNLL 2003-Datenset[42] (Sang und Meulder, 2003) ein, welches lediglich über einen generischen „LOC"-Typ für Lokationen verfügt. Ferner beschränken sich die hier vorgestellten Datensets auf die englische Sprache. Nicht betrachtet werden darüberhinaus Datensets wie beispielsweise in Amitay et al. (2004), da hier kein a-priori annotierter Goldstandard vorlag, sondern Annotationsergebnisse des evaluierten Systems nachträg-

42 Conference on Computational Natural Language Learning

lich manuell als korrekt/falsch verifiziert wurden. Aufgrund dessen ist keine Reproduzierbarkeit der publizierten Ergebnisse möglich.

3.4.1 EXISTIERENDE DATENSETS

GeoSemCor GeoSemCor, erstmals vorgestellt in Buscaldi und Rosso (2008a), ist ein öffentlich verfügbarer Korpus[43]. Er basiert auf dem SemCor-Korpus der Princeton Universität, welcher ursprünglich für Evaluierungszwecke von WordNet geschaffen wurde. GeoSemCor fügt diesem Korpus explizite Annotationen für Toponyme hinzu. Diese Annotationen wurden jedoch auf automatischem Wege generiert – Wörter im Korpus, welche das Synset „location" als Hypernym besitzen, wurden mit einem Geo-Tag versehen. Das Datenset enthält 1.210 Toponyme, von denen 709 mehrdeutig sind. Die Autoren merken selbst die Unausgewogenheit des Korpus in Richtung der häufiger auftretenden Bedeutungen an. Viel mehr muss jedoch der grunsätzliche Sinn des Korpus in Frage gestellt werden – die vorhandenen geo-Annotationen beziehen sich ausschließlich auf große Städe, Länder und Bundesstaaten, welche in der WordNet-Datenbank verzeichnet sind. Kleinere Städte, bei denen uneindeutige Namen deutlich häufiger auftreten, kommen im Korpus überhaupt nicht vor.

TR-CoNLL Dieser Datensatz besteht aus englischen Nachrichtentexten aus dem CoNLL-Datenset von Reuters. In Leidner (2006) wird ausführlich die Erstellung des Datensets beschrieben. Enthalten sind 946 Dokumente mit 6.980 Toponym-Instanzen, davon sind 1.299 unterschiedlich. Ein weiterer Datensatz des gleichen Autors, „TR-MUC4", baut auf 100 Dokumenten von MUC-4 (Fourth Message Understanding Contest) auf (Leidner, 2007). Tr-CoNLL kann als akademische Lizenz für 550 Dollar direkt vom Urheber erworben werden. Leider hat der Urheber des Datensets nach initialer Kontaktaufnahme und Zusendung der Lizenzunterlagen auf Mails nicht mehr geantwortet[44], sodass das Datenset zur Evaluierung dieses Kapitels nicht vorlag.

Local-Global Lexicon Das Local-Global Lexicon (LGL) von Lieberman et al. (2010), besteht aus 588 Nachrichtenartikeln, die aus 78 Datenquellen bezogen wurden. Insgesamt soll der Datensatz 4.793 Toponyme enthalten. Die Inhalte stammen von kleineren Zeitungen mit lokaler Zielgruppe. Es wurden dazu zunächst gezielt kleinere Orte mit hoher Ambiguität ausgewählt (z. B. Paris, Texas; Paris, Tennessee; Paris, Illinois) und dann von lokalen Zeitungen nahe diesen Orten Artikel bezogen. Der Datensatz eigenet sich somit sehr gut für die Evaluierung der Toponymdisambiguierung bei lokal relevanten Inhalten, die Ambiguität entspricht somit jedoch nicht realistischen Gegebenheiten. Der Datensatz ist nicht frei verfügbar, wurde dem Autor für diese Arbeit jedoch auf persönliche Anfrage zur Verfügung gestellt. Eine genaue Analyse des Datensets zeigte jedoch eine wenig konse-

43 http://users.dsic.upv.es/grupos/nle/?file=kop4.php

44 Erste E-Mail vom Autor dieser Arbeit an den Urheber des Datensets: 16.04.2013, Eingangsbestätigung des Lizenzvertrags vom Urheber vom 28.07.2013, Mail mit Bitte um Rechnung für die Lizenzgebühren vom 06.08.2013, danach erfolgte keine Antwort mehr.

quente Annotation; viele Toponyme im Text sind nicht mit Markierungen versehen[45], bei anderen Toponymen sind Typ und/oder Markierungen falsch gesetzt[46] oder es wurden Entitäten anderen Typs irrtümlich als Lokationen markiert[47]. Außerdem wurden Adjektive und Demonyme (Volksbezeichnungen) als Toponyme markiert[48]. Auch im Text falsch geschriebene Toponyme sind nicht korrekt annotiert[49].

ACE SpatialML Der Datensatz „ACE 2005 English SpatialML Annotations" von Mani et al. (2009) besteht aus 428 Dokumenten. Die Dokumente entstammen einem breiten Spektrum unterschiedlicher Quellen, angefangen von Nachrichtenagenturen, transkribierten Nachrichtenbeiträgen, über Blogs bis hin zu Online-Newsgroups. Der Datensatz ist kostenlos für Mitglieder des Linguistic Data Consortium, für Nicht-Mitglieder kostet er hingegen 500 bzw. 1.000 Dollar[50].

TR-CLEF, TR-RNW Andogah (2010) stellt die zwei Korpora „TR-CLEF" und „TR-RNW" vor. Ersterer basiert auf Daten aus dem GeoCLEF-Korpus der CLEF-Initiative[51] (Conference and Labs of the Evaluation Forum). TR-RNW wurde aus Nachrichtenzusammenfassungen von Radio Netherlands Worldwide[52] erzeugt. Lokationen in den Datensätzen wurden durch Annotatoren unter Verwendung der Geonames-Datenbank Koordinaten zugeordnet. Anfragen per E-Mail an den Autor, mit dem Ziel das Datenset für die Evaluierung der vorliegenden Arbeit zur Verfügung zu stellen, wurde nicht beantwortet.

CLIR-WSD „CLIR-WSD"[53] ist ein allgemeiner Datensatz für Word Sense Disambiguation basierend auf dem GeoCLEF-Korpus, welcher Dokumente aus dem Glasgow Herald und der Los Angeles Times enthält. Ebenso wie GeoSemCor liegen WordNet-Synsets zugrunde, eine Disambiguierung erfolgte auf automatischem Wege.

Clust „Clust" von Lieberman und Samet (2011) beinhaltet populäre Nachrichten, die aus den erzeugten Clustern des NewsStand-Systems (Teitler et al., 2008) gewonnen wurden. Das Datenset beinhaltet 1.080 Cluster mit insgesamt 13.327 Artikeln. In jedem Cluster wurde genau ein Artikel zufällig ausgewählt, manuell annotiert und disambiguiert, sodass das Datenset über 1.080 annotierte

45 Beispiel für Inkonsistenz: Dokument 38765806 hat im Datensatz 20 Annotationen, jedoch sind nicht alle Straßennamen sind annotiert. Bei einer testweisen Annotation des gleichen Dokuments durch den Autor dieser Arbeit wurden 44 Toponyme entdeckt.

46 Beispiel für falschen Typ und Markierung: Für das „Woodstock General Hospital" in Dokument 38543488 ist nur der Bestandteil „Woodstock" als Stadt annotiert. Extraktionsverfahren, die korrekterweise die ganze Entität als Krankenhaus erkennen, würden dadurch bei der Evaluierung benachteiligt.

47 Beispiel für irrtümlich markierte Entitäten vom Typ „Nicht-Lokation": „Burgos Monumental" im Dokument 41432379 (Burgos Monumental ist ein Radfahrteam)

48 Beispiel für markierte Adjektive bzw. Demonyme: „Israeli" und „Palestinian" im Dokument 40648857

49 Beispiel für ein falsch klassifiziertes Toponym mit fehlerhafter Schreibung: „Lybia" im Dokument 44028853

50 `http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2011T02`

51 `http://www.clef-initiative.eu/`

52 `http://www.rnw.nl/english`

53 `http://ixa2.si.ehu.es/clirwsd/`

Texte verfügt. Das Datenset wurde, ebenso wie das oben erwähnte LGL, freundlicherweise von dessen Urheber zur Verfügung gestellt. Eine Überprüfung des Datensets ergab jedoch, dass die Annotationsqualität nicht den in dieser Arbeit gestellten Ansprüchen entspricht und auch hier die für das LGL-Datenset bereits angeführten Kritikpunkte gelten.

	Geo SemCor	**TR-CoNLL**	**LGL**	**ACE 2005**	**TR-CLEF**	**TR-NRW**	**CLIR-WSD**	**Clust**
Öffentlich	●	○	◐	○	○	○	○	◐
# Dokumente	186	946	588	428	321	556	169.477	1.082
# Toponyme	1.210	6.980	5.088	6.338	5.783	2.338	104.112	11.962
Types	○		●	●	○	○	○	●
Koordinaten	○		●	●	●	●	○	●
Scopes	○		○	○	○	○	○	○
Annotation	auto.	man.	man.	man.	man.	semiman.	auto.	man.
Qualität	?		◐	?	?	?	?	◐
Basis/Quellen	SemCor	CoNLL	Nachr.	gemischt	GeoCLEF	Nachr.	Nachr.	Nachr.

Abkürzungen: auto. = automatisch, man. = manuell, semiman. = semimanuell, Nachr. = Nachrichten
● = erfüllt, ◐ = teilweise erfüllt, ○ = nicht erfüllt

Tabelle 3.4: Übersicht über Datensets

3.4.2 TUD-LOC-2013

Aufgrund der vorher beschriebenen Defizite bezüglich Qualität und Verfügbarkeit anderer Datensets wurde mit „TUD-Location 2013“ (nachfolgend „TUD-Loc-2013“) ein neuer Goldstandard für die Evaluierung definiert. Die Vorgehensweise für die Erstellung des Datensets wird nachfolgend beschrieben. Neben dem Autor dieser Arbeit waren an der Annotation des Datensatzes zwei weitere Personen beteiligt. Ziel war, eine diversifizierte Sammlung von Texten zu erstellen, welche im Gegensatz zu den oben beschriebenen Datensets nicht nur eine feste Menge von Quellen oder eine spezifische Domäne wie Nachrichtenseiten abdeckt. Die Menge an möglichen Typen wurden initial definiert und ist in Tabelle 3.5 aufgelistet.

Ziel war hier einerseits, eine gute Unterscheid- und Abbildbarkeit möglicher Lokationstypen zu bieten, andererseits jedoch einen gewissen Pragmatismus walten zu lassen und die Anzahl überschaubar zu halten, um die Annotation einfacher zu gestalten. Beispielsweise abstrahiert der Typ `UNIT` administrative Elemente sämtlicher Hierarchieebenen, begonnen von Bundesstaaten bis hinunter zu Verwaltungsbezirken von Städten. Nicht getaggt wurden Ereignisse wie beispielsweise „Battle of Waterloo“, welche zwar an einem spezifischen Ort stattfanden, die jedoch selbst nicht als Lokations-

Typ	Beschreibung/Beispiel
CONTINENT	Kontinente wie beispielsweise „Asia“
COUNTRY	Länder wie beispielsweise „Japan“
CITY	Städte wie beispielsweise „Tokyo“
UNIT	Politische oder administrative Einheiten wie Bundesstaaten, Grafschaften oder Stadtbezirke wie beispielsweise „California“, „Bavaria“ oder „Manhattan“.
REGION	Gebiete, welche jedoch im Gegensatz zu UNIT keine politische oder admistrative Bedeutung besitzen oder sich über mehrere UNITs erstrecken.
LANDMARK	Geographische Objekte wie Flüsse, Berge, Seen, Schluchten, Ebenen wie beispielsweise „Rocky Mountains“.
POI	„Point of Interest“, also Objekte, die im Gegensatz zum Typ LANDMARK vom Menschen erbaut oder geschaffen wurden. Dies beinhaltet Gebäude wie Hotels, Universitäten, Krankenhäuser oder Bauwerke wie Monumente oder Plätze. Beispiele sind „Stanford University“ oder „Tahrir Square“.
STREET	Straßennamen wie zum Beispiel „Madison Avenue“
STREETNR	Hausnummern
ZIP	Postleitzahlen
UNDETERMINED	Entitäten, welche zwar zweifelsfrei als Lokationen identifiziert, deren genauer Typ jedoch nicht eindeutig war oder nicht näher bestimmt werden konnte.

Tabelle 3.5: Verwendete Lokationstypen

typ angesehen werden. Abgeleitete Formen wie Demonyme[54] (z. B. „Frenchman“, „Frenchwoman“) und Adjektive wurden nicht getaggt, da diese keinen direkten Ortsbezug haben (z. B. in „German Chancellor arrives at airport“). Wurden Ortsnamen als sprachliche Stilmittel verwendet – sogenannte Metonyme oder Synekdochen wie z. B. „Moscow criticized Turkey's plans“, wo „Moscow“ und „Turkey“ für die Regierungen der beiden Länder stehen – so wurden diese annotiert[55]. Nicht annotiert wurden ferner Lokationen, die Bestandteil einer Entität anderen Typs sind (z. B. „New York Times“ oder „Voice of Korea“).

Die Annotation fand im Zeitraum von Dezember 2012 bis April 2013 statt. Die Webseiten, von denen die annotierten Texte entnommen wurden, wurden teilweise mithilfe von Suchmaschinen wie Google oder Bing und Anfragen nach dem Schema „COUNTRY news“ oder „COUNTRY blog“ gesucht und teilweise aus Beiträgen der Aggregationsplattform reddit.com[56] ausgewählt. Zwei Bedingungen

54 Volksbezeichnungen mit Staatsbezug

55 Eine detaillierte Betrachtung über unterschieldiche Arten von Metonymie bei Ortsnamen und einen Datensatz mit statistischer Auswertung liefern Markert und Nissim (2002)

56 http://www.reddit.com

für die Auswahl waren, dass die Seite in englischer Sprache war und dass sich der Inhalt auf ein spezifisches Thema beschränkte. Startseiten wurden folglich nicht in das Datenset aufgenommen, sondern ausschließlich Inhalts- und Artikelseiten. Aus jeder so ausgewählten Seite wurde manuell der Hauptinhalt extrahiert. Als „Hauptinhalt" wird hier die Überschrift und der Fließtext der Seite bezeichnet, unerwünschte Inhalte sind dagegen Werbebanner, Navigationsbereiche, Header und Footer, Nutzerkommentare oder Bildunterschriften.

Die anschließende Annotation der Texte erfolgte unter Verwendung von XML-Tags. Eine Besonderheit stellt das Attribut `role` dar – auf diese Weise wurde im Datenset der geographische Fokus jedes Dokuments gekennzeichnet, welcher für die Evaluierung der Fokusbestimmung eine Rolle spielt. Pro Dokument wurde maximal eine Lokation mit dem Attributwert `main` gekennzeichnet, die Entscheidung, ob ein Dokument eine Haupt-Lokation besitzt, wurde jedoch den Annotatoren überlassen. Der nachfolgende Absatz zeigt exemplarisch einen Ausschnitt aus dem Datenset:

```
Tiny <LANDMARK>Heir Island</LANDMARK> - one of the many isles that are scattered across
County <CITY>Cork</CITY>'s <LANDMARK>Roaring Water Bay</LANDMARK> in <COUNTRY
role="main">Ireland</COUNTRY>'s southwest - is one of the country's go-to gourmet spots.
So you will need to book months in advance to dine at <POI>Island Cottage</POI>, a
restaurant run by the husband-and-wife team John Desmond and Ellmary Fenton. [...]
```

Das Datenset besteht aus insgesamt 152 Texten, eine Index-Liste enthält zusätzlich die Quell-URLs der Webseiten sämtlicher enthaltener Texte. Tabelle 3.6 zeigt die Häufigkeit der annotierten Lokationstypen.

In einem zweiten Schritt wurden die Annotationen des Datensets manuell mit geographischen Koordinaten versehen. Mit einer webbasierten Annotationsapplikation wurden den Annotationen tatsächliche Punkte auf der Weltkarte zugeordnet. Die entsprechenden Koordinaten konnten per Textsuche in den Datenbanken von Geonames (siehe Abschnitt 3.2.2) und per Google Geocoding API[57] gesucht und anschließend die korrekte Lokation auf einer Weltkarte markiert werden. Um die Auswahl durch den Nutzer zu erleichtern, wurden zusätzliche Informationen wie Typ und Einwohnerzahl der aktuell ausgewählten Lokation angezeigt. Annotationen, welche in keiner der beiden Datenbanken gefunden wurden, wurden als „unmatched" markiert.

Der Suchbegriff konnte per Suchfeld jeweils manuell angepasst und verfeinert werden; beispielsweise wird für eine Annotation mit dem Wert „Atlantic" kein Treffer gefunden, da sich aus dem Kontext des Textes aber zweifelsfrei ergibt, dass der Atlantische Ozean gemeint ist, waren die Annotatoren dazu angehalten, die Suchanfrage entsprechend zu modifizieren und nach „Atlantic Ocean" zu suchen.

Von den 3.814 Annotationen im Datenset konnten 90,51 % mit Koordinaten versehen werden. 362 Annotationen blieben ohne Koordinatenzuordnung; dies betrifft zum größten Teil den Annotationstyp `POI`, also beispielsweise Lokationen wie Restaurants, welche nicht in den abefragten Datenbanken

57 https://developers.google.com/maps/documentation/geocoding/

	Gesamt		Untersch.		Mit. Koord.	
Typ	**#**	**%**	**#**	**%**	**#**	**%**
CONTINENT	72	1,89	6	0,43	72	100
COUNTRY	1.486	38,96	147	10,49	1.482	99,73
CITY	1.031	27,03	401	28,62	1.008	97,77
UNIT	242	6,35	131	9,35	233	96,28
REGION	139	3,64	83	5,92	108	77,7
LANDMARK	281	7,37	183	13,06	236	83,99
POI	454	11,9	355	25,34	231	50,88
STREET	55	1,44	45	3,21	38	69,09
STREETNR	37	0,97	33	2,36	28	75,68
ZIP	17	0,45	17	1,21	16	94,12
# Annotationen	3.814	100	1.401	100	3.452	90,51

Tabelle 3.6: Häufigkeit der vorkommenden Lokationstypen und Anzahl der mittels Koordinaten disambiguierten Lokationen in TUD-Loc-2013; die Prozentwerte für „Gesamt“ und „Untersch.“ beziehen sich jeweils auf die Summe aller Annotationen, bei „Mit Koord.“ hingegen geben die Prozentwerte den Anteil an den in der ersten Spalte aufgelisteten Anzahl „Gesamt“ an.

verfügbar waren. Tabelle 3.6 zeigt eine Aufschlüsselung nach Typ der mittels Koordinaten disambiguierten Lokationen. Das Resultat liegt als separate CSV-Datei[58] vor, welche aus Referenzen auf das jeweilige Dokument (Dateiname, laufender Index und Zeichenoffset der Annotation), den geographischen Koordinaten und quellspezifischen Identifikatoren (wie z. B. `geonames:2963597`) besteht. Abbildung 3.2 zeigt die geographische Verteilung der annotierten Lokationen. Von den 152 Dokumenten wurden in 129 (84,87 %) mittels `role="main"` der geographische Fokus annotiert.

Im Interesse der Transparenz und Vergleichbarkeit der nachfolgend gezeigten Resultate wird TUD-Loc-2013 auf der Plattform Areca[59] veröffentlicht. Die 152 enthaltenen Dokumente sind dort bereits in die drei disjunkten Teilmengen „Traingsset“, „Validierungsset“ und „Testset“ im Verhältnis 40:20:40 aufgeteilt.

3.5 TERMINOLOGIE

Essentielle Begriffe, die in den nachfolgenden Abschnitten verwendet werden, sollen nachfolgend jeweils anhand eines Beispiels erläutert werden. Als Grundlage dient der folgende kurze Text:

```
Paris is a city located 98 miles northeast of Dallas in Lamar County, Texas, in the U.S.
The Sam Bell Maxey House is a historic house in Paris.
```

58 Comma-separated values

59 http://areca.co/21/TUD-Loc-2013-location-extraction-and-toponym-disambiguation-dataset

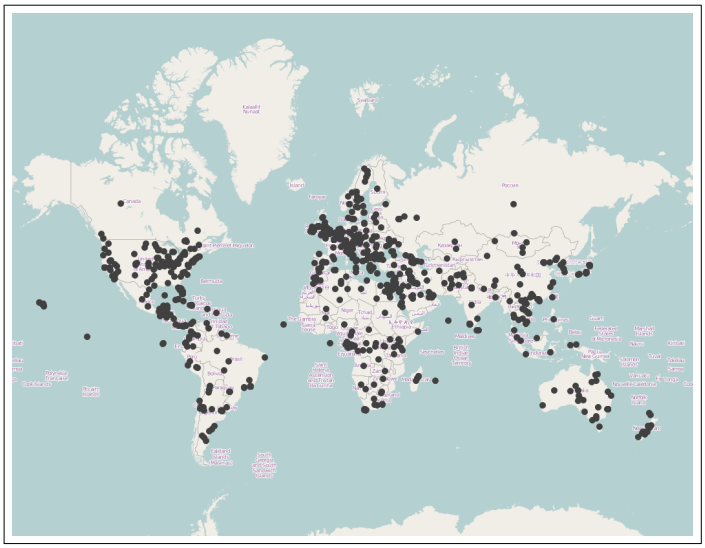


Abbildung 3.2: Verteilung der Lokationen in TUD-Loc-2013 (Kartenmaterial von OpenStreetMap, 2013)

Annotationen Als Annotationen werden Textsegmente bezeichnet, mit denen Entitäten gekennzeichnet werden, die Kandidaten für Lokationsreferenzen darstellen. Diese werden anhand ihrer Position (als Zeichenoffset vom Beginn des Texts) und dem textuellen Inhalt beschrieben. Für den obigen Text besteht die Menge der Annotationen A aus den Elementen:

$$A = \{(0, \texttt{Paris}), (46, \texttt{Dallas}), (56, \texttt{Lamar County}), (70, \texttt{Texas}), (84, \texttt{U.S.}),$$
$$(93, \texttt{Sam Bell Maxey House}), (137, \texttt{Paris})\}$$

Lokationskandidaten Die Menge von Kandidaten $L(a_n)$ für eine Annotation $a_n \in A$ sind mögliche Interpretationen, die tatsächliche Orte auf der Landkarte repräsentieren. Falls für eine Annotation keine Entsprechung in der Datenbasis gefunden wurde, entspricht die Kandidatenmenge der leeren Menge. Die Lokationstupel im angegebenen Beispiel bestehen der Reihe nach aus einem eindeutigen Identifikator, einem Lokationstyp, der Einwohnerzahl und geographischer Breite und Länge. L entspricht der Menge sämtlicher Lokationen für A. Für die Annotationen $a_2 = (\texttt{56}, \texttt{Lamar County})$ und $a_5 = (\texttt{93}, \texttt{Sam Bell Maxey House})$ sehen die Kandidatenmengen wie folgt aus:

$$L((\texttt{56}, \texttt{Lamar County})) = \{ (\texttt{4071676}, \texttt{UNIT}, \texttt{14564}, \texttt{33.75178}, \texttt{-88.11670}),$$
$$(\texttt{4204869}, \texttt{UNIT}, \texttt{18317}, \texttt{33.06679}, \texttt{-84.14992}),$$
$$(\texttt{4432922}, \texttt{UNIT}, \texttt{55658}, \texttt{31.23018}, \texttt{-89.45507}),$$
$$(\texttt{4705086}, \texttt{UNIT}, \texttt{49793}, \texttt{33.66677}, \texttt{-95.58357}) \}$$
$$L((\texttt{93}, \texttt{Sam Bell Maxey House})) = \varnothing$$

Das heißt, dass vier mögliche Orte mit dem Namen „Lamar County“ und kein Ort mit dem Namen „Sam Bell Maxey House“ gefunden wurde.

Disambiguierung Während der Disambiguierung wird für jede Annotation $a \in A$ ein geeigneter Lokationskandidat $l \in L(a)$ ausgewählt, falls $|L(a)| > 0$. Das Ergebnis der Disambiguierung für einen Text ist somit eine Menge D bestehend aus Paaren aus Annotation a und Lokationskandidat l. Annotationen, welche nicht aufgelöst werden konnten, sind nicht in D enthalten. Gleiches gilt, falls die Disambiguierungsstrategie sämtliche Lokationskandidaten als unwahrscheinlich verwirft.

$D = \{$ ((0, Paris), (4717560, CITY, 25171, 33.66094, -95.55551)),
((46, Dallas), (4684888, CITY, 1197816, 32.78306, -96.80667)),
((56, Lamar County), (4705086, UNIT, 49793, 33.66677, -95.58357)),
((70, Texas), (4736286, UNIT, 22875689, 31.25044, -99.25061)),
((84, U.S.), (6252001, COUNTRY, 310232863, 39.76, -98.5)),
((137, Paris), (4717560, CITY, 25171, 33.66094, -95.55551)) $\}$

Für die Kandidatenmenge einer Annotation a existieren $|L(a)| + 1$ unterschiedliche mögliche Interpretationen. Die Anzahl möglicher Disambiguierungen D pro Text ist folglich das Produkt einzelner möglicher Interpretationen für jede Annotation im Text $\prod_{a \in A} |L(a) + 1|$.

3.6 GAZETTEER

Die nachfolgend beschriebenen Verfahren nutzen einen Gazetteer, der aus verschiedenen Quellen aggregiert wurde. Im Gegensatz zu verschiedenen State-of-the-Art-Ansätzen, die vergleichbar kleine Datenbanken nutzen, soll in dieser Arbeit durch eine große Datenbank ein hoher Recall garantiert werden. Der Gazetteer stellt die nachfolgenden Informationen bereit:

id Ein eindeutiger Identifikator der Lokation

primaryName Der primäre Name der Lokation in lokaler Form, z. B. „München“

alternativeNames Eine Menge alternativer Namen und Varianten in anderen Sprachen, z. B. „Munich“, „Monaco“ etc.

type Die Art der Lokation, z. B. `CITY`

longitude, latitude Die geographischen Koordinaten der Lokation, z. B. (48.13743, 11.57549)

population Abhängig vom Typ die Einwohnerzahl der Lokation

hierarchy Eine Hierarchie der Lokation basierend auf administrativen bzw. politischen Gegebenheiten, z. B. „München, Landeshauptstadt → Kreisfreie Stadt München → Upper Bavaria → Freistaat Bayern → Federal Republic of Germany → Europe → Earth“

Zum Zeitpunkt der Experimente enthielt die Datenbank genau 9.276.864 Lokationen sowie 7.014.955 Alternativnamen. Die Inhalte stammen aus den nachfolgenden Quellen:

GeoNames Lokationen von GeoNames[60] machen mit über 8,5 Mio. Einträgen den Hauptanteil in der Datenbank aus. GeoNames ist eine unter Creative-Commons-Lizenz[61] verfügbare Geo-Datenbank, welche nach Wiki-Prinzip gepflegt wird. Aufgrund der stark unterschiedlichen Datenbank-Schemata und im Detail unzureichenden Dokumentation, gestaltete sich die Überführung in die verwendete Datenbank alles andere als trivial.

HotelsBase Eine öffentliche Datenbank mit über 500.000 Hotels[62].

protectedplanet.net Eine Datenbank mit über 200.000 naturbezogenen Lokationen wie Parks, Naturschutzgebieten und Denkmälern[63].

Wikipedia Mit dem Ziel, zusätzliche Orte wie Universitäten und Freizeitparks abzudecken, die nicht in GeoNames vorhanden sind, wurden insgesamt 218.428 Lokationen aus der Wikipedia bezogen. Die Daten werden aus dem englischsprachigen Datenbankdump[64] unter Verwendung des MediaWiki-Parsers aus dem Palladian-Toolkit (Urbansky, Muthmann, Katz und Reichert, 2012) extrahiert. Berücksichtigt wurden ausschließlich Artikel, die einerseits explizit angegebene Geo-Koordinaten in Form des `coord`-Tags[65] beinhalten, die oben rechts auf der Artikelseite angezeigt werden, andererseits über eine sogenannte Infobox verfügten. Durch den Typ der Infobox können Seiten klassifiziert werden. So existieren spezifische Infoboxen für eine Vielzahl von Entitätstypen, beispielsweise für Personen, Städte, Kriege etc. Aus den am häufigsten vorkommenden Infobox-Typen wurden letztlich 103 lokationsspezifische ausgewählt und manuell ein Mapping auf die hier verwendeten Typen (siehe Tabelle 3.5) definiert (z. B. „settlement“ → `CITY`, „french commune“ → `CITY`, „mountain“ → `LANDMARK` etc.).

Eine weitere frei verfügbare und umfangreiche Geo-Datenbank, welche hier jedoch nicht berücksichtigt wurde, existiert mit dem „NGA GEOnet Names Server“[66]. Sie besteht aus etwas über 9 Mio. Einträgen. Die „Free World Cities Database“[67] enthält ca. 3,2 Mio. Städte. Die GeoPlanet-Daten von Yahoo![68] umfassen 5,7 Mio. Orte. Stichprobenartige Tests zeigten jedoch, dass eine Erweiterung um die genannten Datenbanken nicht signifikant mehr zusätzliche Orte liefert.

3.7 VORVERARBEITUNG

Die nachfolgenden Abschnitte 3.8 and 3.9 beschreiben zwei Konzepte zur Toponymerkennung und -disambiguierung. Beiden Ansätzen liegt die gleiche Vorverarbeitungsphase zugrunde, die

60 `http://www.geonames.org`
61 `http://creativecommons.org`
62 `http://www.hotelsbase.org`
63 `http://protectedplanet.net`
64 `http://dumps.wikimedia.org/enwiki/latest/`, verwendet wurde die Version vom 04.05.2013
65 Beispiel: `{{Coord|51|03|10|N|13|44|33|E|region:DE-SN_type:landmark|display=title}}` auf `http://en.wikipedia.org/wiki/Dresden_Academy_of_Art`
66 `http://earth-info.nga.mil/gns/html/namefiles.htm`
67 `http://www.maxmind.com/en/worldcities`
68 `https://developer.yahoo.com/geo/geoplanet/data/`; momentan nicht mehr offiziell verfügbar, jedoch aufgrund der „CC BY 3.0“-Lizenz via `http://archive.org/search.php?query=geoplanet` zu beziehen

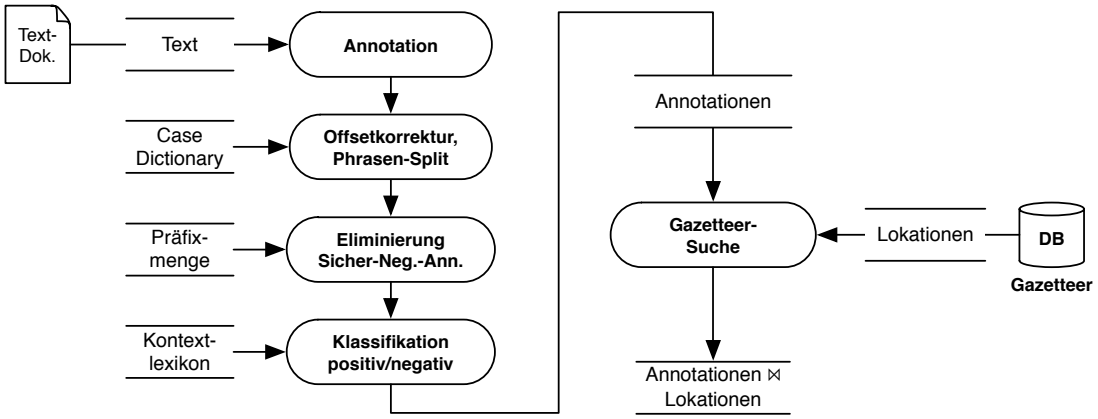


Abbildung 3.3: Schritte der Vorverarbeitung zur Lokationsextraktion

im Wesentlichen darin besteht, potentielle Entitäten im Text zu erkennen, korrekt zu markieren und unerwünschte Kandidaten zu filtern. Abbildung 3.3 zeigt den Ablauf der Vorverarbeitung. Wie nachfolgend beschrieben, sind die hier angewendeten Schritte bewusst einfach und universell gehalten. Nicht alle in dieser Phase identifizierte Kandidaten müssen tatsächlich Lokationen sein, vielmehr ist die Vorverarbeitung auf einen hohen Recall optimiert und verschiebt im Zweifelsfall die Entscheidung, Kandidaten als Nicht-Lokationen auszufiltern in die nachfolgende Disambiguierungsphase.

Annotation Die Extraktion erfolgt regelbasiert unter der Verwendung regulärer Ausdrücke. Da der Extraktionsmechanismus für die Verwendung englischer Dokumente ausgelegt ist, kann der Kandidaten-Tagger sämtliche Terme und Termgruppen innerhalb des Eingabetextes, welche mit Großbuchstaben beginnen, extrahieren. Ferner existiert eine Reihe von Heuristiken, welche spezifische Wortgruppen, beispielsweise mit Präpositionen oder besonderen Zeichen, erkennen (z. B. „United States of America“, „Rue de Rivoli“ oder „Grand Tradition Estate & Gardens“).

Offsetkorrektur und Trennung langer Phrasen Aufgrund der Großschreibung von Wörtern am Satzanfang enthält die Kandidatenmenge viele Nicht-Eigennamen. Beispielsweise wird aus dem Satz „This club has a short history of 12 years.“ irrtümlich der Term „This“ als Eigenname extrahiert. Durch ein sogenanntes „Case Dictionary“ können solche Kandidaten eliminiert werden. Das Case Dictionary, dessen Konzept bereits von Millan, Sánchez und Moreno (2008) verwendet wurde, stellt einen Korpus von Termen dar, welcher jeweils Vorkommenshäufigkeiten der klein- und großgeschriebenen Variante enthält (siehe Tabelle 3.7 für ein Beispiel).

Alle Kandidaten, für die $uppercaseRatio \leq 0{,}5$ gilt, werden aus der Kandidatenmenge entfernt. Falsch gesetzte Startpositionen potentieller Phrasen werden ebenfalls mittels Case Dictionary korrigiert. Die Kandidatenextraktion extrahiert beispielsweise groß geschriebene Satzanfänge, denen Eigennamen folgen als zusammenhängende Einheit. Aus dem Satz „Tiny Heir Island is one of the country's go-to

Token	# total	# uppercase	Ratio
arabia	1.396	1.396	1
guide	2.088	800	0,38
saudi	1.315	1.315	1
travel	4.701	202	0,04
tiny	1.116	80	0,07
your	4.324	1.105	0,26

Tabelle 3.7: Beispiel für Case Dictionary

gourmet spots.“ beispielsweise wird durch die hier verwendeten Regeln irrtümlich „Tiny Heir Island“ als Kandidat extrahiert. Durch die Offset-Korrektur wird der Kandidat auf „Heir Island“ verkürzt. Das verwendete Case Dictionary wurde aus englischen Wikipedia-Artikeln gewonnen und besteht aus 91.407 Einträgen.

Das Case Dictionary wird ferner dafür genutzt, lange groß geschriebene Phrasen, wie sie mitunter in Überschriften vorkommen, in kleinere Einheiten zu zerlegen. Sofern Annotationen aus mehr als drei Tokens bestehen, werden diese zusätzlich in ihre Bestandteile zerlegt. Dazu wird eine Trennung in Sequenzen aus Tokens vorgenommen, für die t *uppercaseRatio* > 0,5 gilt. Mit dem in Tabelle 3.7 abgebildeten beispielhaften Case Dictionary würde aus der großgeschriebenen Phrase „Your Saudi Arabia Travel Guide“, die Sub-Annotation „Saudi Arabia“ extrahiert.

Es zeigt sich, dass sich in der verbleibenden Kandidatenmenge viele irrtümlich extrahierte Personennamen befinden, die der Precision des Extraktionsergebnisses schaden. In zwei weiteren Schritten sollen deshalb offensichtliche Personenentitäten (Sicher-Negativ-Kandidaten) direkt entfernt bzw. potentielle Personennamen (Potentiell-Negativ-Kandidaten) als solche markiert werden.

Eliminierung von Sicher-Negativ-Annotationen Der erste Schritt zur Identifikation von Personennamen erfolgt durch eine manuell gepflegte Liste eindeutiger Präfixe. Kandidaten mit diesen Präfixen werden unmittelbar aus der Kandidatenmenge eliminiert. Dies schließt auch spätere Repetitionen mit ein; so werden bei der Eliminierung des Kandidaten „Mr. John Smith“ auch sämtliche Vorkommen von „John“ und „Smith“ aus der Kandidatenmenge entfernt. Dies entspricht der Annahme „eine Bedeutung pro Diskurs“ (Gale et al., 1992), das heißt, dass davon ausgegangen wird, dass sämtliche Vorkommen von „John“ sich auf die identifizierte Person beziehen. Die verwendete Liste besteht aus 13 personentypischen Präfixen[69].

Klassifikation von Potentiell-Positiv- und Potentiell-Negativ-Annotationen Kontexte dienen dazu, Kandidaten zunächst in die Klassen „Lokation“ bzw. „Nicht-Lokation“ zu klassifizieren, ohne diese sofort aus der Kandidatenmenge zu entfernen. Als Beispiel für eine kontextbasierte Klassifikation kann der Satz „Georgia attended the conference“ betrachtet werden. Das nachfolgende

69 Dr, Dr., Lady, Mr, Mr., Mrs, Mrs., Ms, Ms., Prof, Prof., Professor, Sir

Token „attended“ deutet darauf hin, dass es sich bei der Entität „Georgia“ um eine Person handelt, wohingegen im Satz „Georgia president concedes election defeat“ das Token „president“ ein starkes Anzeichen dafür ist, dass der vorangehende Kandidat eine Lokation ist. Das hier verwendete Kontextlexikon wurde aus einer großen Menge von personen- und lokationsspezifischen Texten gewonnen. Die Vorgehensweise orientiert sich an der von Etzioni et al. (2005): Auf manuellem Wege wurde eine Saat-Liste mit Personen- und Lokationsentitäten erzeugt (z. B. „Dmitry Medvedev“ oder „Great Basin Desert“). Dabei wurde darauf geachtet, nur solche Entitäten in die Liste aufzunehmen, bei denen keine offensichtlichen Typambiguitäten bestehen (wie dies zum Beispiel bei „Georgia“ der Fall ist).

Verwendet wurden pro Typ 800 Saat-Entitäten, deren Namen jeweils als Saat-Anfragen für Bing[70] dienten. Pro Saat-Entität wurden jeweils maximal 100 URLs gesucht. Für den Entitätstyp `PERSON` wurden 29.642 HTML-Seiten, für `LOCATION` 33.454 Seiten heruntergeladen. Von jeder Seite wurde mittels Inhaltsextraktor aus Palladian (Urbansky, Muthmann, Katz und Reichert, 2012) der Haupt-Textblock extrahiert, zu kurze Texte unter 100 Zeichen ausgefiltert bzw. kurze Fragmente innerhalb der Texte entfernt und anschließend alle vorkommenden Entitäten im Text markiert (z. B. „`<PERSON>`Dilma Rousseff`</PERSON>`, the president of `<LOCATION>`Brazil`</LOCATION>` […]“). Somit entstand ein Datenset bestehend aus 17.215 Texten mit 126.377 annotierten `PERSON`- und 23.830 Texten mit 184.841 annotierten `LOCATION`-Entitäten.

Aus dem so gewonnenen Datenset wurden anschließend für beide Entitätstypen Kontexte auf unterschiedliche Weise extrahiert: Experimentiert wurde mit unterschiedlichen Kontextgrößen (n-Gramme mit $n = [1 \dots 4]$ vor/nach der Entität) und einem „Fuzzy Matching“ (Term kommt innerhalb Fenstergröße n vor). Eine feste Kontextgröße von 1 bewährte sich experimentell am Besten. Auch hier wurde die Annahme „eine Bedeutung pro Diskurs“ getroffen, das heißt, dass die hier vorgenommene Klassifikation an andere Vorkommen mit gleichem Namen weitergegeben wird.

Die linken und rechten Kontexte wurden separat analysiert und nur solche, welche mit einer Wahrscheinlichkeit von über 90 % für eine der beiden Klassen „Lokation“ bzw. „Person“ vorkamen, in ein Kontextlexikon aufgenommen[71]. Das erzeugte Kontextlexikon besteht aus 318 Präfix- und Suffix-Kontexten für die Typen `PERSON` und `LOCATION`. Wie bereits eingangs erwähnt, erfolgt keine direkte Filterung anhand der Kontexte. Die Motivation hinter dieser „Deferred Commitment“-Strategie ist, dass die Typklassifikation nicht perfekt funktioniert und durch gängige, vor allem im journalistischen Umfeld angewendete, sprachliche Stilmittel wie Metonymie erschwert wird. Beispielsweise würde die Kontextklassifikation im Satz „U. S. says Rwanda aids Congo rebels“ die Entität „U. S.“ aufgrund des Suffixes „says“ als Person klassifiziert. Durch die Verlagerung der Entscheidung, ob der jeweilige Kandidat ausgefiltert werden soll oder nicht, kann später auf Informationen aus dem Gazetteer

70 `http://www.bing.com`

71 Der Schwellwert von 90 % wurde bewusst konservativ gewählt; durch Variation können hier eventuell noch bessere Resultate erzielt werden.

zurückgegriffen werden, um trotz der Klassifikation als Nicht-Lokation probate Ausnahmen zu machen, beispielsweise bei bedeutenden Lokationen wie Ländern oder Hauptstädten.

Gazetteer-Suche Im letzten Schritt der Vorverarbeitungsphase werden die annotierten Kandidaten im Gazetteer gesucht. Für jede verbleibende, eindeutige Annotation a_n wird aus der Datenbank eine Menge L_n von Lokationskandidaten mit passendem Namen abgefragt. Dabei werden die in Abschnitt 3.6 beschriebenen primären und Alternativnamen berücksichtigt. Um den Recall zu erhöhen werden diakritische Zeichen wie Akzente für die Übereinstimmung ignoriert. Die nachfolgend beschriebenen Strategien zur Toponymdisambiguierung erhalten eine Liste aller Annotationen $A = \{a_1, \ldots, a_n\}$ und die aus dem Gazetteer extrahierten Lokationskandidaten L. Pro Annotation $a_n \in A$ wird dann entweder ein Lokationskandidat $l \in L$ ausgewählt oder der Kandidat als Nicht-Lokation ausgeschlossen (siehe Abschnitt 3.5.

3.8 HEURISTISCHE ERKENNUNG UND DISAMBIGUIERUNG

Dieser Abschnitt beschreibt die konzipierte heuristische Strategie zur Lokationsdisambiguierung. Diese macht von zwei Mechanismen Gebrauch, die zunächst nur solche Lokationen extrahieren, bei denen eine hohe Precision garantiert werden kann. Die erste Strategie nutzt dazu sogenannte „Anker-Lokationen“[72]. Anker-Lokationen sind bedeutende (z. B. Länder oder Kontinente) oder eindeutig identifizierbare Orte. In dem Fall, dass in einem Text keine Anker-Lokationen extrahiert werden konnten, wird eine sogenannte „Lasso-Methode“ angewendet. Diese Methode zielt auf stark lokal geprägte Texte, die ausschließlich kleine, unbekannte und mehrdeutige Lokationen enthalten. Die nachfolgenden Abschnitte beschreiben zunächst beide Methoden, abschließend wird die komplette Disambiguierungsstrategie beschrieben.

Anker-Methode Für die Bestimmung von Anker-Lokationen kommen drei einfache Heuristiken zum Einsatz:

1. Die Lokation hat eine Einwohnerzahl, die über *anchorPopulation* liegt.
2. Die Lokation ist vom Typ `CONTINENT` oder `COUNTRY`.
3. Die Lokation ist eindeutig in der Datenbank und hat einen „markanten“ Namen.

Das zuletzt aufgelistete Kriterium wurde als erfüllt betrachtet, sofern der Name aus einer Mindestanzahl von Tokens *tokenThreshold* besteht. Die Lokation „University of Cambridge“ beispielsweise besteht aus drei Tokens und hat somit einen markanteren Namen als „York“. Zur Erfüllung des Kriteriums „Eindeutigkeit“ muss der Lokationsname nicht zwingend nur einmal in der Datenbank vorkommen, vielmehr wurde die Annahme getroffen, dass auch mehrere Lokationen mit gleichem Namen, die alle sehr nah beieinanderliegen, als Anker dienen können, da sie auf einen

72 Der Terminus „Anchor Locations“ wird in einer Reihe weiterer Arbeiten verwendet, beispielsweise in Rauch et al. (2003) oder Lieberman et al. (2010), die Definition in dieser Arbeit ist jedoch eine eigenständige.

eindeutigen Punkt zeigen. Die maximale Distanz, bis zu der Lokationen mit gleichem Namen als identisch angsehen werden wird durch den Schwellwertparameter *sameDistanceThreshold* festgelegt. largestDistance(g) ermittelt dazu aus der Menge von Lokationen mit identischem Namen g unter Betrachtung jedes Lokationspaares die maximale Distanz. Der Ablauf der Anker-Bestimmung ist in Abbildung 3.4 dargestellt.

```
funct getAnchors(L, minAnchorPopulation, maxDistanceSame, minPopulation, minTokensUnique) ≡
  Anchors := {}
  for l in L do                                          // Continents, countries, big locations.
    if type(l) ∈ {CONTINENT, COUNTRY} ∨ population(l) ≥ minAnchorPopulation
      then Anchors ← l; fi
  end
  for g in groupByName(L) do                             // Group locations by names.
    if largestDistance(g) ≤ maxDistanceSame              // Single location or a cluster.
      then
          l := getBiggestLocation(g);
          if population(l) ≥ minPopulation ∨ |tokenize(name(l))| ≥ minTokensUnique
            then Anchors ← l; fi                         // Found unique location.
    fi
  end
  return Anchors.
```

Abbildung 3.4: Pseudocode für Anker-Bestimmung

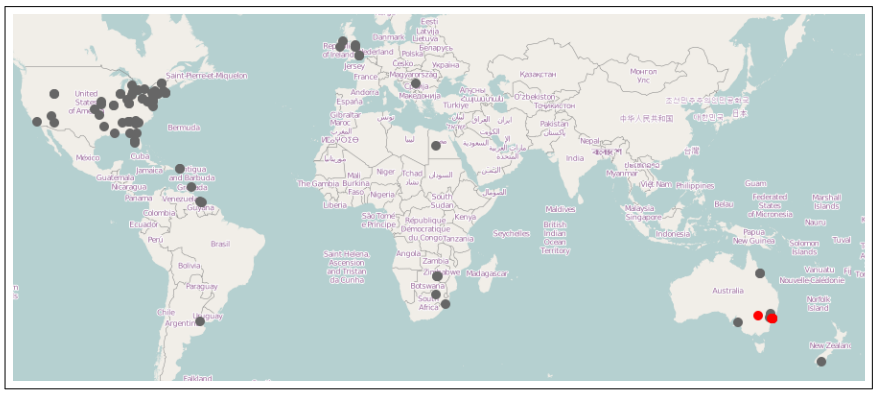


Abbildung 3.5: Beispiel für Anker-Methode; die roten Punkte sind Anker-Lokationen, die grauen noch nicht disambiguierte Lokationskandidaten für „Hyde Park“ (Kartenmaterial von OpenStreetMap, 2013).

Abbildung 3.5 zeigt ein Beispiel für einen Text, in dem der Lokationskandidat „Hyde Park“ vorkommt. Der Gazetteer enthält hierfür 83 mögliche Lokationen. Als Ankerlokationen können aus dem Text die rot abgebildeten Punkte, die den Orten „Darling Harbour“, „Parramatta“, „New South Wales“ und „Sydney“ entsprechen, extrahiert werden. Im abschließenden Disambiguierungsschritt wird dieses

Wissen genutzt, um bei mehrdeutigen Interpretationen näher bei den Ankerlokationen liegende Kandidaten zu bevorzugen. Im beschriebenen Fall kann aus den 83 Kandidaten somit der Hyde Park in Sydney statt in London ausgewählt werden.

Lasso-Methode Falls durch die oben beschriebene Anker-Methode keine Referenzlokationen bestimmt werden konnten, wird eine schrittweise räumliche Konvergenz vorgenommen. Vergleichbar mit einem Lasso, welches kontinuierlich zusammengezogen wird, wird hierbei kontinuierlich die am weitesten entferne Lokation vom Mittelpunkt der Kandidatenmenge (siehe Abschnitt 2.1.3) entfernt. Die zugrundeliegende Idee stammt von Smith und Crane (2001), allerdings stoppt der in Abbildung 3.6 angegebene Algorithmus, sobald die maximale Distanz zwischen jedem Lokationspaar in der verbleibenden Menge unter dem Schwellwert *lassoDistanceThreshold* liegt.

```
funct getLasso(L, maxLassoDistance) ≡
  Lasso ← L
  while |Lasso| > 1 do
      mostDistantLocation := null;
      highestDistance := 0;
      for l in Lasso do                                  // Get most outlying location in set.
          distance := distance(midpoint(Lasso), l);                    // siehe Abschnitt 2.1.3
          if distance > highestDistance then
                                          mostDistantLocation := l;
                                          highestDistance := distance;
          fi
      end
      if highestDistance < maxLassoDistance              // Below maximum distance, stop.
        then break; fi
      Lasso := Lasso ∖ mostDistantLocation;              // Remove most outlying location.
  end
  if |groupByNames(Lasso)| ≤ 1                           // Check plausibility of lasso set.
    then return ∅;
     else return Lasso; fi.
```

Abbildung 3.6: Pseudocode für Lasso-Bestimmung

Um das Risiko einer Konvergenz in die falsche Richtung zu vermindern, werden die auf diesem Wege bestimmten Lokationen nur dann verwendet, wenn zumindest zwei Lokationen mit unterschiedlichem Namen in der endgültigen Ergebnismenge vorhanden sind. Hiermit ist eine größere Evidenz als bei nur einer übrigen Lokation gewährleistet.

Der Satz „We went from Salamanca to Cortazar via Villagran“ beispielsweise enthält keine repräsentativen Anker-Lokationen und jeweils mehrere mögliche Interpretationen für die Orte „Salamanca“, „Cortazar“ und „Villagran“. Unter der Annahme, dass in einem Text genannte Orte vergleichsweise nah beieinander liegen, kann durch eine schrittweise Eliminierung weit außerhalb liegender

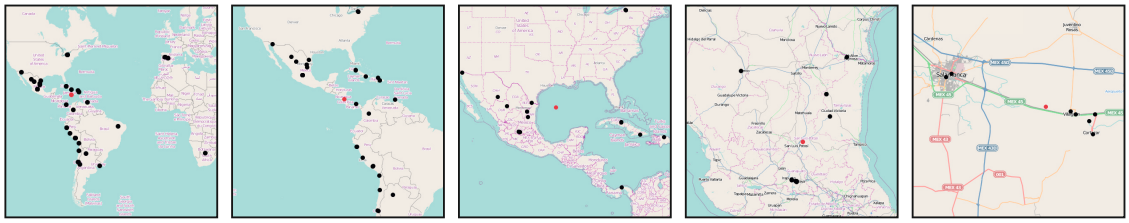


Abbildung 3.7: Ablauf der Lasso-Methode für den Satz „We went from Salamanca to Cortazar via Villagran“ mit schrittweiser Eliminierung der außen liegenden Lokationen (schwarze Punkte), der geographische Mittelpunkt ist jeweils als roter Punkt dargestellt (Kartenmaterial von OpenStreetMap, 2013).

Lokationskandidaten auf die vermeintlich korrekten Orte geschlossen werden. Der Ablauf ist in Abbildung 3.7 dargestellt.

Disambiguierung Der komplette Disambiguierungsablauf ist in Abbildung 3.8 dargestellt. Zunächst werden hier jene Annotationen eliminiert, die während der Vorverarbeitung mit dem Typ `PERSON` vorklassifiziert wurden und die keinen Lokationskandidaten vom Typ `CONTINENT` oder `COUNTRY` enthalten oder bei denen die Bevölkerungsanzahl sämtlicher Lokationskandidaten unter dem Parameter *unlikelyPopulation* liegt.

Die Lokationen, welche in den zwei vorangehenden Schritten extrahiert wurden, dienen als Referenzpunkte, um die verbleibenden, noch nicht disambiguierten Lokationen zu filtern. Konnten weder durch die Anker- noch die Lasso-Methode Bezugspunkte identifiziert werden, wird ersatzweise auf die bevölkerungsmäßig größte Lokation innerhalb der Kandidatenmenge zurückgegriffen. Anschließend werden zwei Kriterien angewendet: Räumliche Entfernungen und hierarchische Beziehungen zwischen Bezugspunkten und nicht disambiguierten Kandidaten. Für jede Annotation a aus A werden einerseits Lokationskandidaten in die Vorauswahl *Selection* genommen, deren Distanz zu einer Anker-Lokation unter dem Schwellwert *maxAnchorDistance* liegt, also beispielsweise die Stadt Glashütte, deren Distanz 22,38 km von Dresden beträgt. Andererseits werden innerhalb der Hierarchie Kinder oder Nachfahren einer Ankerlokation vom Typ `CITY`, `UNIT` oder `COUNTRY`, die einen *minPopulation*-Schwellwert überschreiten in die Vorauswahl genommen, also beispielsweise der Münchner Stadtteil Maxvorstadt, sofern München als Ankerlokation erkannt wurde.

Sämtliche Konfigurationsparameter der Heuristik sind in Tabelle 3.8 dargestellt. Diese wurden im Rahmen der Entwicklung zunächst auf Standardwerte festgelegt, die ebenfalls in der Tabelle dargestellt sind. Ob die hier festgelegten Parameter tatsächlich optimal sind, wird nachfolgend im Rahmen der Optimierung untersucht.

```
funct disambiguate(
  A, L                                                    // Set of annotations, set of locations.
  minPopulationUnlikely, minAnchorPopulation, maxDistanceSame, minPopulation,
  minTokensUnique, maxLassoDistance, maxAnchorDistance) ≡
  for a in A do                                           // Remove unlikely candidates.
    CurrentLocations := getLocations(L, a);               // Get locations L(a) for annotation a.
    biggestLocation := getBiggestLocation(CurrentLocations);
    if getClassification(a) = PERSON ∧ type(biggestLocation) ∉ {COUNTRY, CONTINENT} ∧
      population(biggestLocation) ≤ minPopulationUnlikely
      then L := L ∖ CurrentLocations;
    fi
  end
  ReferenceLocations := getAnchors(L);                    // Get anchors (Abb. 3.4).
  if |ReferenceLocations| = 0
    then ReferenceLocations := getLasso(L); fi            // Get lasso (Abb. 3.6).
  if |ReferenceLocations| = 0
    then ReferenceLocations := getBiggestLocation(L); fi  // Fallback: take biggest location.
  D := {};                                                // Result: the disambiguated annotations.
  for a in A do
    CurrentLocations := getLocations(L, a);
    References := ReferenceLocations ∖ CurrentLocations;  // Reference without current locations.
    Selection := {};
    for l in CurrentLocations do
      if l ∈ ReferenceLocations
        then Selection ← l                                // Location already in anchors or lasso.
        else for anchor in References do
               if distance(l, anchor) ≤ maxAnchorDistance
                 then Selection ← l; break; fi            // Close to anchor.
               if type(anchor) ∈ {CITY, UNIT, COUNTRY} ∧
                 isDescendant(l, anchor) ∧ population(l) ≥ minPopulation
                 then Selection ← l; break; fi            // Descendant of anchor.
             end
      fi
    end
    if |Selection| > 0                                    // Found disambiguation.
      then D ← {a, getBiggestLocation(Selection)}; fi     // Add {annotation, location} to result.
  end
  return D.
```

Abbildung 3.8: Pseudocode für heuristische Lokationsextraktion und -disambiguierung

Parameter	Beschreibung	Standard
minPopulationUnlikely	Minimale Einwohneranzahl, unter der als „unwahrscheinlich“ klassifizierte Lokationen ausgefiltert werden.	100.000
minAnchorPopulation	Minimale Einwohnerzahl für eine Anker-Lokation.	1.000.000
maxDistanceSame	Maximale Distanz zweier Lokationen gleichen Namens, bis zu der diese als gleiche Lokation betrachtet werden	50 km
minPopulation	Minimale Einwohneranzahl für Lokation, sodass diese als Nachfahre eines Ankers disambiguiert wird	5.000
minTokensUnique	Mindestanzahl von Tokens, ab der ein Lokationsname als „markant“ und somit als Anker berücksichtigt wird	2
maxLassoDistance	Maximale Distanz, bei deren Unterschreiten die Lasso-Konvergenz gestoppt wird	100 km
maxAnchorDistance	Maximale Distanz zwischen Anker-Lokation und potentiellem Lokationskandidaten	100 km

Tabelle 3.8: Parameter für heuristische Disambiguierung und die während der Entwicklung festgelegten Standardwerte

3.9 MACHINE-LEARNING-BASIERTE ERKENNUNG UND DISAMBIGUIERUNG

Die Erfahrungen aus dem heuristischen Ansatz zeigen, dass die Aufnahme weiterer Regeln zunehmend kompliziert wird und mit dem Risiko eines Overfittings einhergeht (siehe Abschnitt 2.3). Dieser Abschnitt stellt deshalb eine zweites Verfahren für die Erkennung und Disambiguierung von Toponymen vor. Die in Abschnitt 3.8 vorgestellten Heuristiken bilden die Grundlage für ein Feature Engineering, bei dem eine Reihe deskriptiver Attribute beschrieben werden, die anschließend dazu dienen, einen Klassifikator zu trainieren und diesen für die Erkennung und Disambiguierung zu nutzen.

Ein exemplarisches binäres Feature in diesem Zusammenhang wäre beispielsweise die Eigenschaft „Lokationskandidat ist Kontinent oder Land“. Für den Fall, dass innerhalb der verwendeten Daten solche Annotationen, die den Namen eines Kontinents oder Landes tragen, tatsächlich immer die korrekt disambiguierten Lokationen darstellen, wäre dieses Kriterium ein „wertvolles“ Feature für die Klassifikation. Um eine gute Vorhersagegenauigkeit des Klassifikators zu erreichen, müssen jedoch nicht nur „wertvolle“ Features verwendet werden. Guyon und Elisseeff (2003) beschreiben, dass auch für sich isoliert „schwache“ Features in Kombination mit anderen zur Klassifikationsqualität beitragen können. Demnach diskutiert der nachfolgende Abschnitt zunächst sämtliche betrachteten Features, ohne eine Wertung vorzunehmen. Diese erfolgt unter Verwendung realer Datensets im Rahmen der in Abschnitt 3.11 durchgeführten Optimierung.

3.9.1 FEATURES ZUR KLASSIFIKATION

Die nachfolgend vorgestellten Features gliedern sich, abhängig davon, woher die Informationen extrahiert werden, in fünf Gruppen. Manche der vorgestellten Features sind parametrisierbar, was durch eine Parameterschreibweise, wie beispielsweise `uniqueIn(R)` verdeutlicht wird.

Annotationsfeatures Diese intrinsischen Features (McDonald, 1996) beziehen sich jeweils auf den Textwert der betrachteten Annotation. Sie umfassen die numerischen Features `numCharacters` und `numTokens`, welche die Anzahl an Zeichen und Tokens in der Annotation angeben, sowie ein binäres Feature, welches ausdrückt, ob es sich bei dem betrachteten Kandidaten um ein Akronym (wie beispielsweise „U. S." oder „UAE") handelt. Das Feature `caseSignature` charakterisiert die Kombination aus Groß- und Kleinbuchstaben einer Annotation (Collins, 2002). Bestimmte Lokationsnamen verfügen über charakteristische Kombinationen, wie beispielsweise „University of California" oder „Isle of Man" (Case-Signatur `Aa a Aa`), wohingegen Kombinationen wie „McDonald" oder „InterCity" (Case-Signatur `AaAa`) Anhaltspunkte für eine Nicht-Lokationen liefern.

Textfeatures Textfeatures sind extrinsische Features (McDonald, 1996), da sie jeweils unter Betrachtung des Gesamttexts des jeweiligen Dokuments extrahiert werden. Die hier verwendeten Features geben die Vorkommenshäufigkeit einer Annotation `count` und die daraus abgeleitete Frequenz `frequency` an, welche den Wert von `count` einer Annotation mit der höchsten Anzahl für `count` im Gesamtdokument normalisiert. Die zwei weiteren Features `firstOccurrenceAbs` und `firstOccurrenceRel` geben an, an welcher Stelle im Dokument ein Annotationstext das erste Mal vorkommt. Die erste Variante gibt die absolute Zeichenanzahl vom Beginn des Texts an, die zweite Variante den relativen Wert, normalisiert mit der Gesamtlänge des Texts. Mit dem Feature `partialAnnotation` wird ausgedrückt, ob die betreffende Annotation Teil einer längeren Annotation ist, die unter Verwendung des in Abschnitt 3.7 beschriebenen Case Dictionarys aufgetrennt wurde.

Korpusfeatures Diese ebenfalls extrinsischen Features werden unter Verwendung im Vorfeld erzeugter Listen und Textkorpora extrahiert. Die beiden binären Features `likelyCandidate` und `unlikelyCandidate` geben die in Abschnitt 3.7 beschriebene Klassifikation „Potentiell-Positiv" und „Potentiell-Negativ" wieder, die auf Basis von Termkontexten ermittelt werden. Das binäre Feature `stopword` gibt an, ob sich die Annotation in einer Liste mit 590 englischen Stoppwörtern befindet. Mit `containsMarker` hingegen wird angezeigt, dass die Annotation einen Token enthält, der explizit auf eine Lokation hindeutet. Die verwendete Whitelist wurde unter Verwendung der englischsprachigen Wikipedia generiert und besteht aus 24 Einträgen, die häufig innerhalb von Lokationsnamen vorkommen[73].

73 District, Township, Route, Lake, County, State, Highway, River, Mount, Island, Gmina, Municipality, Road, Mountain, Creek, Hill, City, Park, Peak, Canton, Falls, Province, Olya, Formation

Datenbankfeatures Während die vorangehend beschriebenen Annotations-, Text- und Korpusfeatures pro Annotation im Text ermittelt werden, werden die Datenbankfeatures pro Lokationskandidat ermittelt, da sie sich aus sämtlichen zur Annotation gefundenen Einträgen im Gazetteer ableiten (siehe Abschnitt 3.5). Das nominale Feature `locationType` gibt den Typ der Lokation gemäß der in Tabelle 3.5 aufgeführten Klassen an. Mit den Features `country`, `continent` und `city` wird die Klasse zusätzlich noch als binäres Feature angegeben. Die Einwohnerzahl des jeweiligen Lokationskandidaten wird angegeben als absoluter Wert `population`, als dezimale Größenordnung `populationMagnitude` und als `populationNormalized`, welcher die Einwohnerzahl mit der höchsten im Dokument vorkommenden normalisiert.

Das numerische Feature `indexScore` zielt auf das Problem der Geo/Non-Geo-Ambiguität und folgt dem Gedankengang von Amitay et al. (2004). Die Einwohnerzahl der jeweiligen Lokation wird hier ins Verhältnis zur Vorkommenshäufigkeit des Annotationstexts innerhalb eines großen Web-Korpus gesetzt. Eine vergleichsweise geringe Einwohneranzahl bei vielen Vorkommen im Index deutet darauf hin, dass der entsprechende Term vielfach als Nicht-Lokation auftaucht. Das Feature kann prinzipiell mit jeder Suchmaschine, wie beispielsweise Bing gewonnen werden. Im Rahmen der durchgeführten Experimente wurde auf den englischen ClueWeb09[74]-Datensatz („TREC Category B"-Subset) zurückgegriffen, welcher aus 50 Millionen Webseiten besteht.

Ein weiteres numerisches Feature `hierarchyDepth` modelliert die hierarchische Tiefe der Lokation, also die Anzahl der Vorfahren in der Hierarchie (die Lokation „Erde" hat somit eine Hierarchietiefe von null, wohingegen sehr spezifische, kleine Lokationen eine vergleichsweise hohe Hierarchietiefe haben). Das binäre Feature `leaf` gibt an, dass der Lokationskandidat keine Elternlokation mit gleichem primären oder Alternativnamen besitzt. Dieses Feature zielt darauf ab, bei der Klassifikation möglichst spezifische Lokationen zu selektieren. Beispielsweise enthält die Datenbank für die Bezeichung „München" sowohl Einträge vom Typ `CITY`, als auch übergeordnete Verwaltungseinheiten vom Typ `UNIT`, die „München" als Alternativnamen ausweisen.

Ferner wird als numerisches Feature `nameAmbiguity` die Eindeutigkeit einer Lokation als Bruchteil von eins ($1 / |\mathit{locationsWithSameName}|$) angegeben, da für Lokationen, unter deren Namen sich mehrere Einträge im Gazetteer finden die Wahrscheinlichkeit einer falschen Auswahl steigt. Das numerische Feature `nameDiversity` auf der anderen Seite gibt an, wie viele Alternativnamen für den aktuellen Lokationskandidaten existieren ($1 / |\mathit{namesForLocation}|$). Die Annahme ist, dass relevante Orte unter mehr Alternativnamen bekannt sind als weniger relevante. Die Features `nameAmbiguity` und `nameDiversity` wurden bereits von Lieberman und Samet (2012) verwendet.

Das Feature `geoDiversity` drückt aus, wie stark Lokationskandidaten mit dem gleichen Namen räumlich verteilt sind. Der Gedanke hinter diesem Feature ist, dass die möglichen Fehlerdistanzen bei einer Fehlklassifikation steigen, je weiter die jeweiligen Kandidaten auseinanderliegen. Hingegen ist das Risiko einer Fehlklassifikation bei nah beieinanderliegenden Orten geringer. Die gleiche

74 `http://lemurproject.org/clueweb09/`

Motivation steckt hinter den binären Features `unique` und `uniqueAndLong`. Diese geben an, ob die maximale Distanz zwischen sämtlichen Paaren von Lokationskandidaten für eine Annotation unter einem Schwellwert von 50 km liegt. Für `uniqueAndLong` wird zusätzlich gefordert, dass der Text der zugehörigen Annotation aus mindestens drei Tokens besteht (beispielsweise „University of Cambridge“).

Kombinierte Text- und Datenbankfeatures Die letzte Gruppe von Features wird sowohl aus dem Gesamttext, insbesondere anderen vorkommenden Lokationskandidaten, als auch aus dem Gazetteer bestimmt. Die fünf binären Features `contains(p|a|s|c|d)` geben an, ob sich für den aktuell betrachteten Lokationskandidaten Eltern-/Vorfahrens-/Geschwister-/Kind-/Nachfahrenslokationen aus der Hierarchie befinden. Die vier numerischen Features `num(a|s|c|d)` zählen zustätzlich zu dem oben beschriebenen Feature die vorkommenden Vorfahrens-/Geschwister-/Kind-/Nachfahrenslokationen. Beide Features zielen unter Verwendung der Hierarchie darauf ab, Kohäsionen zwischen einzelnen Lokationen im Text aufzudecken, wie sie beispielsweise bei explizit disambiguierten Phrasen oder Aufzählungen vorkommen. Für die Nennung der Stadtteile „Reinhardtsgrimma and Dittersdorf“ innerhalb eines Text, die beide zu Glashütte in Sachsen gehören wäre beispielsweise das Feature `contains(s)` wahr, für „Houston is a city in Texas“ wäre für den Lokationskandidaten Houston in Texas das Feature `contains(p)` wahr. Von Lieberman und Samet (2012) wurde im Rahmen der „Sibling Features“ eine ähnliche Idee umgesetzt, die sich jedoch ausschließlich auf Geschwisterrelationen bezog.

Die Ausprägungen des binären Features `containedIn(T)` ermittelt wie `contains(a)`, ob der aktuelle Lokationskandidat Nachfahre eines anderen aus dem Text extrahierten Lokationskandidaten ist, beschränkt dies jedoch auf explizit angegebene Lokationstypen für potentielle Vorfahrenslokationen. Das Feature wird für die folgenden Ausprägungen von `T` ermittelt: `CONTINENT`, `COUNTRY`, `UNIT`, `COUNTRY` $\vee$ `UNIT` und `CONTINENT` $\vee$ `COUNTRY` $\vee$ `UNIT`.

`primaryName` gibt als binäres Feature an, ob der Annotationstext dem primären Namen der Lokation entspricht. Mit dem Feature `alternativeMention` wird angegeben, ob der jeweilige Lokationskandidat im Text anhand mehrerer alternativer Namen genannt wird (beispielsweise „New York“ und „Big Apple“), was ein starker Hinweis dafür ist, dass eben jene Lokation gemeint ist und nicht eine andere gleichen Names, welche nicht über den Alternativnamen verfügt.

Die weiteren Features basieren auf räumlichen Distanzen (siehe Abschnitt 2.1.2) und nutzen den Umstand, dass in Texten üblicherweise viele nahe beieinander liegende Lokationen vorkommen. Aufgrund dessen, dass sie parametrisierbar sind und aus dem Kandidatenkontext abgeleitet werden, werden diese nachfolgend als „parametrisierbare Kontextfeatures“ bezeichnet. Das numerische Feature `numLocIn(R)` gibt die Anzahl im Text vorkommender Lokationskandidaten an, die sich in einer maximalen Entfernung `R` zu dem gerade betrachteten Lokationskandidaten befinden. Das Feature `distLoc(P,S)` gibt die minimale Distanz des betrachteten zu sämtlichen weiteren Lokationskandidaten im Text mit einer Mindesteinwohnerzahl von `P` an. Der boolesche Parameter `S` gibt dabei

an, ob der aktuell betrachtete Lokationskandidat selbst mitgezählt werden soll oder nicht. Mit dem numerischen `populationIn(R,S)` werden die Einwohnerzahlen sämtlicher Lokationskandidaten, die eine maximale Distanz von `R` zum betrachteten Kandidaten haben, aufsummiert. Mit dem Parameter `S` wird auch hier angegeben, ob die betrachtete Lokation selbst mitbetrachtet wird. Das binäre Feature `locSentence(R)` drückt aus, ob innerhalb für eine andere Annotation im Satz ein Lokationskandidat mit einer maximalen Distanz von `R` zum aktuellen Kandidaten vorkommt. Mit dem binären Feature `uniqueIn(R)` wird angezeigt, ob innerhalb eines Radius `R` um den aktuellen Kandidaten eine als eindeutig ermittelte Lokation vorkommt (siehe oben unter „Datenbankfeatures“ für eine Beschreibung des Kriteriums). Das letzte Feature in diesem Zusammenhang ist `hasLoc(R,P,S)`. Dieses binäre Feature verzeichnet, ob im Text ein Lokationskandidat mit mindestens `P` Einwohnern und einer maximalen Entfernung von `R` auftaucht. Für den Parameter `S` gilt wie oben: Ist dieser wahr, wird die betrachtete Lokation selbst mitgezählt.

Zusammenfassung Der vorangehende Abschnitt zeigte die Ergebnisse des Feature Engineerings, bei dem 46 Features vorgestellt wurden. Einen Überblick und Einordnung der präsentierten Features findet sich in Tabelle 3.9. Die Features `nameAmbiguity`, `nameDiversity` und `population` wurden bereits von Lieberman und Samet (2012) genutzt. Hierarchische Geschwisterrelationen wurden von Lieberman und Samet (2012) vorgestellt, die hier vorgestellten hierarchischen Features umfassen jedoch deutlich mehr Relationen in unterschiedlichen Ausprägungen.

Die restlichen der hier gezeigten Features sind neu und wurden im Rahmen einer Machine-Learning-basierten Lokationsextraktion in keiner anderen verwandten Arbeit verwendet. Besonders herauszustellen sind hier die parametrisierbaren Kontextfeatures, von denen sechs vorgestellt wurden, wie beispielsweise `distLoc(P,S)`. Durch die Verwendung verschiedener Schwellwerte für die einzelnen Parameter kann hier eine Vielzahl von Features extrahiert und miteinander kombiniert werden. Während Lieberman und Samet (2012) ihr kontextbasiertes „Proximity Feature“ unter Betrachtung jedes ermittelten Lokationskandidaten berechnen, kann durch die hier vorgestellte Parametrisierung der Kontext auf eine kleinere Teilmenge von Kandidaten beschränkt werden, womit analog zu der in Abschnitt 3.8 Anker-Methode eine höhere Precision im Kontext gewährleistet wird. Im Rahmen der nachfolgend durchgeführten Optimierungen werden die parametrisierbaren Kontextfeatures in unterschiedlichen Konfigurationen evaluiert.

3.9.2 TRAINING DES KLASSIFIKATORS

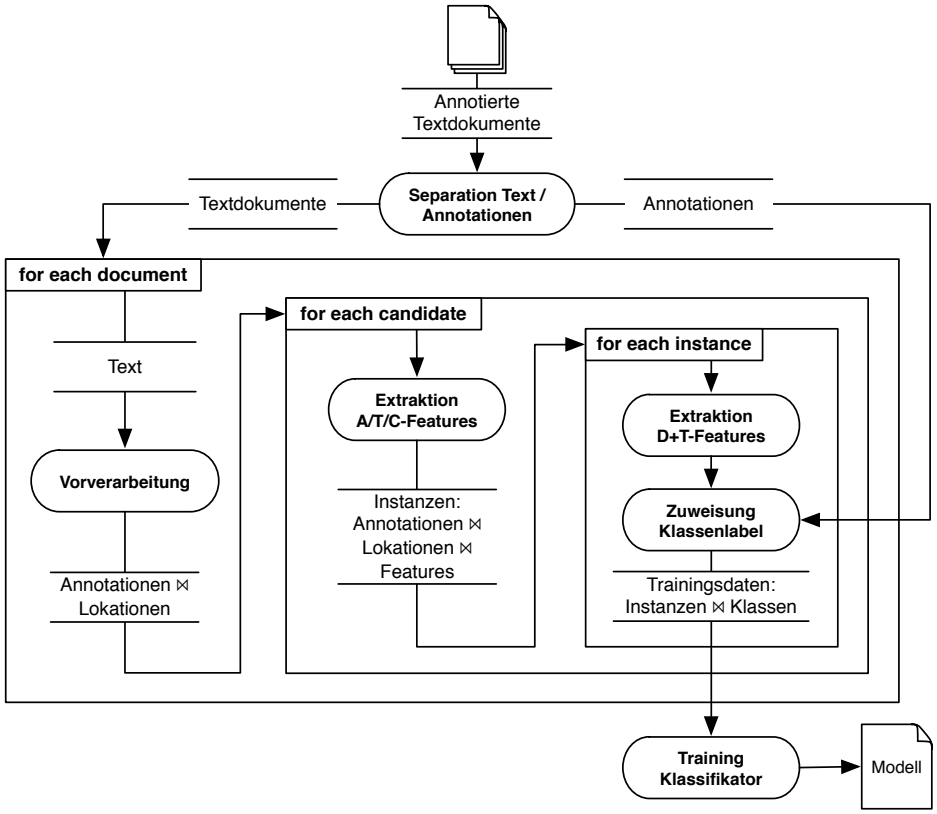
Der Klassifikator wird mittels annotierter Texte trainiert, in denen sämtliche Toponyme annotiert und mit Koordinaten versehen sind (siehe Abschnitt 3.4.2). Abbildung 3.9 zeigt den Ablauf des Trainingsvorgangs. Zunächst werden die annotierten Trainingsdokumente jeweils in ihren ursprünglichen Text und die enthaltenen Annotationen getrennt, die später dazu dienen, die positiven Trainingsbeispiele zu identifizieren. Für jedes Dokument erfolgt anschließend die Extraktion und Filterung potentieller Toponymkandidaten entsprechend der in Abschnitt 3.7 beschriebenen

Bezeichnung	Typ	Beschreibung
Annotationsfeatures		
`numCharacters`	num.	Zeichenanzahl der Annotation
`numTokens`	num.	Tokenanzahl der Annotation
`acronym`	bin.	Annotation ist ein Akronym
`caseSignature`	nom.	Upper-/Lowercase-Signatur der Annotation
Textfeatures		
`count`	num.	Vorkommenshäufigkeit der Annotation
`frequency`	num.	`count` normalisiert mit höchstem Wert
`firstOccurrenceAbs`	num.	Zeichenoffset des ersten Auftretens
`firstOccurrenceRel`	num.	`firstOccurrenceAbs` normalisiert mit Textlänge
`partialAnnotation`	bin.	Annotation ist Teil einer größeren Annotation
Korpusfeatures		
`unlikelyCandidate`	bin.	Term-Kontext der Annotation deutet auf Nicht-Lokation hin
`likelyCandidate`	bin.	Term-Kontext der Annotation deutet auf Lokation hin
`stopword`	bin.	Annotation befindet sich auf Stoppwortliste
`containsMarker`	bin.	Annotation enthält Whitelist-Marker, z. B. „city“
Datenbankfeatures		
`locationType`	nom.	Typ der Lokation (siehe Tabelle 3.5)
`country`	bin.	Lokation ist vom Typ `COUNTRY`
`continent`	bin.	Lokation ist vom Typ `CONTINENT`
`city`	bin.	Lokation ist vom Typ `CITY`
`population`	num.	Einwohnerzahl der Lokation
`populationMagnitude`	num.	Größenordnung von `population`
`populationNorm`	num.	`population` normalisiert mit höchster Einwohnerzahl
`indexScore`	num.	Quotient aus `population` und Häufigkeit im Textindex
`hierarchyDepth`	num.	Tiefe in der Hierarchie
`leaf`	bin.	Lokation hat keine Nachfahren mit gleichem Namen
`nameAmbiguity`	num.	Eindeutigkeit der Lokation bezüglich ihres Namens
`nameDiversity`	num.	Vielfältigkeit von Alternativnamen für Lokation
`geoDiversity`	num.	Räumliche Streuung gleichnamiger Lokationen
`unique`	bin.	Lokation ist im Hinblick auf ihre Position eindeutig
`uniqueAndLong`	bin.	`unique` und Name mit mindestens drei Tokens
Text- und Datenbankfeatures		
`contains(p|a|s|c|d)`	bin.	Text enthält Eltern-/Vorfahrens-/Geschwister-/Kind-/Nachfahrenskandidat
`num(a|s|c|d)`	bin.	Anzahl von Vorfahrens-/Geschwister-/Kind-/Nachfahrenskandidaten im Text
`in(T)`	bin.	Lokation ist Nachfahre einer im Text vorkommenden Lokation vom Typ `T`
`primaryName`	bin.	Annotation ist primärer Name im Gazetteer
`alternativeMention`	bin.	Lokation wird mit mehreren Namen genannt
`numLocIn(R)`	num.	Anzahl von Kandidaten mit maximaler Distanz `R`
`distLoc(P,S)`	num.	Minimale Entfernung zu anderen Kandidaten mit mindestens `P` Einwohnern; `S` gibt an, ob die betrachtete Lokation selbst mitgezählt wird
`populationIn(R,S)`	num.	Summierte Einwohnerzahlen im Text vorkommender Lokationen mit einer Entfernung von höchstens `R`; für `S` siehe `distLoc(P,S)`
`locSentence(R)`	bin.	Satz enthält Lokation mit maximaler Distanz `R`
`uniqueLocIn(R)`	bin.	Text enthält eindeutige Lokation mit maximaler Distanz `R`
`hasLoc(R,P,S)`	bin.	Text enthält Lokation mit mindestens `P` Einwohnern und maximaler Distanz `R`; für `S` siehe `distLoc(P,S)`

Abkürzungen: bin. = binär, nom. = nominal, num. = numerisch

Tabelle 3.9: Features für Machine-Learning-basierte Erkennung und Disambiguierung

Schritte. Für jede Annotation werden die Annotations-, Text- und Korpusfeatures extrahiert (siehe Tabelle 3.9), für jede Kombination aus Annotation und Lokationskandidat aus dem Gazetteer werden anschließend die datenbankspezifischen Features extrahiert. Annotationen, für welche keine Entsprechungen im Gazetteer gefunden wurden, werden hier verworfen.



Legende: A/C/D/T Features = Annotations-/Korpus-/Datenbank-/Textfeatures,
⋈ = Join anhand Annotations-/Lokationsname

Abbildung 3.9: Trainingsablauf des Machine-Learning-basierten Ansatzes

Aus dieser Menge der Lokationskandidaten mitsamt ihrer Features werden im nächsten Schritt Instanzen erzeugt, wofür die manuell zugewiesenen Trainingsannotationen genutzt werden. Für jede Kombination aus manuell erzeugter Trainingsannotation und automatisch extrahierter Instanz werden die nachfolgenden Kriterien geprüft. Sind alle erfüllt, dient die Instanz als positives, sonst als negatives Beispiel. Dieses „Fuzzy"-Zuordnungsverfahren kommt deshalb zum Einsatz, da die Datenbestände zwischen Trainingsdaten und verwendetem Gazetteer nicht zwangsläufig identisch sind.

1. Der Name oder Alternativname stimmt mit der Trainingsannotation überein.
2. Der Typ stimmt mit dem Typ der Trainingsannotation überein.
3. Die Distanz (siehe Abschnitt 2.1.1) zur Trainingslokation beträgt maximal 50 km.

Instanzen, die keine Entsprechung innerhalb der Menge der Trainingsannotation haben, werden ebenfalls als negative Beispiele extrahiert. Dies ist der Fall, wenn irrtümlich Lokationen extrahiert werden (Geo/Non-Geo-Ambiguitäten). Aus den Trainingsinstanzen wird anschließend ein Klassifikationsmodell trainiert. Denkbar ist theoretisch die Verwendung beliebiger Klassifikationsverfahren aus dem überwachten maschinellen Lernen mit der Unterstützung für nominale und numerische Eingabefeatures. Verwendet werden hier aufgrund der in Abschnitt 2.3.3 beschriebenen Vorteile gegenüber anderen Klassifikatoren Random Forests.

3.9.3 ERKENNUNG UND DISAMBIGUIERUNG MITTELS KLASSIFIKATION

Das Modell zur Toponymerkennung und -disambiguierung, welches gemäß der in Abschnitt 3.9.2 beschriebenen Schritte aus Trainingsdaten erzeugt wurde, wird wie in Formel 3.1 dargestellt zur Klassifikation verwendet.

$$\text{disambiguate}(L(a)) = \begin{cases} \arg\max\limits_{l \in L(a)} \text{classify}(l, +) & \text{wenn } \text{classify}(l, +) \geq \textit{confidenceThreshold} \\ \varnothing & \text{sonst} \end{cases} \tag{3.1}$$

Die meisten Verarbeitungsschritte, wie die Vorverarbeitung und Featureextraktion, sind mit dem Trainingsablauf (siehe Abbildung 3.9) identisch. Der Klassifikator nutzt dann das vorhandene Modell, um sämtlichen erzeugten Instanzen einen Wahrscheinlichkeitswert im Intervall $[0, 1]$ für die positive Klasse zuzuordnen. Wie in Abschnitt 3.5 erläutert, existieren pro Annotation a im Allgemeinen mehrere Lokationskandidaten $L(a)$. Selektiert wird hier pro Annotation jener Kandidat, der mit der höchsten positiven Wahrscheinlichkeit $\text{classify}(l, +)$ klassifiziert wurde, sofern der höchste Wahrscheinlichkeitswert mindestenes einen festgelegten Schwellwert *confidenceThreshold* annimmt. Auf diese Weise kann jener Kandidat mit der höchsten Konfidenz ausgewählt oder sämtliche Kandidaten verworfen werden (in Formel 3.1 mittels $\varnothing$ dargestellt), weil diese nach Klassifikation als Nicht-Lokationen angesehen werden.

Es ist offensichtlich, dass durch die Wahl des Parameters *confidenceThreshold* = $[0, 1]$ direkt Einfluss auf den Precision-Recall-Tradeoff (siehe Abschnitt 2.4) genommen werden kann. Hohe Werte für *confidenceThreshold* führen dazu, dass nur wenige Lokationskandidaten, die mit hoher Konfidenz klassifiziert wurden, akzeptiert werden, womit eine hohe Precision erzielt werden kann. Bei einem geringen *confidenceThreshold* hingegen werden auch Kandidaten mit geringer Konfidenz akzeptiert, was dem Recall zugute kommt. Eine Analyse der Schwellwerte und die Auswirkungen auf die Klassifikationsergebnisse erfolgt im Rahmen der Optimierungen in Abschnitt 3.11.2.

3.10 NACHVERARBEITUNG

In der Nachverarbeitungsphase, die sowohl für die Heuristik als auch die Machine-Learning-basierte Methode angewendet wird, werden Lokationen der Typen `STREET`, `STREETNR` und `ZIP` annotiert,

die nicht in der Gazetteer-Datenbank vorhanden sind. Aufgrund dessen, dass im News-Bereich Adressdaten nur in wenigen Ausnahmefällen auftauchen, wurde eine Adressextraktion im Rahmen der Arbeit nur geringe Priorität eingeräumt.

Die umgesetzte Extraktion erfolgt rein regelbasiert. So werden zunächst Straßennamen durch eine Reihe von Präfix- und Suffixregeln (wie „*street“, „*road“, „rue *“ etc.) erkannt. Davon ausgehend werden Zahlen, welche unmittelbar vor oder nach den erkannten Straßennamen stehen als Hausnummern annotiert. Ähnlich wird mit Postleitzahlen verfahren, welche sich vor/nach Entitäten, die als `CITY` annotiert wurden, befinden. Um auch Adressdaten in die oben beschriebenen Disambiguierungsabläufe einzubinden, bieten sich zukünftige Erweiterungen des Gazetteers an, beispielsweise mit dem Datenbestand von OpenStreetMap[75].

Als abschließender Schritt der Nachverarbeitung werden mittels regulärer Ausdrücke explizit vorkommende Koordinaten extrahiert, wie beispielsweise in „Mast Hill (68° 11′ S 67° 0′ W) is a hill 14 metres (46 ft) high at the western end of Stonington Island […]“. Der verwendete reguläre Ausdruck deckt Nennungen in Dezimal- und DMS-Schreibweise ab (siehe Abschnitt 2.1.1).

3.11 REALISIERUNG UND OPTIMIERUNG

Die beiden hier vorgestellten Strategien zur Lokationsextraktion wurden innerhalb des Palladian-Toolkits Urbansky, Muthmann, Katz und Reichert (2012) implementiert. Nachfolgend wird anhand des in Abschnitt 3.4.2 vorgestellten TUD-Loc-2013-Datensets und teilweise unter Verwendung der Datensets Clust und LGL (siehe Abschnitt 3.4.1) eine Feinabstimmung der beiden Verfahren vorgenommen.

Der heuristische Ansatz kann anhand der sieben in Tabelle 3.8 dargestellten Parameter optimiert werden. Für den Machine-Learning-basierten Ansatz wurde als Ergebnis der Feature Engineerings in Tabelle 3.9 eine umfangreiche Auswahl potentieller Features für die Klassifikation vorgestellt. Nachfolgend wird untersucht, welche dieser Features tatsächlich nützlich und wie viele Features für die Klassifkation benötigt werden. Die Auswertung der Ergebnisse erfolgt anhand der in Abschnitt 3.3 besprochenen Evaluierungsmethoden.

3.11.1 OPTIMIERUNG DES HEURISTISCHEN ANSATZES

Die Heuristik wurde unter Verwendung des Trainingssets von TUD-Loc-2013 entwickelt und kann mit sieben Parametern instanziiert werden. Nachfolgend wird die Auswirkung bei Variation jeweils einer der Parameter untersucht, während die restlichen Parameter auf den in Tabelle 3.8 aufgeführten Standardwerten verbleiben.

75 `http://wiki.openstreetmap.org/wiki/Planet.osm`

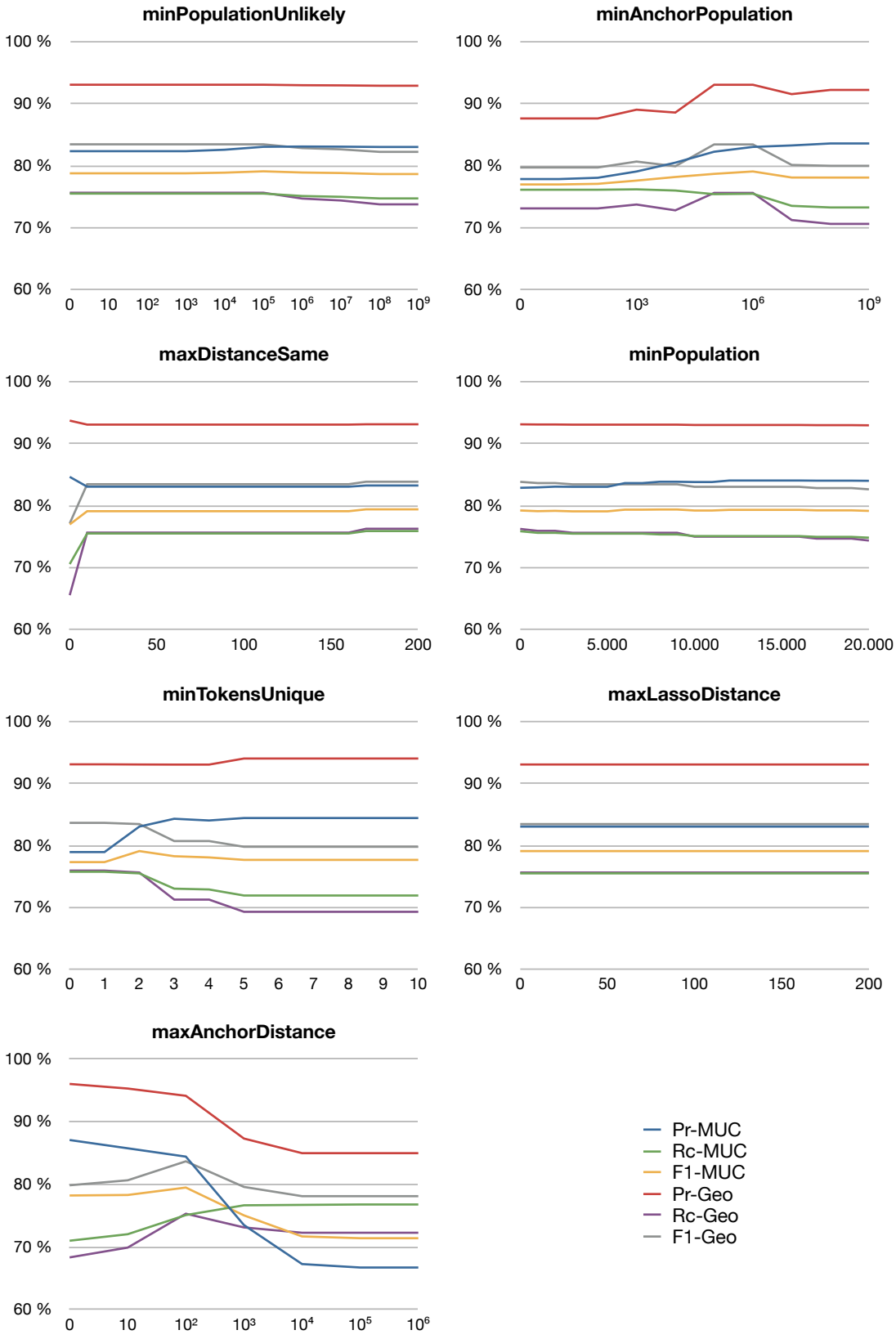


Abbildung 3.10: Einfluss der Parameter der heuristischen Erkennung und Disambiguierung unter Verwendung des Validierungssets von TUD-Loc-2013

Abbildung 3.10 zeigt die Auswirkungen von Precision, Recall und F1-Maß nach MUC- und GeoSchema unter Verwendung des Validierungssets von TUD-Loc-2013. Die Resultate zeigen, dass mit der Variation von *minAnchorPopulation* und *maxAnchorDistance* markante Einflüsse auf die Extraktion und Disambiguierung genommen werden kann. Der initial gewählte Wert von 2 für den *tokenThreshold* liefert bereits die besten Ergebnisse, höhere bzw. niedrigere Werte resultieren in schlechteren F1-Werten. Eine Änderung der restlichen Parameter hat keinen signifikanten Einfluss. Dies verdeutlicht einerseits, dass der vorgestellte Ansatz nicht auf das Datenset überangepasst ist und gut generalisiert, andererseits wird offensichtlich, dass aufwendigere Verfahren zur Parameteroptimierung nicht notwendig sind. Insgesamt erzielen die initial, unter Verwendung des Trainingssets von TUD-Loc-2013, gewählten Standardparameter bereits optimale Resultate im Hinblick auf das F1-Maß.

Eine Optimierung in Richtung höherer Precision oder höherem Recall erlaubt die *maxAnchorDistance*. Je geringer der Schwellwert zwischen Anker und potentiellen weiteren Kandidaten gewählt wird, desto höher die erzielte Precision. Durch die Erhöhung der Distanz wird ein besserer Recall erzielt, der jedoch mit einem vergleichbar starken Verlust bei der Precision einhergeht.

Es fällt auf, dass eine Veränderung des Parameters *maxLassoDistance* bei den betrachteten Daten keine Auswirkung hat. Dies begründet sich darin, dass in dem verwendeten Validierungsset für jedes Dokument die Anker-Methode angewendet werden kann. Der Werteverlauf für die einzelnen Parameter wurde ebenfalls unter Verwendung eines Validierungssets für LGL und Clust untersucht, wobei der Werteverlauf jeweils entsprechend war und die gleichen Optima erzielt wurden. Einzig bei LGL zeigt die Variation von *maxLassoDistance* minimale Auswirkungen, als Optimum können allerdings auch hier die initial gewählten 100 km für diesen Parameter beibehalten werden.

3.11.2 Optimierung des Machine-Learning-basierten Ansatzes

In Tabelle 3.9 wurden 46 Features beschrieben, die für die Machine-Learning-basierte Erkennung und Disambiguierung verwendet werden können. Sechs davon gehören der Kategorie „parametrisierbare Kontextfeatures“ an und können anhand eines oder mehrerer Parameter instanziiert werden. Dies betrifft die Features `numLocIn(R)`, `distLoc(P,S)`, `populationIn(R,S)`, `locSentence(R)`, `uniqueLocIn(R)` und `hasLoc(R,P,S)`. Der Distanzparameter `R` wurde jeweils auf die Werte 10, 50, 100 und 250 km gesetzt, der Parameter `P`, der eine Bevölkerungsanzahl angibt wurde für Werte von 1.000, 10.000, 100.000 und 1 Mio. extrahiert. Ferner wurde der Parameter `S` jeweils für die Ausprägung `true` (Berücksichtigung der Lokationskandidaten der aktuellen Annotation bei der Featureextraktion) und `false` (ohne Berücksichtigung der Lokationskandidaten der aktuellen Annotation) extrahiert. Zur Klassifikation wurde die Random-Forest-Implementierung des Decision-Tree-Klassifikators (siehe Abschnitt 2.3.3) QuickDT[76] verwendet, die erzeugten Random Forests bestanden jeweils aus zehn Decision Trees.

76 https://github.com/sanity/quickdt

Combined-Datenset Um nachfolgend möglichst generelle und datensatzunabhängige Aussagen treffen zu können, wurden das Datenset TUD-Loc-2013 mit LGL und Clust zu einem Gesamtdatenset kombiniert, welches nachfolgend mit „Combined“ bezeichnet wird. Für LGL und Clust wurde entsprechend dem Verhältnis in TUD-Loc-2013 einmalig eine zufällige 40:20:40-Aufteilung vorgenommen, somit besteht Combined aus 729 Trainings- und 365 Validierungsdokumenten.

Backward Feature Elimination Mittels Backward Feature Elimination (siehe Abschnitt 2.3.4) sollte untersucht werden, wie stark die Feature-Menge reduziert werden kann, ohne signifikate Einbußen bei der Klassifikationsqualität zu verzeichnen. Mit den 104 Features[77] wurden 5.460 Klassifikatoren mittels Combined-Trainingsset trainiert und unter Verwendung des Combined-Validierungssets getestet. Die erzielten Ergebnisse zeigten, dass eine große Menge der Features entfernt werden kann, ohne die Klassifikationsqualität zu verschlechtern. Ergebnis ist das in Abbildung 3.11 dargestellte Featureset bestehend aus 15 Features. Das Featureset enthält acht der hier neu vorgestellten parametrisierbaren Kontextfeatures. Nur zwei der hier ermittelten Features, nämlich `nameAmbiguity` und `populationNorm` wurden bereits von Lieberman und Samet (2012) verwendet.

1. `country`
2. `caseSignature`
3. `uniqueLocIn(250)`
4. `populationIn(10,true)`
5. `city`
6. `nameAmbiguity`
7. `populationIn(50,false)`
8. `populationNorm`
9. `hasLoc(100k,250,true)`
10. `numLocIn(50)`
11. `stopword`
12. `hierarchyDepth`
13. `leaf`
14. `hasLoc(100k,50,false)`
15. `hasLoc(10k,100,false)`

Abbildung 3.11: Ausgewählte Features für die Machine-Learning-basierte Disambiguierung nach Backward Feature Elimination unter Verwendung des Combined-Datensets (TUD-Loc-2013, LGL und Clust)

Information Gain Wie in Abschnitt 2.3.4 beschrieben, sagt das Ergebnis der Backward Feature Elimination nichts über den „Wert“ der einzelnen Features aus. Aus diesem Grund wurde für sämtliche der 105 extrahierten Features[78] der Information Gain (siehe Abschnitt 2.3.2) ermittelt. Beim Vergleich mit der Featuremenge aus Abbildung 3.11 fällt einerseits auf, dass die per Feature-Eliminierung ermittelten Features über das komplette Information-Gain-Spektrum verteilt sind, andererseits zeigt sich, dass die ermittelte Featuremenge sowohl starke als auch sehr schwach gerankte Features umfasst. So findet sich `populationIn(10,true)` an vierter Stelle nach Information Gain, das Feature `stopword` hingegen an vorletzter. Dies stützt die These von Guyon und Elisseeff (2003), dass auch schwache Features für die Klassifikation nützlich sein können.

77 Da sich das Feature `indexScore` in der verwendeten Implementierung aufgrund der Index-Zugriffe einen hohen Berechnungsaufwand mit sich brachte, wurde dieses initial eliminiert, um es aus der ermittelten Feature-Menge auszuschließen.

78 Im Gegensatz zu der vorangehenden Untersuchung wird hier das Feature `indexScore` mitbetrachtet.

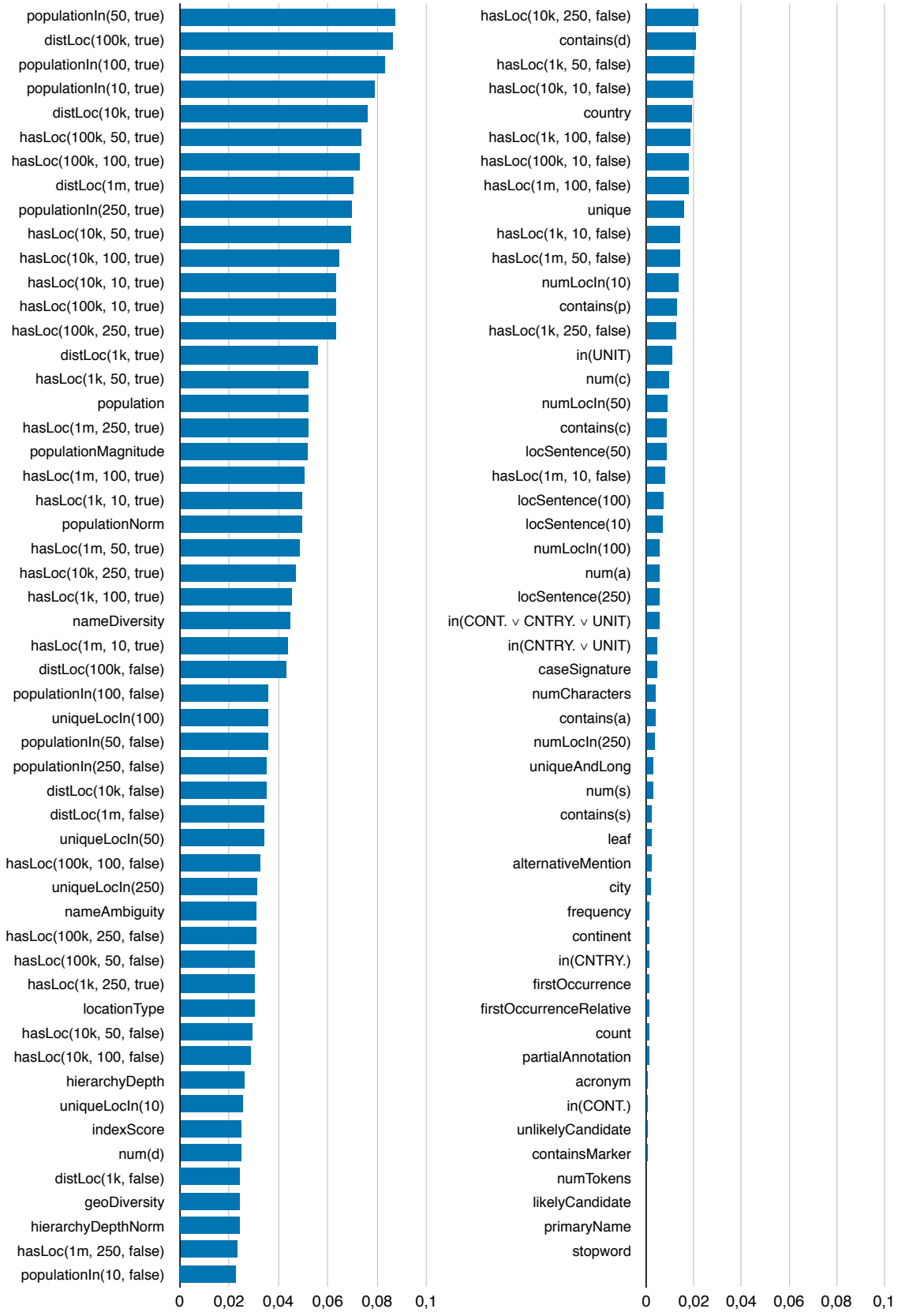


Abbildung 3.12: Information Gain der vorgestellten Features ermittelt auf dem Combined-Datenset

Ferner wird deutlich, dass die in Abschnitt 3.9 vorgestellten parametrisierbaren Kontextfeatures nach Information Gain durchweg hoch gerankt sind. So sind die ersten 16 Features des Rankings durchweg parametrisierbare Kontextfeatures, die auf räumlichen Distanzen ermittelt werden. Ihnen folgt das Feature `population` (die Varianten `populationNorm` und `populationMagnitude` folgen kurz darauf). Die aus hierarchische Beziehungen extrahierten Features `contains(p|a|s|c|d)`, `contains(a|s|c|d)` und `in(T)` sind deutlich tiefer gerankt als die räumlichen Distanzen, was deutlich macht, dass sich die vorgestellten räumliche Features als aussagekräftiger für die Disambiguierung erweisen als die hierarchischen.

Schwellwertanalyse Wie in Formel 3.1 angegeben, kann für die Klassifikation ein Wahrscheinlichkeitsschwellwert *confidenceThreshold* festgelegt werden. Unterschreiten die klassifizierten Wahrscheinlichkeiten sämtlicher Lokationskandidaten einer Annotation diesen Schwellwert, wird die Annotation als Nicht-Lokation betrachtet und sämtliche Kandidaten verworfen. Für denn Fall, dass *confidenceThreshold* = 0, werden sämtliche in der Vorverarbeitungsphase (siehe Abschnitt 3.7) extrahierten Annotationen, für die eine Entsprechung in der Gazetteer-Datenbank gefunden wird, als Lokation betrachtet und disambiguiert, womit der maximal mögliche Recall erzielt wird. Die Klassifikation dient in diesem Fall also nur zum Ranking und nicht zur Filterung potentieller Nicht-Lokationen.

Bei einer Erhöhung des Schwellwerts werden entsprechend mehr Annotationen verworfen, sofern für eine Annotation a keine Lokationskandidaten in $L(a)$ existieren, die mit einer Konfidenz von mindestens *confidenceThreshold* klassifiziert wurden. Der Recall sinkt in diesem Fall, wohingegen die Precision gesteigert werden kann. In Abbildung 3.13 sind die Ergebnisse der Schwellwertanalyse unter Verwendung des reduzierten Featuresets aus Abbildung 3.11 dargestellt im Hinblick auf Precision, Recall und F1-Maß nach MUC- und Geo-Schema (siehe Abschnitt 3.3) dargestellt.

Die Abbildung zeigt, dass bereits bei einem Schwellwert von 0 die Precision so hoch liegt, dass annähernd Bestwerte für die F1-Maße erzielt werden. Die Precision hat hier bereits einen Wert von 78,44 % nach MUC-Methode bzw. 82,61 % nach Geo-Methode. Eine Erhöhung des Konfidenzschwellwerts führt zu steigenden Werten für die Precision, was jedoch früh und überproportional zu Lasten des Recalls geht. Die Bestwerte im Hinblick auf das F1-Maß werden für einen Schwellwert von 0,2 erzielt. Nach MUC-Evaluierung beträgt die Precision hier 83,6 % bei einem Recall von 73,93 % und somit einem F1-Wert in Höhe von 78,47 %. Die Geo-Evaluierung weist bei diesem Schwellwert eine Precision von 87,94 %, einen Recall von 80,59 % und somit ein F1-Wert von 84,11 % aus.

3.12 VERGLEICH

Die zwei hier vorgestellten Verfahren werden abschließend mit den folgenden in den Abschnitten 3.2.2 und 3.2.3 vorgestellten State-of-the-Art-Systemen verglichen: Yahoo! BOSS Geo Services (nachfolgend „Yahoo“), AlchemyAPI (nachfolgend „Alchemy”), OpenCalais, Extractiv, CLAVIN und Unlock. Verwendet werden die Testsets von TUD-Loc-2013, LGL und Clust. Als Baseline wird

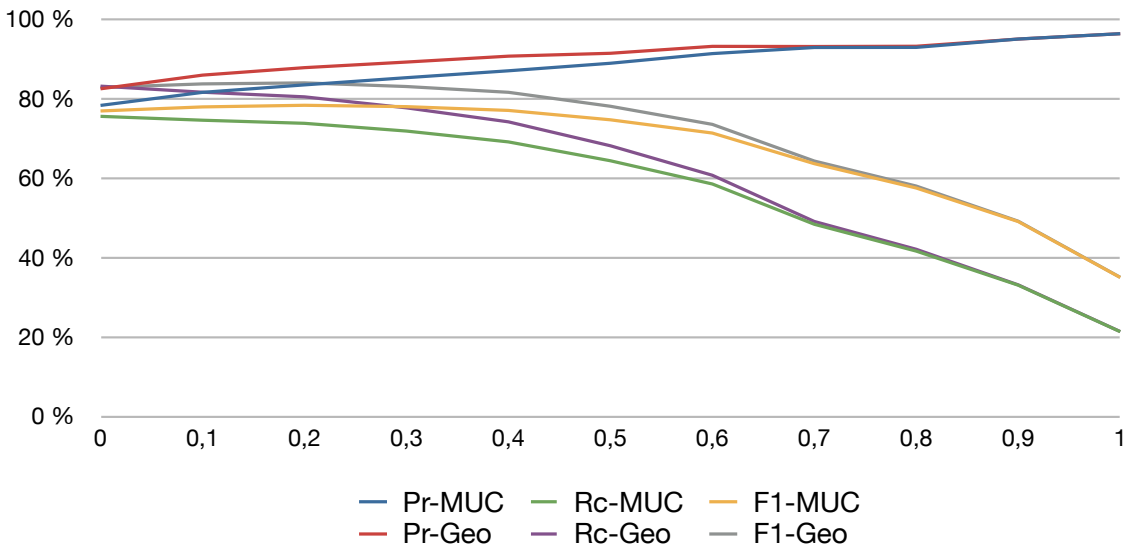


Abbildung 3.13: Schwellwertanalyse für die Machine-Learning-basierte Disambiguierung unter Verwendung des Validierungssets von TUD-Loc-2013

außerdem eine „Maximum Population"-Heuristik evaluiert; hier wird jede Annotation sofern möglich entweder als `CONTINENT` oder `COUNTRY` disambiguiert oder jener Lokationskandidat mit der höchsten Bevölkerungsanzahl gewählt.

Nicht alle der verglichenen Systeme führen sowohl eine Toponymerkennung und -klassifikation als auch eine -disambiguierung durch. OpenCalais und Extraktiv geben für einen Teil der erkannten Lokationsentitäten Koordinaten an, wohingegen Alchemy bei keiner extrahierten Lokation Koordinaten mitlieferte. Unlock auf der anderen Seite nimmt keine Typklassifikation der Lokationen vor, weshalb hier nur die Geo-Komponente evaluiert wird[79].

Da jedes System auf eine eigene Menge von Entitäts- bzw. Lokationstypen zurückgreift, musste ein Mapping zwischen den systemspezifischen Klassen und den in dieser Arbeit definierten Lokationsklassen vorgenommen werden. Das Mapping wurde unter Verwendung des Trainingssets soweit getestet und optimiert, sodass ein fairer Vergleich gewährleistet ist. Die verwendeten Mappings sind im Anhang A dokumentiert. Die gleiche Vorgehensweise wurde für das Mapping der in LGL und Clust definierten Lokationstypen angewendet, die dem GeoNames-Typsystem entsprechen (siehe Tabelle A.1).

Lokationserkennung und -klassifikation Die Evaluierung der Erkennung und Klassifikation von Lokationen wurde anhand der in Abschnitt 3.3.1 erläuterten MUC-Methode vorgenommen. Die

79 Bei der Verarbeitung der Evaluierungsdaten aus TUD-Loc-2013 durch Unlock trat bei einem Dokument der reproduzierbare Fehler „geo parser returned an error code 1" auf. Da die anderen Systeme dieses Dokument problemlos verarbeiten konnten und der Text weder in Form seines Umfangs noch besonderer Struktur aus dem Rahmen fiel, wurde dies als Fehlextraktion bewertet.

Ergebnisse sind in Tabelle 3.10 und Abbildung 3.15 dargestellt. Im Gegensatz zu Lieberman und Samet (2012), die ausschließlich die Erkennung von Lokationen im Text evaluieren, wird hier die Kombination aus Erkennung und Typklassifikation überprüft, die nach Auffassung des Autors ein wichtiger Bestandteil bei der Lokationsextraktion darstellt. Neben der Erkennung bezieht die Evaluierung also mit ein, ob die zugewiesenen Lokationstypen korrekt sind (siehe Tabelle 3.6).

		Diese Arbeit			**State-of-the-Art**				
Daten-set		**Base-line**	**Heu-ristik**	**ML**	**Yahoo**	**Alch-emy**	**Open Calais**	**Extrac-tiv**	**CLA-VIN**
TUD	Pr-MUC	61,69	81,33	81,7	64,69	78,57	**86,93**	73,38	81,33
	Rc-MUC	**74,85**	72,63	73,74	56,78	69,96	61,43	67,55	42,54
	F1-MUC	67,64	76,73	**77,52**	60,47	74,01	71,99	70,34	55,86
LGL	Pr-MUC	39,22	70,04	69,79	67,98	71,19	73,73	71,42	**77,83**
	Rc-MUC	**65,91**	58,1	63,73	59,39	59,01	40,38	59,75	34,21
	F1-MUC	49,18	63,51	**66,62**	63,4	64,53	52,18	65,06	47,53
Clust	Pr-MUC	51,45	80,96	79,71	71,12	80,11	82,95	82,41	**83,8**
	Rc-MUC	**73,61**	70,48	72,99	61,28	63,81	54,03	72,49	42,85
	F1-MUC	60,57	75,36	76,2	65,83	71,03	65,43	**77,13**	56,7

Abkürzungen: Pr = Precision, Rc = Recall, ML = Machine Learning

Tabelle 3.10: Evaluierungsergebnisse für Lokationserkennung und -klassifikation nach MUC; alle Werte in Prozent, Bestwert jeweils fett hervorgehoben

Auffallend ist hier jeweils der hohe Recall der verwendeten Baseline, der bei jedem Datenset die andere Ansätze übertrifft. Die höchste Precision auf TUD-Loc-2013 wird von OpenCalais erzielt, diese beträgt 86,93 %. Aufgrund des geringen Recalls in Höhe von 61,43 % erreicht OpenCalais jedoch nur ein F1-Maß von 71,99 %. Das beste Ergebnis in Bezug auf das harmonische Mittel F1 in Höhe von 77,52 % wird durch den hier vorgestellten Machine-Learning-basierten Ansatz (ML) erreicht, gefolgt von der Heuristik mit 76,73 %. An dritter Stelle liegt Alchemy mit 74,01 %.

Für das LGL-Datenset erzielt die hier vorgestellte ML-Strategie die besten Werte hinsichtlich des F1-Werts (66,62 %), gefolgt von Extractiv (65,06 %). Für Clust erzielt Extractiv den besten F1-Wert in Höhe von 77,13 % vor ML, womit 76,2 % F1 erreicht werden.

Lokationsdisambiguierung Die Evaluierung der Disambiguierung erfolgt anhand der in Abschnitt 3.3.2 beschriebenen Methodik. Die maximal erlaubte Abweichung *maxDistance* wird auf 100 km festgelegt. Tabelle 3.11 und Abbildung 3.15 zeigen die Resultate. Alchemy und Extractiv sind hier nicht vertreten, da diese keine Koordinaten extrahieren, statt dessen ist Unlock aufgeführt, welches keine Typklassifikation vornimmt.

Noch stärker als bei der vorangehend betrachteten Erkennung fällt hier die starke Baseline auf, die auf TUD-Loc-2013 bezüglich F1-Geo sämtliche der verglichenen State-of-the-Art-Systeme übertrifft. Der

		Diese Arbeit			State-of-the-Art			
Daten set	**Maß**	**Base-line**	**Heu-ristik**	**ML**	**Yahoo**	**Open Calais**	**Un-lock**	**CLA-VIN**
TUD	Pr-Geo	70,75	86,67	86,64	87,42	55,78	71,79	**89,88**
	Rc-Geo	58,23	**62,22**	62,09	48,81	34,83	46,81	35,41
	F1-Geo	63,88	**72,43**	72,34	62,64	42,88	56,67	50,8
LGL	Pr-Geo	58,43	70,07	**82,78**	74,51	38,38	58,73	56,84
	Rc-Geo	55,48	55,76	**71,24**	58,13	20,2	42,72	23,74
	F1-Geo	56,92	62,1	**76,58**	65,31	26,47	49,46	33,49
Clust	Pr-Geo	77,19	88,61	92,02	**93,54**	66,31	72,68	75,48
	Rc-Geo	74,59	76,1	**84,5**	72,18	44,61	50,79	33,67
	F1-Geo	75,87	81,88	**88,1**	81,48	53,34	59,79	46,57

Abkürzungen: Pr = Precision, Rc = Recall, ML = Machine Learning

Tabelle 3.11: Evaluierungsergebnisse für Lokationsdisambiguierung nach Geo-Schema; alle Werte in Prozent, Bestwert jeweils fett hervorgehoben

Grund ist vor allem in dem umfangreichen verwendeten Gazetteer zu suchen. CLAVIN erzielt auf TUD-Loc-2013 den Bestwert für die Precision in Höhe von 89,88 %, der jedoch durch den geringen Recall 35,41 % zunichte gemacht wird. Sowohl die hier vorgestellte Heuristik als auch die ML-Methode erreichen annähernd gleiche Ergebnisse für F1-Geo in Höhe von 72,34 bzw. 72,34 %.

Bei Betrachtung der LGL- und Clust-Datensets tritt eine stärkere Diskrepanz zwischen ML und Heuristik zu Tage. Hier werden die besten Werte bezüglich F1-Geo im Vergleich zu sämtlichen anderen Ansätzen jeweils mittels ML erzielt. Bei dem stark lokal geprägten LGL-Datenset, wo dies besonders zu Tage tritt, zeigt sich, dass mit der hier vorgestellten Variante ML (F1-Geo 76,58 %) eine deutlich bessere Disambiguierung erreicht wird als mit der Heuristik (F1-Geo 62,1 %). Bei Clust ist dieser Unterschied ebenfalls deutlich vorhanden, wenn auch nicht so stark ausgeprägt wie bei LGL. Die Machine-Learning-basierte Methode profitiert hier von der besseren Anpassungsfähigkeit auf die lokal geprägten Datensets.

3.13 ZUSAMMENFASSUNG

In diesem Kapitel wurden zwei neue Verfahren zur Lokationsextraktion aus englischen Texten vorgestellt; ein heuristisches und eines unter Verwendung eines Machine-Learning-basierten Klassifikators. Im Zuge dessen wurde ein umfangreiches Feature Engineering betrieben und in Abschnitt 3.9 eine Reihe neuer Features vorgestellt. Besonders hervorzuheben sind die parametrisierbaren Kontextfeatures, die in keiner anderen verwandten Arbeit bisher verwendet wurden und deren Effektivität in Abschnitt 3.11.2 gezeigt wurde.

Unter Verwendung von drei Datensets wurden Lokationserkennung/-klassifikation und Disambiguierung mit einer Reihe von State-of-the-Art-Systemen verglichen. Bei der Erkennung/-klassifikation erzielt der ML-basierte Ansatz bei zwei von drei Datensets bessere Ergebnisse bezüglich F1-MUC. Bei der Disambiguierung werden mittels ML-Ansatz auf allen drei Datensets signifikant bessere Werte für den F1-Geo erzielt. Unter Verwendung von TUD-Loc-2013 wird ein um 9,7 Prozentpunkte höherer F1-Geo gegenüber dem besten State-of-the-Art-System Yahoo erzielt. Bei LGL und Clust beträgt der Vorsprung vor Yahoo 11,27 respektive 6,69 Prozentpunkte.

Die erste Hypothese dieser Arbeit, die besagte, dass mit Techniken des Machine Learnings ein Extraktionsmechanismus für Lokationen und deren Koordinaten geschaffen werden kann, der die Extraktionsqualität von State-of-the-Art-Systeme übertrifft, wurde hiermit bestätigt.

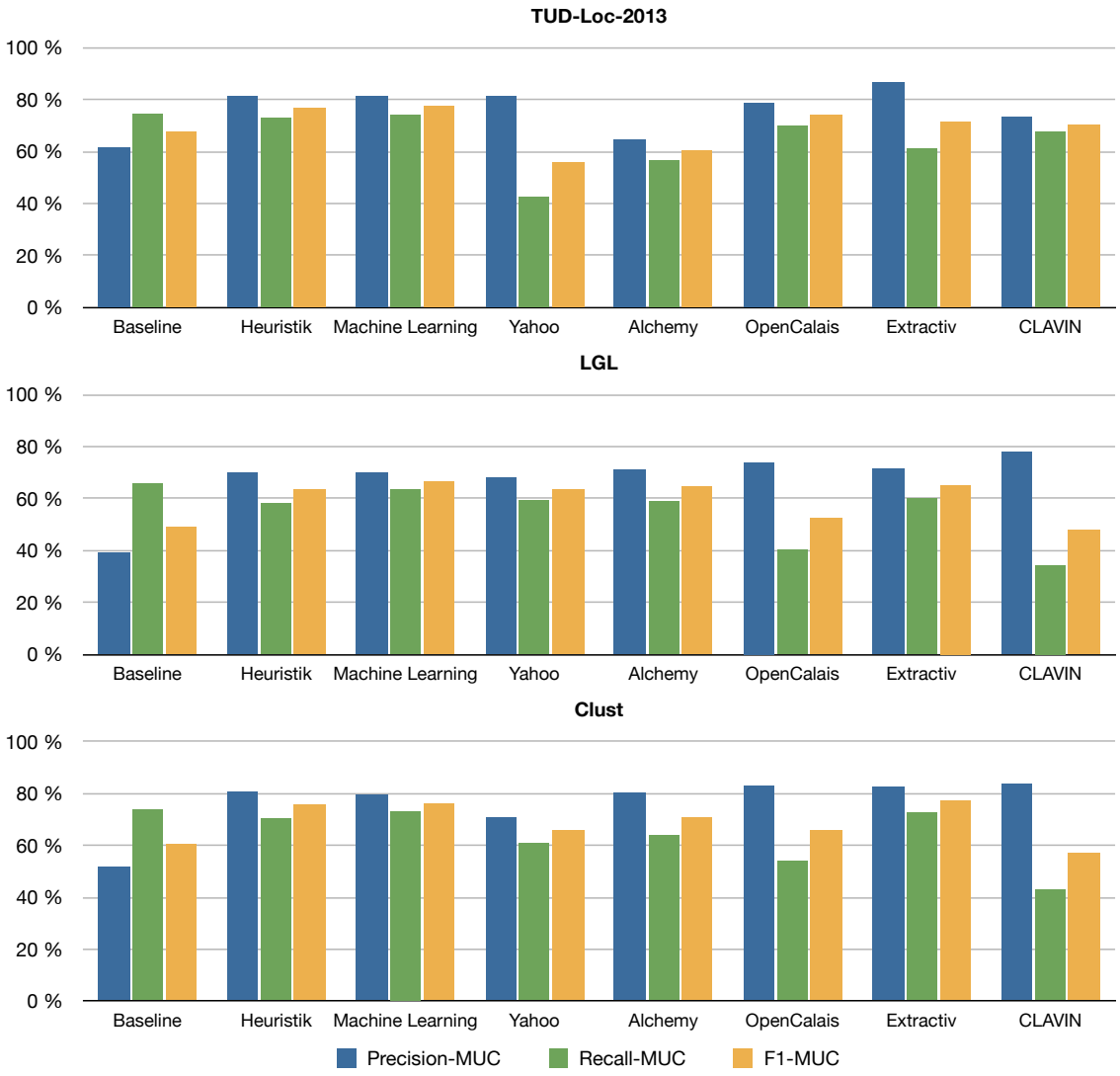


Abbildung 3.14: Evaluierungsergebnisse für Lokationserkennung und -klassifikation nach MUC

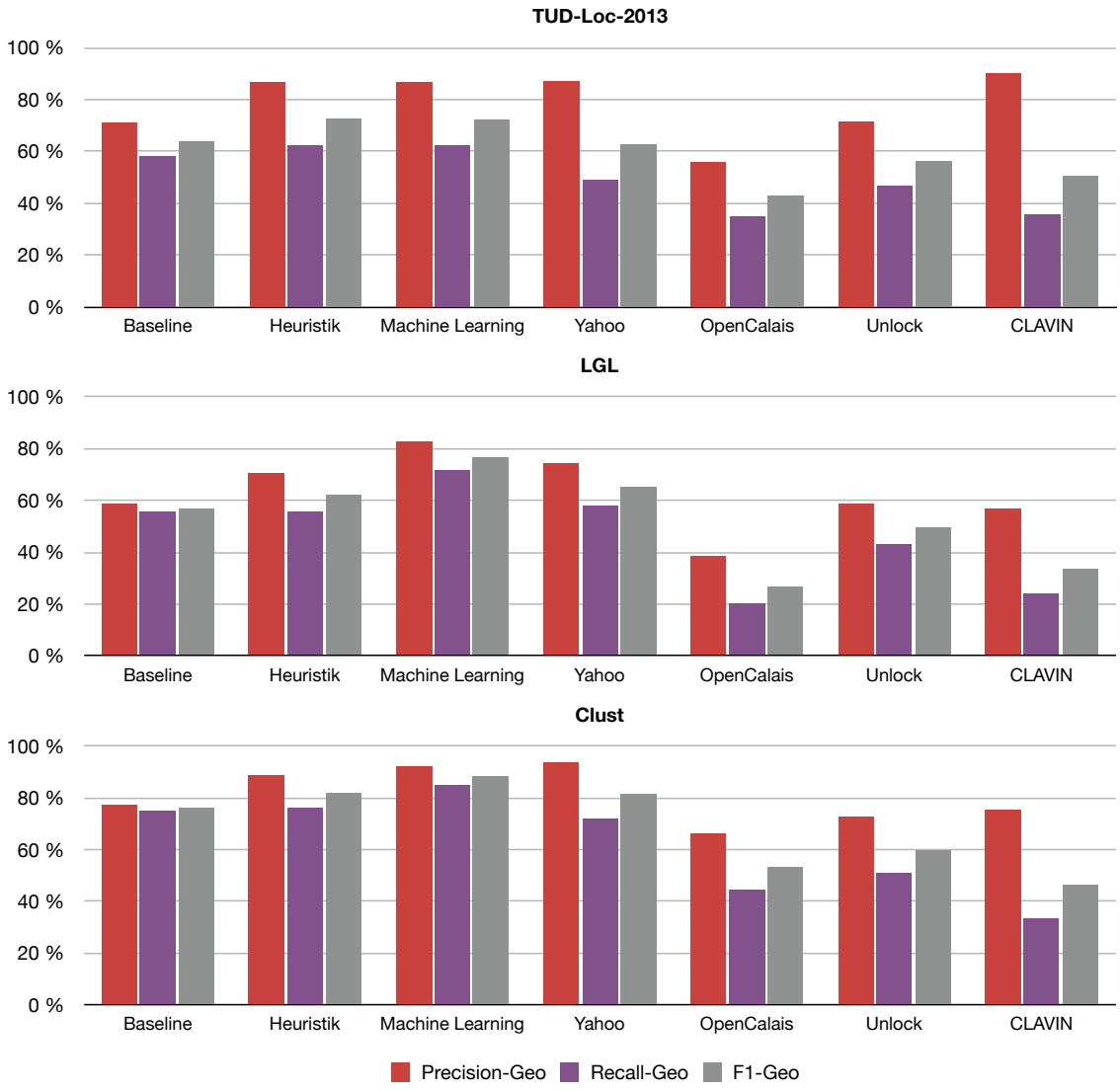


Abbildung 3.15: Evaluierungsergebnisse für Lokationsdisambiguierung nach Geo-Schema

4 FOKUSBESTIMMUNG

Mit den Methoden aus Kapitel 3 können sämtliche Lokationsvorkommen innerhalb eines Texts extrahiert werden. Wie in der Vorbetrachtung in Abschnitt 1.1 gezeigt wurde, enthalten Nachrichtenartikel jedoch meist ein Vielzahl von Ortsreferenzen. Um im Rahmen der 5W1H-Fragen den Ort eines Ereignisses bestimmen zu können, muss eine Auswahl für eine repräsentative Lokation getroffen werden. Jene Lokation, welche die Frage nach dem „Wo“ eines Ereignisses beantwortet, wird nachfolgend als „geographischer Fokus“ bezeichnet.

Geographischer Fokus Im Gegensatz zu der Lokationsextraktion, bei der eine klare Definition des Problems möglich ist, lässt die Wahl des geographischen Fokus einen größeren Interpretationsspielraum, der auch in verwandten Arbeiten unterschiedlich ausgelegt wird (siehe Abschnitt 4.1). Diese Arbeit definiert den Begriff wie folgt: Der geographische Fokus eines Dokuments wird durch genau eine Lokation oder Koordinate repräsentiert, die den Inhalt eines Dokuments so allgemein wie notwendig und so spezifisch wie möglich wiedergibt. Anhand eines Beispiels soll dies verdeutlicht werden: Ein Dokument, in dem Pariser Sehenswürdigkeiten wie „Eiffel Tower“, „Place de la Concorde“ und „Louvre“ vorkommen, hat als Fokus „Paris“. Einem Dokument mit den Sehenswürdigkeiten „Eiffel Tower“, „Charles Bridge“ und „Parc Güell“ hingegen würde der Fokus „Europe“ zugewiesen. Die hier vorgestellten Verfahren können den Kategorien „wissensbasiert“ und „datenbasiert“ zugeordnet werden, eine Einordnung zeigt Abbildung 4.1.

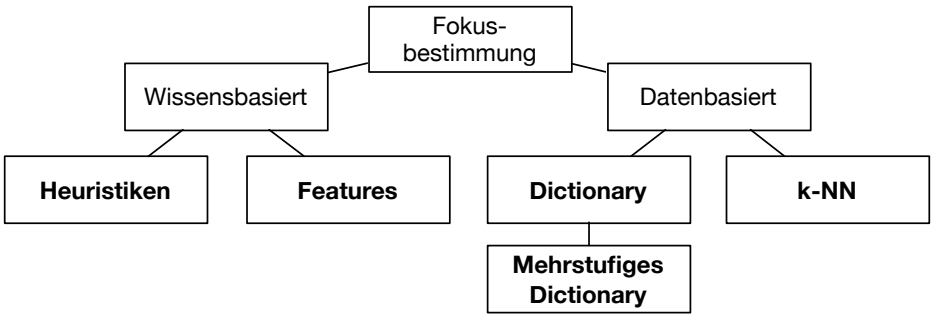


Abbildung 4.1: Einordnung der nachfolgend vorgestellten Verfahren zur Fokusbestimmung

Wissensbasierte Fokusbestimmung Wissensgetriebene Verfahren nutzen wie die im vorangehenden Kapitel beschriebenen Datenbanken mit Lokationen für die Fokusbestimmung. Die nachfolgend vorgestellten Mechanismen setzen jeweils eine Menge bereits extrahierter Lokationen voraus und nehmen ein Ranking vor, um die repräsentativste Lokation zu bestimmen. Dazu werden verschiedene Heuristiken beschrieben (siehe Abschnitt 4.2). Darauf aufbauend wird in Abschnitt 4.3 eine Machine-Learning-basierte Rankingmethode vorgestellt, welche die einzelnen Heuristiken mit weiteren Features kombiniert, um die Vorhersagegenauigkeit zu steigern.

Datenbasierte Fokusbestimmung Die im weiteren Verlauf präsentierten Strategien sind datengetrieben und bedienen sich der Textklassifikation (siehe Abschnitt 4.4). Im Gegensatz zu den wissensgetriebenen Strategien wählen sie den Fokus nicht aus einer zunächst extrahierten Menge von Lokationen, sondern verwenden ausschließlich Textfeatures, um unter Verwendung statistischer Methoden eine Koordinate als Fokus zu ermitteln. Die Motivation hinter diesen Ansätzen rührt daher, dass die in Dokumenten vorkommenden Wörter eine Reihe impliziter Indikatoren über ihren Standort liefern. Ein Text, welcher häufig den Term „gondola“ erwähnt, bezieht sich statistisch gesehen eher auf den Ort Venedig in Italien, als ein Text, in dem häufig die Terme „cable car“ vorkommen. Die textklassifikationsbasierten Stategien nutzen kein Wissen aus einem Gazetteer, sondern greifen auf ein aus Textdaten trainiertes Modell zurück. Nachfolgend werden drei textklassifikationsbasierte Ansätze vorgestellt. Zwei von diesen verwenden ein Dictionary, einer führt eine k-Nearest-Neighbor-Klassifikation durch und ermittelt den Fokus durch Betrachtung ähnlicher Dokumente, die während des Trainings zur Verfügung gestellt wurden.

4.1 VERWANDTE ARBEITEN

Eine Reihe verschiedener Arbeiten beschäftigt sich mit der geographischen Fokusbestimmung. In verwandten englischsprachigen Arbeiten wird das Problem oftmals als „Scope Resolution“, „Scope Estimation“ oder „Georeferencing“ bezeichnet. Die nachfolgend vorgestellten Arbeiten werden den vorangehend beschriebenen Kategorien „wissensbasiert“ und „datenbasiert“ zugeordnet.

4.1.1 WISSENSBASIERTE FOKUSBESTIMMUNG

In einer der ältesten Arbeiten zur automatisierten Fokusbestimmungen stellen Ding, Gravano und Shivakumar (2000) zwei Methoden vor, um die geographische Relevanz von Webdokumenten zu bestimmen. Die erste Methode analysiert die geographische Herkunft eingehender Hyperlinks. Die zweite Methode betrachtet die explizite Nennung von Lokationsnamen im Textinhalt der Seiten. Es werden zwei Hypothesen angewendet: (1) Eine signifikante Anzahl von Links bzw. Ortsnennungen muss in der Ressource vorhanden sein bzw. auf diese verweisen. (2) Links bzw. Ortsnennungen müssen gleichmäßig über die geographische Fokuslokation verteilt sein. Beiden Methoden liegt eine feste dreistufige Hierarchie mit den USA als Wurzel sowie Staaten und Städtenamen zugrunde.

Amitay et al. (2004) führen die Fokusbestimmung im Anschluss an die Lokationsextraktion durch (siehe Abschnitt 3.2). Maßgebliches Kriterium ist die hierarchische Struktur der extrahierten Lokationen. So werden Scores für die Extraktionen basierend auf der Vorkommenshäufigkeit und der Konfidenz aus der Extraktion ermittelt und diese zu einem gewissen Anteil an die Elternlokationen in der Hierarchie propagiert, die aus einem Gazetteer mit fast 40.000 Orten weltweit stammt. Anschließend wird eine festgelegte Anzahl von Lokationen, die einen definierten Scoring-Schwellwert überschreiten, als Fokuslokationen ausgewählt.

Zong et al. (2005) segmentieren Webseiten und bestimmen den geographischen Fokus unter Verwendung hierarchischer Beziehungen. Für ausreichend große Segmente wird ein Teilbaum des Gazetteers konstruiert, der sämtliche Lokationsvorkommen des Segments und deren Vorfahren enthält. Für Blattknoten wird anschließend unter Berücksichtigung der Vorkommenshäufigkeit im Segment ein Scoring ermittelt. Innere Knoten des Teilbaums erhalten ein Scoring, welches sich aus der Summe und Verteilung der Kind-Scorings ermittelt. Lokationen, welche einen festgelegten Schwellwert überschreiten, werden als Fokus bestimmt. Der Gazetteer ist auf US-Lokationen beschränkt.

Wang, Xie, Wang, Lu und Ma (2005a, 2005b) schlagen vor, bei Webseiten explizit zwischen den Typen „Provider Location“, „Content Location“ und „Serving Location“ zu unterscheiden und stellen Algorithmen zur Bestimmung vor. Während die Provider-Lokation den Ort des Anbieters der Webseite angibt, bezeichnet die Serving-Lokation den geographischen Bereich der Zielgruppe. Die Content-Lokation ist der geographische Fokus des Seiteninhalts, wie er auch im Rahmen dieser Arbeit definiert ist. Die Extraktion der Content-Lokation erfolgt auf Basis von Hyperlinkstrukturen und hierarchischen Eigenschaften wie in Ding et al. (2000) und Amitay et al. (2004).

Ein Ranking extrahierter Lokationen mittels PageRank-Algorithmus (Page, Brin, Motwani und Winograd, 1999) nehmen Martins und Silva (2005) vor. Dazu werden aus einem Text extrahierte Lokationen, die als Knoten repräsentiert werden, nach Auftrittshäufigkeit gewichtet. Das propagierte Gewicht zwischen zwei Knoten wird aus dem Relationstyp der zugrundeliegenden Ontologie ermittelt. Die experimentellen Ergebnisse zeigen, dass mit PageRank bessere Resultate als mit einer einfachen „Most Frequent“-Heuristik erzielt werden.

Einen ähnlichen Weg wie Zong et al. (2005) gehen Campelo und de Souza Baptista (2008). Extrahierte Lokationen werden auf einen Baum abgebildet, der sowohl explizit im Text vorhandene und implizit als Elternlokationen extrahierte Knoten enthält. Für die Gewichtung der Knoten wird hier jedoch auf Basis der spatialen Verteilung eine geographische Dispersion berechnet.

Andogah et al. (2008) greifen auf einen Dokumentindex für die Fokusbestimmung zurück. Dieser enthält Dokumente, die Referenzgebiete wie Kontinente, Länder, Provinzen und Regionen repräsentieren. Pro Dokument werden mehrere Datenfelder mit Lokationsnamen unterschiedlicher Kategorien (z. B. Nachbarn, Hauptstädte, enthaltene Regionen etc.) erfasst. Aus dem zu verarbeitenden Dokument werden Lokationsnamen extrahiert und eine Suchanfrage an den Index gestellt.

Das Suchresultat stellt das Ergebnis mit gerankten Fokusregionen dar. Ein weiteres als „GeoVIPs“ bezeichnetes Konzept (Andogah, 2010) erweitert den Index um länderspezifische, charakteristische Personennamen, beispielsweise von Politikern.Die Fokusbestimmung wird auch zur Lokationsextraktion und -disambiguierung eingesetzt.

Alexopoulos und Ruiz (2012) präsentieren eine Lösung für die Lokationsextraktion und Fokusbestimmung, die domänenspezifische Ontologien nutzt. Die Ontologien beinhalten Nicht-Geo-Entitäten zu verschiedenen Lokationen, die als Hinweise herangezogen werden, um Lokationsvorkommen korrekt zu disambiguieren. Die Ontologien müssen jedoch maßgeschneidert für den jeweiligen Anwendungszweck mit manuellem Aufwand erzeugt werden.

4.1.2 DATENBASIERTE FOKUSBESTIMMUNG

Serdyukov, Murdock und van Zwol (2009) repräsentieren die Erdoberfläche durch Rasterzellen und wenden ein probabilistisches Sprachmodell an, um Bilder von Flickr anhand der zugeordneten Tags zu positionieren. Die Rasterzellen enthalten die Wahrscheinlichkeiten für die Tags aus den Trainingsdaten. Gemäß Maximum-Likelihood-Bestimmung wird die höchste Wahrscheinlichkeit für die betrachtete Tag-Kombination im Hinblick auf jede Zelle ermittelt. Glättungsmethoden, welche dem Problem dünnbesetzter Rasterzellen vorbeugen, konnten die Genauigkeit bei der Vorhersage weiter verbessern.

Vorhersagen über den Aufenthaltsort von Twitter-Nutzern durch Analyse der Tweet-Texte treffen Cheng, Caverlee und Lee (2010). Kernstück bildet ein Sprachmodell, welches zur Maximum-Likelihood-Bestimmung herangezogen wird. Die Autoren identifizieren zwei maßgebliche Probleme: Die Wörter in den Tweets eignen sich unterschiedlich gut für die Vorhersage. Deshalb identifiziert das vorgestellte Verfahren „lokale“ Wörter, die konzentriert an bestimmten Orten vorkommen und sich im Gegensatz zu stark „verstreuten“ Wörtern besser für die Vorhersage eignen. Das zweite Problem ist die spärliche Menge vorhandener Terme, die zur Klassifikation zur Verfügung stehen. Die Autoren präsentieren unterschiedliche Glättungsmethoden, die beispielsweise Termvorkommen von Nachbarlokationen mit berücksichtigen. Wohingegen die damit erzielten Verbesserungen gering sind, kann durch die Filterung lokaler Terme eine beträchtliche Steigerung der Vorhersagegenauigkeit erzielt werden.

Baba, Ishikawa und Honiden (2010) stellen ein Verfahren zur Fokusbestimmung für Flickr-Bilder auf Basis von Tags vor. Für jedes einzelne Tag werden zugehörige Orte als zweidimensionale Wahrscheinlichkeitsverteilungen repräsentiert und zur Bestimmung des Fokus eine Maximum-Likelihood-Ermittlung vorgenommen.

Ebenfalls für Flickr ermitteln Laere, Schockaert und Dhoedt (2010b) geographische Koordinaten. Die Untersuchungen beschränken sich auf eine Auswahl von Bildern aus lediglich 55 europäischen Städten. Trainingsdaten werden nach Koordinaten geclustert und ein Naïve-Bayes-Klassifikator

trainiert, der anhand zugewiesener Tags die Zugehörigkeit zu den Clustern klassifiziert. In Laere, Schockaert und Dhoedt (2010a) wird die Anzahl der erzeugten Cluster variiert, um eine Vorhersage in unterschiedlichen Granularitätsstufen zu erlauben und eine Kombination der Vorhersagen aus den unterschiedlichen Stufen vorgenommen, womit die Genauigkeit gesteigert werden kann. Laere, Schockaert und Dhoedt (2011) kombinieren das Sprachmodell aus vorherigen Arbeiten mit einer Ähnlichkeitssuche, welche nach einer groben Klassifikation vorgenommen wird und versucht, aus ähnlichen bekannten Dokumenten auf genaue Koordinaten zu schließen.

In Eisenstein, O'Connor, Smith und Xing (2010) wird ein linguistisches generatives Sprachmodell vorgestellt, welches unterschiedliche Themen und geographische Regionen beinhaltet. Das komplexe Modell besteht aus einer Vielzahl von Zufallsvariablen. Die Annahme der Autoren ist, dass sich Themen wie „sports" oder „entertainment" in verschiedenen Regionen aus unterschiedlichem Vokabular zusammensetzen. Dies wird unter anderem zur Lokalisierung von Twitter-Nutzern innerhalb der USA genutzt. Die Autoren stellen den in der Arbeit verwendeten „Geo-tagged Microblog Corpus" zur Verfügung[80]. Mit diesem Datenset erreichen sie unter Verwendung des vorgestellten Modells eine Abweichung von den Referenzkoordinaten um 494 km nach Median bzw. 900 km nach arithmetischem Mittel.

Wing und Baldridge (2011) beschreiben ein Verfahren zur Fokusbestimmung, welches ebenso wie Serdyukov et al. (2009) Rasterzellen in Kombination mit Textähnlichkeiten verwendet. Mittels Kullback-Leibler-Divergenz (Kullback und Leibler, 1951) wird die Ähnlichkeit zwischen den Termverteilungen in den einzelnen Rasterzellen und den zu lokalisierenden Dokumenten ermittelt. Für Wikipedia-Seiten konnte bei der Klassifikation eine Medianabweichung von 11,8 km und eine Abweichung nach arithmetischem Mittel von 221 km erzielt werden. Bei Twitter unter Verwendung des Datensets aus Eisenstein et al. (2010) betrug der Median dagegen 479 km, das arithmetische Mittel 967 km. Wing und Baldridge (2011) stellen das als „TextGrounder" bezeichnete System unter Open-Source-Lizenz zur Verfügung. Im Rahmen der Evaluierungen dieser Arbeit war es jedoch nicht möglich, dieses erfolgreich zu nutzen, da der Importvorgang des Wikipedia-Korpus mit einer Fehlermeldung abgebrochen wurde.

In Roller, Speriosu, Rallapalli, Wing und Baldridge (2012) wird vorgeschlagen, statt einer festen Rastergröße wie in Wing und Baldridge (2011) adaptive Raster zu verwenden, bei denen sich die Zellengröße dynamisch an die Anzahl enthaltener Dokumente anpasst. Realisisiert werden diese in Form von k-d-Bäumen, bei denen Knoten dann geteilt werden, wenn eine festgelegte Anzahl enthaltener Dokumente überschritten wird. Durch Verwendung der adaptiven Zellgrößen kann die Genauigkeit bei der Fokusbestimmung im Vergleich zu Wing und Baldridge (2011) leicht gesteigert werden.

80 http://www.ark.cs.cmu.edu/GeoText/

4.1.3 WEITERE ANSÄTZE

Ausschließlich Mechanismen des Scene Matchings aus der Computer Vision nutzen Hays und Efros (2008), um Bilder anhand visueller Features zu lokalisieren. Für die extrahierten Features werden mittels k-Nearest-Neighbor-Klassifikation visuell ähnliche Bilder aus Trainingsdaten mit Koordinaten ermittelt, aus denen eine Vorhersage bezüglich geographischer Position getroffen wird. Für die verwendeten Daten, bestehend aus über 6 Mio. Bildern von Flickr, können hier Median-Fehler von etwas über 3.000 km erreicht werden.

Crandall, Backstrom, Huttenlocher und Kleinberg (2009) klassifizieren Fotos in eine feste Auswahl von vorher trainierten Orten, die initial mittels Clustering bestimmt werden. Die Auswertungen der Autoren zeigen, dass die Kombination visueller und textueller Features (im vorliegenden Fall zugewiesene Tags) bei der Klassifikation deutlich bessere Resultate erzielt als die ausschließliche Verwendung nur einer der beiden Featurekategorien.

Hauff und Houben (2012) zeigen, dass die Lokalisierung von Flickr-Bildern deutlich verbessert werden kann, wenn zusätzlich zu intrinsischen Metadaten der Bilder in Form von Tags und Titeln externe Informationen über die jeweiligen Urheber zur Vorhersage herangezogen werden. So steigt die Vorhersagegenauigkeit deutlich, wenn von den jeweiligen Nutzern Inhalte von Twitter mit ausgewertet werden.

Gerade in Bezug auf die Lokalisierung von Twitter-Nutzern sind in jüngster Zeit eine Reihe von Arbeiten veröffentlicht worden, die Variationen der hier bereits aufgeführten Ansätze darstellen und deshalb hier nicht weiter detailliert werden sollen. Dies umfasst beispielsweise Graham, Hale und Gaffney (2013); Han, Cook und Baldwin (2013); Hong, Ahmed, Gurumurthy, Smola und Tsioutsiouliklis (2012); Mahmud, Nichols und Drews (2013); Peregrino, Tomás und Llopis (2013); Priedhorsky, Culotta und Valle (2014).

Im Rahmen des Placing Tasks der MediaEval-Workshops[81] werden durch die Teilnehmer seit 2010 verschiedene Strategien für die Koordinatenermittlung von Bild- und Videodaten vorgestellt. Die angewendeten Methoden reichen von reinem Text Mining wie von Trevisiol, Jégou, Delhumeau und Gravier (2013) über Methoden aus der Computer Vision wie beispielsweise von Penatti, Li, Almeida und da S. Torres (2012) hin zu hybriden Ansätzen wie von Kelm, Schmiedeke und Sikora (2011), welche Text und visuelle Features für die Vorhersage kombinieren.

4.1.4 ZUSAMMENFASSUNG VERWANDTER ARBEITEN

Tabelle 4.1 gibt einen Überblick über die relevanten verwandten Arbeiten und die jeweils angewendeten Methoden zur Fokusbestimmung. Methoden, welche auf der Computer Vision aufbauen, liegen außerhalb des Fokus dieser Arbeit. Der Überblick zeigt jedoch, dass auch zur Lokalisierung von

81 http://www.multimediaeval.org

Bilddaten in verwandten Arbeiten hauptsächlich auf Methoden der Textanalyse zurückgegriffen wird. Während die älteren Arbeiten bis 2008 die Fokusbestimmung wissensbasiert vornahmen, arbeiten die neueren Arbeiten maßgeblich datenbasiert. Ein direkter Vergleich zwischen diesen Methoden wurde in der Literatur bisher jedoch nicht gezogen. Nachfolgend werden aus beiden Bereichen neue Strategien zur Fokusbestimmung vorgestellt und eine direkte Gegenüberstellung unter Verwendung mehrerer Datensätze vorgenommen.

	Domäne			**Kategorie**		**Methoden**					
Autor und Jahr	Web-seiten	Text	Bilder	Wissens-basiert	Daten-basiert	Graphen, Bäume	Text-statistik	Computer Vision	Räuml. Raster	Räuml. Clustering	sonstige
Ding et al. (2000)	✓			✓		✓					
Amitay et al. (2004)		✓		✓		✓					
Zong et al. (2005)	✓			✓		✓					
Martins und Silva (2005)		✓		✓		✓					
Campelo und de S. B. (2008)		✓		✓		✓					
Andogah et al. (2008)		✓		✓			✓				
Hays und Efros (2008)			✓		✓			✓			
Serdyukov et al. (2009)			✓		✓		✓		✓		
Crandall et al. (2009)			✓		✓		✓	✓		✓	
Eisenstein et al. (2010)		✓			✓		✓				
Cheng et al. (2010)		✓			✓		✓				
Baba et al. (2010)			✓		✓		✓				
Laere et al. (2010a, 2010b, 2011)			✓		✓		✓				
Wing und Baldridge (2011)		✓			✓		✓		✓		
Roller et al. (2012)		✓			✓		✓		✓		
Alexopoulos und Ruiz (2012)		✓		✓							✓
Hauff und Houben (2012)			✓		✓		✓		✓		✓

Tabelle 4.1: Überblick über verwandte Arbeiten zur Fokusbestimmung (chronologisch sortiert)

4.2 RANKING MITTELS HEURISTIKEN

Die nachfolgend vorgestellten Ranking-basierten Verfahren können als Funktionen focus(L) betrachtet werden, die aus einer nichtleeren Menge von extrahierten und disambiguierten Lokationen L, welche mittels der in Kapitel 3 beschriebenen Lokationsextraktion ermittelt wurden, genau eine Lokation $l \in L$ als Fokus auswählen. Dies geschieht nach den nachfolgend beschriebenen Kriterien.

Position Die erste im Text auftretende Lokation wird, wie in Formel 4.1 angegeben, als Hauptlokation betrachet. Diese Strategie nutzt aus, dass in vielen Texten, vor allem bei Nachrichten, der relevante Ort als erster im Text genannt wird.

$$\text{focus}_{position}(L) = \arg\min_{l \in L} \text{offset}(l) \tag{4.1}$$

Einwohnerzahl Es wird gemäß Formel 4.2 jene Lokation mit der höchsten Einwohnerzahl oder das erste Vorkommen im Text vom Typ `CONTINENT` oder `COUNTRY` als Hauptlokation ausgewählt. Der zugrunde liegende Gedanke ist, dass große Orte die Hauptlokation eines Dokuments besser wiedergeben als kleine, mustmaßlich weniger bekannte Orte.

$$\text{focus}_{population}(L) = \begin{cases} \arg\min_{l \in L} \text{offset}(l) & \text{wenn } \text{type}(l) \in \{\texttt{CONTINENT, COUNTRY}\} \\ \arg\max_{l \in L} \text{population}(l) & \text{sonst} \end{cases} \tag{4.2}$$

Frequenz Wie in Formel 4.3 angegeben wird die am häufigsten im Text auftretende Lokation als Hauptlokation betrachtet. Die Annahme dabei ist, dass relevante Orte im Text mehrfach genannt und wiederholt werden.

$$\text{focus}_{frequency}(L) = \arg\max_{l \in L} \text{count}(l) \tag{4.3}$$

Mittelpunkt Für sämtliche im Text vorkommende Lokationen L wird der geographische Mittelpunkt $\text{midpoint}(L)$ bestimmt. Als Hauptlokation wird jene bestimmt, die die geringste Distanz zum Mittelpunkt hat (siehe Formel 4.4 und Abschnitt 2.1.3). Der zugrunde liegende Gedanke bei dieser Strategie ist, dass bei der Nennung mehrerer Lokationen im Text ein zentral liegender Punkt den geographischen Fokus gut wiedergeben kann.

$$\text{focus}_{midpoint}(L) = \arg\min_{l \in L} \text{distance}(l, \text{midpoint}(L)) \tag{4.4}$$

Minimale Distanz Hier wird, wie in Formel 4.5 angegeben, die Lokation mit der minimalen Distanz zu allen anderen Lokationen als Hauptlokation bestimmt. Die Vorgehensweise ähnelt der in Abschnitt 2.1.4 beschriebenen Methodik zur Ermittlung des geometrischen Medians, jedoch wird hier kein beliebiger Punkt berechnet, sondern jene aus gegebenen Lokationen L ausgewählt, der die Minimalitätsbedingung erfüllt. Im Gegensatz zur zuvor beschriebenen Heuristik „Mittelpunkt“ werden hier einzelne „Ausreißer“, die möglicherweise von einer Gruppe nah beieinander liegender Lokationen weit entfernt sind, weniger stark berücksichtigt.

$$\text{focus}_{minimumDistance}(L) = \arg\min_{l \in L} \sum_{m \in L} \text{distance}(l, m) \tag{4.5}$$

Trust Bei der Klassifikation durch die Machine-Learning-basierte Lokationsextraktion wurden Wahrscheinlichkeiten ermittelt, die das „Vertrauen“ des Klassifikators bezüglich korrekter Klassifizierung angeben (siehe Abschnitt 3.9). Bei der Strategie „Trust“ wird jene Lokation ausgewählt, der bei der Klassifikation die höchste Wahrscheinlichkeit zugewiesen wurde (siehe Formel 4.6).

$$\text{focus}_{maximumTrust}(L) = \arg\max_{l \in L} \text{trust}(l) \tag{4.6}$$

4.3 RANKING MITTELS MACHINE LEARNING

Diese Strategie baut auf den vorgestellten Heuristiken (Abschnitt 4.2) auf und verwendet diese als numerische Features. Desweiteren werden eine Reihe weiterer Features extrahiert, die aus hierarchischen Eigenschaften der Lokationen (`descendantCount`, `ancestorCount`, `descendantPercentage`, `ancestorPercentage`), den Positionen im Text (`offsetFirst`, `offsetLast` und `offsetSpread`), sowie einer Reihe von Distanzstatistiken zu anderen Lokationen ermittelt werden. Tabelle 4.2 gibt einen detaillierten Überblick über die insgesamt 27 (mit einer Ausnahme durchweg numerischen) Features. Wie die Analyse der verwandten Arbeiten zeigte (siehe Abschnitt 4.1 und Tabelle 4.1), existiert bisher kein vergleichbares Verfahren zur Bestimmung des geographischen Fokus.

Ein trainierter, binärer Klassifikator (Formel 4.7) klassifiziert die Wahrscheinlichkeit jedes Toponyms für die Klasse „Hauptlokation". Jenes Toponym mit dem höchsten Konfidenzwert für die Klasse „Hauptlokation" wird als Fokus extrahiert.

$$\text{focus}_{classifier}(L) = \arg\max_{l \in L} \text{classify}(l) \tag{4.7}$$

Zum Training wird jenes Toponym mit der minimalen Distanz zur tatsächlichen Fokuslokation ermittelt. Unterschreitet die Distanz einen festgelegten Schwellwert[82], wird ein positives Trainingsbeispiel generiert, ansonsten ein negatives. Für sämtliche verbleibenden Toponyme wird ebenfalls ein negatives Trainingsbeispiel erzeugt. Während im Normallfall die annotierte Fokuslokation und eine der extrahierten Toponyme identisch sind, wird durch den Distanzschwellwert erreicht, dass auch ein positives Trainingsbeispiel generiert werden kann, wenn die annotierte Fokuslokation nicht extrahiert wurde, jedoch eine räumlich naheliegende Lokation in L existiert.

4.4 TEXTKLASSIFIKATION ZUR FOKUSBESTIMMUNG

Die nachfolgenden zwei vorgestellten Verfahren adressieren die Fokusbestimmung als Textklassifikationsproblem. Im Gegensatz zu den vorangehend beschriebenen Strategien, welche ein Ranking extrahierter Lokationen vornehmen, verarbeiten die nachfolgend vorgestellten Verfahren direkt einen Dokumenttext D. Die notwendigen Modelle werden hier ebenfalls direkt aus Texten generiert, die über Koordinaten verfügen, die den geographischen Fokus ausdrücken.

4.4.1 DICTIONARY-BASIERTE STRATEGIE

Die Dictionary-basierte Strategie baut aus den Trainingsdokumenten ein Wörterbuch auf, welches die Vorkommenshäufigkeiten der einzelnen Merkmale pro Zelle $\text{count}(t, cell)$ erfasst. Um zwischen den kategorialen Werten des Klassifikators und den spatialen Werten der Geokoordinaten abzubilden,

82 In den nachfolgenden Experimenten wurde eine maximale Distanz von 50 km zwischen Referenz-Fokuslokation und Trainingstoponym gewählt.

Bezeichnung	Typ	Beschreibung
Lokationsfeatures		
`populationNorm`	num.	Einwohnerzahl normiert mit höchster Einwohnerzahl
`populationMagnitude`	num.	Größenordnung der Einwohnerzahl
`locationType`	nom.	Typ der Lokation
Distanzfeatures		
`midpointDistance`	num.	Distanz vom geographischen Mittelpunkt aller Lokationen
`midpointDistanceNorm`	num.	`midpointDistance` normalisiert mit höchstem Wert
`centerpointDistance`	num.	Distanz vom geographischen Median aller Lokationen
`centerpointDistanceNorm`	num.	`centerpointDistance` normalisiert mit höchstem Wert
`minDistanceToOthers`	num.	Minimale Distanz zu anderen Lokationen
`maxDistanceToOthers`	num.	Maximale Distanz zu anderen Lokationen
`meanDistanceToOthers`	num.	Arithmetisches Mittel der Distanzen zu anderen Lokationen
`medianDistanceToOthers`	num.	Median der Distanzen zu anderen Lokationen
`minDistanceToOthersNorm`	num.	Wie oben, normalisiert mit maximaler Distanz zwischen sämtlichen Lokationen
`maxDistanceToOthersNorm`	num.	
`meanDistanceToOthersNorm`	num.	
`medianDistanceToOthersNorm`	num.	
Hierarchische Features		
`descendantCount`	num.	Anzahl vorkommender Nachfahrenslokationen
`descendantPercentage`	num.	Anteil vorkommender Nachfahrenslokationen
`ancestorCount`	num.	Anzahl vorkommender Vorfahrenslokationen
`ancestorPercentage`	num.	Anteil vorkommender Vorfahrenslokationen
`hierarchyDepth`	num.	Tiefe in der Hierarchie
`hierarchyDepthNorm`	num.	`hierarchyDepth` normalisiert mit höchster Tiefe
Textfeatures		
`occurrenceCount`	num.	Vorkommenshäufigkeit der Lokation
`occurrenceFrequency`	num.	`occurrenceCount` normalisiert mit Gesamtanzahl der Lokationen
`offsetFirst`	num.	Zeichenposition des ersten Auftretens normalisiert mit Gesamtlänge
`offsetLast`	num.	Zeichenposition des letzten Auftretens normalisiert mit Gesamtlänge
`offsetSpread`	num.	Normalisierte Differenz zwischen erstem und letztem Auftreten
Sonstige Features		
`disambiguationTrust`	num.	Wahrscheinlichkeitswert aus Disambiguierung (siehe Abschnitt 3.9)

Abkürzungen: nom. = nominal, num. = numerisch

Tabelle 4.2: Features für Machine-Learning-basierte Fokusbestimmung

wird ein Raster verwendet, welches die Erde konzeptionell in Zellen teilt, die in horizontaler und vertikaler Richtung jeweils einen Bereich von $d°$ umspannen. Die Zellen sind folglich am Äquator annähernd quadratisch und werden in Richtung der Pole schmaler[83]. Die einzelnen Zellen werden mittels Tupel aus x- und y-Identifikator bezeichnet, beispielsweise „(05, 12)“.

Tabelle 4.3 zeigt ein beispielhaftes Dictionary bestehend aus zehn unterschiedlichen Termen und vier Rasterzellen. Im Gegensatz zu gewöhnlichen Textklassifikationsproblemen muss hier jedoch zwischen einer exorbitanten Anzahl trainierter Rasterzellen unterschieden werden. Bei einer Größe von 0,1° beträgt die theoretische Anzahl an Zellen beispielsweise 6,48 Mio. (in der Praxis sind viele dieser Zellen jedoch leer, wie später gezeigt wird).

	cell ∈ *Cells*				
t ∈ *Terms*	(05, 12)	(11, 13)	(26, 11)	(32, 05)	count(t)
american	53	18	3	1	75
bay	45	18	3	7	73
canadian	1	77	1		79
county	97	82	3	3	185
indian	4	6	72		82
municipality		43	17	2	62
pacific	55	3		6	64
scotia		86			86
valley	43	9	5	12	69
victoria		7	1	83	91
termCount(*cell*)	298	349	105	114	866
docCount(*cell*)	163	151	208	133	655

Tabelle 4.3: Beispielausschnitt für die Term-Zell-Matrix eines Dictionarys mit vier Zellen und zehn Termen und der dazugehörigen count(t, *cell*), sowie der Anzahl der observierten Zellen count(*cell*) während des Trainings für die Bestimmung der A-priori-Wahrscheinlichkeiten

Die Merkmalsextraktion, nachfolgend mit preprocess(D) bezeichnet, extrahiert komplette Terme oder wort- bzw. zeichenbasierte n-Gramme (siehe Abschnitt 2.2). Es werden zwei Klassifikationsfunktionen vorgestellt, die die Wahrscheinlichkeit p(*cell* | D) ermitteln, die angibt, dass ein Dokument D einer Zelle *cell* angehört. Die einfachere Variante (Formel 4.8 und Urbansky, Muthmann, Katz und Reichert (2012)) addiert für jede Zelle *cell* und für jeden im Dokument D vorkommenden Term t die Wahrscheinlichkeiten, dass der Term in der jeweiligen Zelle vorkommt. Abschließend wird, wie in Formel 4.9 gezeigt, mit focus(D) jene Zelle mit der höchsten Wahrscheinlichkeit als Vorhersage gewählt.

83 Für eine Zellgröße von 0,1° beispielsweise beträgt die Seitenlänge am Äquator ca. 11,13 km.

$$\mathrm{p}(cell \mid D) = \sum_{t \in \mathrm{preprocess}(D)} \frac{\mathrm{count}(t, cell)}{\mathrm{count}(t)} \tag{4.8}$$

$$\mathrm{focus}(D) = \arg \max_{cell \in Cells} \mathrm{p}(cell \mid D) \tag{4.9}$$

Die zweite hier verwendete Klassifikationsfunktion basiert auf einem Naïve-Bayes-Klassifikator (siehe Abschnitt 2.3.1), bei dem die von Rennie et al. (2003) vorgeschlagene Maßnahme der „komplementären Klassen“ umgesetzt wurde. Hierbei wird das Scoring bei jeder Klasse nicht durch die Wahrscheinlichkeitswerte in der Klasse selbst, sondern durch die übrigen Wahrscheinlichkeitswerte ermittelt wie in Formel 4.10 dargestellt. Sinngemäß wird hier also nicht die Kategorie mit der höchsten Wahrscheinlichkeit als Klassifikationsergebnis bestimmt, sondern jene, bei der die komplementäre Wahrscheinlichkeit $\mathrm{p}(\neg\, cell \mid D)$ am geringsten ist (Formel 4.11). Hiermit wird im Hinblick auf jede Zelle eine stärkere Gleichverteilung der verwendeten Trainingsdaten erzielt, was nach Rennie et al. (2003) die Klassifikationsgenauigkeit steigert. $\mathrm{count}(t, \neg\, cell)$ gibt hier an, wie häufig t in sämtlichen Zellen außer *cell* vorkommt. Das Scoring beinhaltet ein Laplace-Smoothing (Annahme, dass jeder Term einmal mehr als tatsächlich vorkam) und die einzelnen Terme werden mittels TF-IDF (siehe Abschnitt 2.2) gewichtet.

$$\mathrm{p}(\neg\, cell \mid D) = \sum_{t \in \mathrm{preprocess}(D)} \mathrm{tfidf}(t) \log \frac{\mathrm{count}(t, \neg\, cell) + 1}{\mathrm{termCount}(\neg\, cell) + |\, Terms\,|} \tag{4.10}$$

$$\mathrm{focus}(D) = \arg \max_{cell \in Cells} \log \mathrm{p}(cell) - \mathrm{p}(\neg\, cell \mid D) \tag{4.11}$$

Die A-priori-Wahrscheinlichkeit für eine Zelle ermittelt sich aus der letzten Zeile des Dictionarys in Tabelle 4.3 gemäß Formel 4.12.

$$\mathrm{p}(cell) = \frac{\mathrm{docCount}(cell)}{\sum_{c \in Cells} \mathrm{docCount}(c)} \tag{4.12}$$

Zur Rücktransformation der Zellidentifikatoren zu geographischen Koordinaten könnte jeweils der Mittelpunkt der Zelle bestimmt werden. Je gröber das Raster, desto stärker sind jedoch die Abweichungen einer Koordinate nach Hin- und Rücktransformation. Bei nicht gleich verteilten Koordinaten innerhalb einer Zelle (beispielsweise, weil die Zelle eine Küstenlinie mit sowohl Land als auch Meer beinhaltet), sind die Abweichungen, die bei Wahl des Zellmittelpunkts entstehen, vergleichbar stark. Aufgrund dessen wird während der Trainingsphase aus den tatsächlich vorkommenden Koordinaten in jeder Zelle der geometrische Median (siehe Abschnitt 2.1.4) bestimmt und jener für die Rücktransformation von Zellidentifikator zu Geokoordinate verwendet. Damit werden die räumlichen Abweichungen zwischen repräsentativer Zellkoordinate und den tatsächlich in den Zellen vorkommenden Koordinaten verringert.

Intuitiv steigt die mögliche Vorhersagegenauigkeit, je feiner das Raster gewählt wird. Bei konstanter Dokumentmenge entsteht jedoch mit zunehmend feiner gewähltem Raster das Problem, dass die einzelnen Zellen nur noch wenige Dokumente und somit eine geringe Anzahl von Features enthalten. Durch die dünnbesetzten Featuremengen einzelner Zellen entsteht letztlich genau der gegenteilige

Effekt und die Genauigkeit bei der Fokusbestimmung sinkt, da räumlich nah beieinander liegende Features auf verschiedene Zellen verteilt sind und somit die Wahrscheinlichkeit steigt, dass falsche Zellen klassifiziert werden. Diesem Problem kann mit einer Vergrößerung der Dokumentmenge zum Training begegnet werden. Eine andere Strategie, die mit gleicher Trainingsmenge auskommt, wird nachfolgend vorgestellt.

4.4.2 MEHRSTUFIGE DICTIONARY-BASIERTE STRATEGIE

Die mehrstufige Strategie zur Fokusbestimmung baut auf der vorangehend beschriebenen Dictionary-basierten Strategie (Abschnitt 4.4.1) auf. Hier werden zur Klassifikation jedoch mehrere Raster $S = \{s_1, \ldots, s_n\}$ eingesetzt, deren Zellgröße während der Klassifikation sukzessive verringert wird. Auf diese Weise soll verhindert werden, dass die Fokusvorhersage in die falsche Richtung läuft. Die groben Rasterzellen enthalten vergleichsweise viele Daten (entsprechend der Summe einer Spalte in Tabelle 4.3), womit eine akkurate, jedoch unpräzise Vorhersage getroffen werden kann. Das heißt, es werden tendenziell zwar meist die korrekten Zellen klassifiziert, aufgrund der groben Rastergröße weicht jedoch der vorhergesagte Punkt stark von der tatsächlich korrekten Fokuskoordinate ab.

Die fein aufgelösten Zellen hingegen erlauben eine präzisere Prädiktion, enthalten jedoch jeweils weniger Daten und die Wahrscheinlichkeit, eine falsche Zelle auszuwählen, die weit von der tatsächlich korrekten Koordinate entfernt liegt, ist höher. Die hier vorgestellte Strategie nimmt, wie

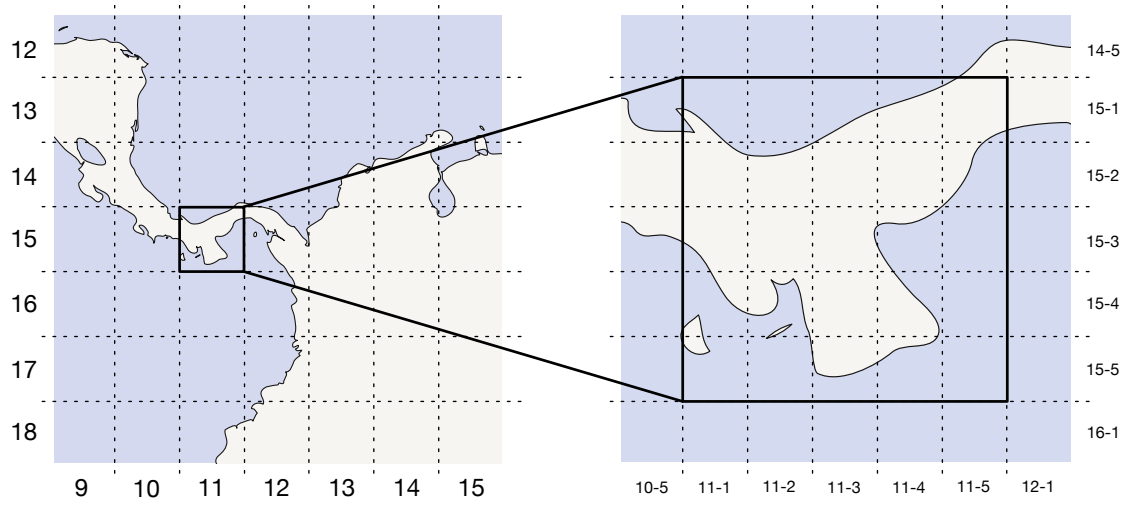


Abbildung 4.2: Schematische Darstellung des zweistufigen Rasters für $m = 5$

in Abbildung 4.2 dargestellt, zunächst eine vergleichsweise grobe Vorhersage vor, die schrittweise präzisiert wird. Nachfolgende Klassifikationsschritte beschränken sich jeweils auf die im vorangegangenen Schritt klassifizierte Region. Der algorithmische Ablauf der Klassifikation ist in Abbildung 4.3

dargestellt. Um zu garantieren, dass eine feinere Rasterzelle jeweils in genau einer gröberen liegt, wird vorausgesetzt, dass gilt $s_k \geq m \cdot s_{k+1}$ mit $m \in \mathbb{N} \wedge m \geq 2$.

Mit der erläuterten Problematik setzen sich bereits verwandte Arbeiten auseinander, die hier beschriebene Idee stellt jedoch eine erheblich einfachere Lösung für das Problem dar. So müssen nach der von Laere et al. (2010a) beschriebenen Vorgehensweise mehrere Clusterings unterschiedlicher Granularitäten vorgenommen und dann jeweils Klassifikatoren trainiert werden. Roller et al. (2012) schlägt vor, adaptive Raster zu verwenden, bei denen Zellen je nach Anzahl darin vorhandener Traingsdokumente aufgeteilt werden.

Der hier vorgestellte Ansatz benötigt nur ein einziges Dictionary als Modell, welches nach der feinsten vorgegebenen Zellgröße aus $s_{min} = \min(S)$ trainiert wird. Die Dictionarys nach gröberem Raster können, sofern die oben genannte Bedingung erfüllt ist, während der Klassifikationsphase durch das feinere Raster simuliert werden, indem die Vorkommenshäufigkeiten der feineren Zellen (großes schwarzes Quadrat in Abbildung 4.2) jeweils zusammenaddiert werden. Dies geschieht in Abbildung 4.3 mittels simulateGridSize(*Dictionary*, s). Im Gegensatz zu Roller et al. (2012) steht hier also initial ein vollständiges feines Dictionary zur Verfügung, aus dem nondestruktiv gröbere Varianten mit vielfacher Rasterzellengröße simuliert werden können.

```
funct getFocusMultistep(D, S, Dictionary) ≡
  cell := null;
  for s in S do                                          // Begin with coarse, go to fine.
    currentDictionary := simulateGridSize(Dictionary, s);
    CellsByProbability := classifyCells(D, currentDictionary);
    if cell = null
      then cell := CellsByProbability[0];          // First iteration, accept cell with highest probability.
       else
          for c in CellsByProbability do            // Iterate cells ordered by descending probability.
            if c ∈ cell                          // Accept only cells within cell from previous iteration.
              then cell := c; break; fi
          end
    fi
  end
  (φ, λ) := transformCellToCoordinate(cell);              // Create latitude/longitude pair from cell.
  return (φ, λ).
```

Abbildung 4.3: Pseudocode für die mehrstufige Dictionary-basierte Strategie

4.4.3 K-NN-BASIERTE STRATEGIE

Diese Stategie verwendet einen k-NN-Klassifikator (k-Nearest-Neighbor). k-NN-Klassifikation gehört zu der Kategorie des „Lazy Learnings“. Hierbei wird während des Trainings kein deskriptives oder generatives Modell im Sinne klassischer Machine-Learning-Verfahren (siehe Abschnitt 2.3)

erzeugt, sondern es werden die Featurevektoren sämtlicher Trainingsdokumente und deren Geokoordinaten in einem Modell gespeichert. Zur Klassifikation eines Textdokuments D werden wie in Formel 4.13 dargestellt die k ähnlichsten Dokumente im Modell *Index* bestimmt und als Fokuslokation der geometrische Median (siehe Abschnitt 2.1.4) aus den zu den Dokumenten zugehörigen Fokuskoordinaten ermittelt.

$$\text{focus}_{k\text{-}NN}(D) = \text{geoMedian}(\text{findSimilarDocuments}(k, D, \mathit{Index})) \tag{4.13}$$

Zur Ermittlung der ähnlichsten Dokumente bietet sich beispielsweise, wie bei k-NN-Klassifikation meist üblich, die Euklidsche Distanz, $d(D_1, D_2) = \| D_1 - D_2 \|_2$, ermittelt über die Frequenzen der Features an. Im Gegensatz zu den vorangehend beschriebenen Strategien muss hier keine Transformation der Koordinaten in ein Raster vorgenommen werden, da die Geo-Koordinaten direkt mit den Dokumenten im Index gespeichert und später bei der Prädiktion mittels Geo-Median-Aggregationsfunktion zur vorhergesagten Fokuslokation kombiniert werden.

4.5 REALISIERUNG UND OPTIMIERUNG

Nachfolgend werden verschiedene der vorgestellten Strategien zur Fokusbestimmung optimiert. Für die Ranking-basierten Strategien (Abschnitt 4.2 und 4.3) wird der in Abschnitt 3.9 vorgestellte Machine-Learning-basierte Extraktor verwendet. Als Datenset wird TUD-Loc-2013 (siehe Abschnitt 3.4.2 und Abschnitt 3.11) verwendet, als zweites Datenset kommen 40.576 Artikel aus der englischsprachigen Wikipedia zum Einsatz. Dazu wurden zufällige Wikipedia-Artikel[84] via MediaWiki-API[85] und Palladian-MediaWiki-Parser (Urbansky, Muthmann, Katz und Reichert, 2012) abgerufen und jene gesammelt, die über ein `coord`-Tag mit dem `display`-Attribut `title` oder `t` verfügen (siehe Abschnitt 3.6). Solche Koordinaten werden auf der entsprechenden Wikipedia-Seite im oberen rechten Rand angezeigt. Ignoriert wurden solche Artikel, deren Titel mit „List of“ beginnt, da dies Auflistungen mehrerer Entitäten sind, die in der Regel über keinen klaren lokalen Fokus verfügen, auch wenn eine Koordinate vorhanden ist[86]. Das `coord`-Tag drückt den geographischen Fokus eines Artikels aus, somit eignet sich das zusätzliche Datenset sehr gut, um die Fokusbestimmung in größerem Rahmen zu evaluieren als dies mit den 152 Dokumenten in TUD-Loc-2013 möglich wäre.

Aus den Texten der Wikipedia wurde der gesamte MediaWiki-Markup[87] entfernt, sodass diese jeweils im Wesentlichen dem Artikelinhalt entsprechen, der beim Aufruf im Webbrowser dargestellt wird. Elemente wie Tabellen, Bildunterschriften und Referenzen wurden komplett entfernt. Der Wikipedia-Datensatz wurde ebenfalls im 40:20:40-Verhältnis in disjunkte Trainings-, Validierungs- und Testsets aufgeteilt (zweimal 16.231, einmal 8.114 Dokumente).

84 `http://en.wikipedia.org/wiki/Wikipedia:Random`

85 `http://www.mediawiki.org/wiki/API:Main_page`

86 Beispiel für einen „List of“-Wikipedia-Artikel mit `coord`-Tags, aber ohne klaren geographischen Fokus: `http://en.wikipedia.org/wiki/List_of_shipwrecks_of_the_United_States`

87 `http://en.wikipedia.org/wiki/Help:Wiki_markup`

4.5.1 OPTIMIERUNG DER MACHINE-LEARNING-BASIERTEN STRATEGIE

Training und Feintuning der featurebasierten Fokusbestimmung fand mit Trainings- und Validierungssets aus TUD-Loc-2013 statt. Zur Lokationsextraktion wurde die in Abschnitt 3.9 beschriebene Machine-Learning-basierte Extraktionsmethode verwendet. Experimente zeigten, dass mit einem geringeren Threshold als den Abschnitt 3.13 ermittelten 0,2 bei der Fokusbestimmung bessere Ergebnisse erzielt werden. Dies liegt daran, dass sich hier ein höherer Recall zur Extraktion von Fokuskandidaten vorteilhafter als eine hohe Precision erweist, da falsche Extraktionen aus dem vorangehenden Verarbeitungsschritt hier nach wie vor noch eliminiert werden können. Dementsprechend wurde für die nachfolgenden Betrachtungen der Wahrscheinlichkeitsschwellwert auf Null gesetzt, um für die Extraktion einen maximalen Recall zu erreichen.

Verwendet wurde wie bereits für die Disambiguierung ein mittels QuickDT trainierter Random-Forest-Klassifikator (siehe Abschnitt 2.3.3). Auch hier wurde eine Backward Feature Elmininationn (siehe Abschnitt 2.3.4) vorgenommen und die Featurekombination mit dem besten F1-Wert bestimmt. Diese umfasst von den ursprünglich angegebenen 27 Features (siehe Tabelle 4.2) die in Abbildung 4.4 aufgelisteten 13 Features.

1. `midpointDistance`
2. `disambiguationTrust`
3. `hierarchyDepth`
4. `descendantCount`
5. `offsetFirst`
6. `childPercentage`
7. `ancestorPercentage`
8. `offsetSpread`
9. `normalizedHierarchyDepth`
10. `populationNorm`
11. `occurrenceFrequency`
12. `occurrenceCount`
13. `offsetLast`

Abbildung 4.4: Ausgewählte Features für die Machine-Learning-basierte Strategie nach Backward Feature Elimination unter Verwendung des Trainings- und Validierungssets von TUD-Loc-2013

Mit dem Ziel, die Aussagekraft der einzelnen Features isoliert zu quantifizieren, wurde unter Verwendung des Trainings- und Validierungssets auch der Information Gain ermittelt. Das Ranking der einzelnen Features dazu zeigt Abbildung 4.5. Als „gute“ Features für die Klassifikation zeigen sich hier an der ersten Stelle die Auftrittshäufigkeit `occurrenceFrequency`, gefolgt von den Positionsfeatures `offsetFirst` und `offsetSpread`, die das erste Vorkommen bzw. den Abstand zwischen erstem und letztem Vorkommen im Text ausdrücken. Die Features, die die Distanzen vom Mittelpunkt sämtlicher Lokationen L finden sich an vierter und fünfter Stelle. An sechster Position rangiert das Feature `occurrenceCount`, welches die Vorkommenshäufigkeit der jeweiligen Lokation im Text angibt. Eine Reihe der vorgestellten Features haben für die Klassifikation keinen Wert, diese weisen in der Abbildung einen Information Gain von Null auf. Bei genauer Betrachtung fällt auf, dass das mit Null gerankte Feature `ancestorPercentage` in der mittels Backward Feature Elimination bestimmten Featuremenge (siehe Abbildung 4.4) vorhanden ist. Die Ursache ist darin zu suchen, dass

durch die schrittweise Feature-Eliminierung nicht zwangsläufig die kleinste mögliche Featuremenge ermittelt wird.

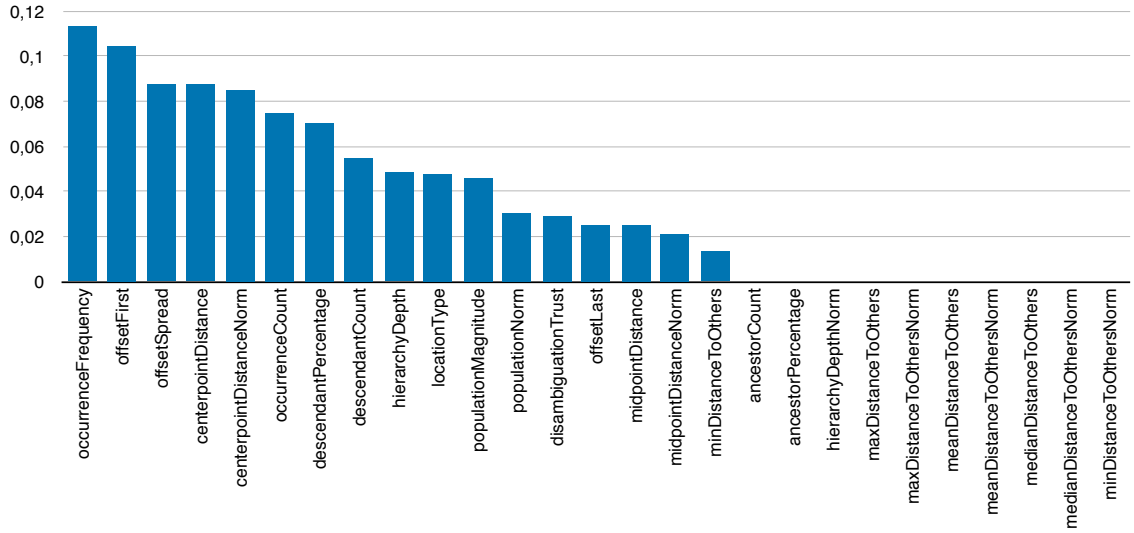


Abbildung 4.5: Information Gain der vorgestellten Features ermittelt auf dem TUD-Loc-2013 Trainings- und Validierungsset

4.5.2 Optimierung der Dictionary-basierten Strategien

Die auf Textklassifikation basierenden Strategien benötigen zum Aufbau des notwendigen Dictionarys eine große Menge an Trainingsdaten. Das Datenset TUD-Loc-2013 stellt davon nicht genügend zur Verfügung, weshalb die Optimierung der nachfolgenden Strategien auf dem Subset der Wikipedia erfolgt. Für die Optimierung wird das arithmetische Mittel (nachfolgend mit $\overline{d}$ abgekürzt) und der Median[88] (nachfolgend mit $\widetilde{d}$ abgekürzt) der Fehlerdistanzen jedes Dokuments, also die Abweichung zwischen Referenzkoordinate und durch die Strategien ermittelte Koordinate in Kilometern bestimmt.

Die Realisierung der Dictionary-basierten Strategien (siehe Abschnitte 4.4.1 und 4.4.2) erfolgte auf Basis des Textklassifikators aus dem Palladian-Toolkit (Urbansky, Muthmann, Katz und Reichert, 2012). Der vorhandene Textklassifikator wurde für die Anforderungen der Fokusbestimmung an vielen Stellen grundlegend erweitert. Zusätzlich zu der bereits vorhandenen Scoring-Formel (beschrieben in Formel 4.8 und 4.9) wurden die beschriebenen Konzepte des komplementären Naïve-Bayes-Scorings (Formel 4.10 und 4.11) implementiert. Da der Textklassifikator bis dato nicht auf große Trainingsmengen (in Bezug auf die Anzahl der Dokumente und der Klassen) optimiert war, mussten außerdem Speicheroptimierungen in den Datenstrukturen des Dictionarys vorgenom-

88 Da hier Distanzen summiert werden, ist der Median im eigentliche Sinne gemeint, nicht der in Abschnitt 2.1.4 eingeführte geographische Median.

men werden. Die vorhandene Implementierung auf Basis von Hashtabellen wurde im Zuge dessen auf eine Trie-Struktur (de la Briandais, 1959) umgestellt und in weiteren Details optimiert, womit der Speicherverbrauch in der Praxis auf ungefähr 10 % im Vergleich zur Ausgangsbasis reduziert werden konnte (bzw. damit einhergehend die Größe der möglichen Trainingsmengen bei gleichem Speicher auf circa das Zehnfache steigt). Tries erweisen sich im vorliegenden Anwendungsfall als deutlich speichereffizienter im Vergleich zu Hashtabellen, da so identische Präfixe mehrerer Terme (beispielsweise für die Terme „the“, „theme“, „theater“) als gemeinsamer Zweig repräsentiert werden können.

Nachfolgend wird betrachtet, wie sich einerseits unterschiedliche Featurekonfigurationen, andererseits verschiedene Rastergrößen auf das Klassifikationsergebnis auswirken. Außerdem wurden so genannte Lernkurven ermittelt (Abschnitt 4.5.4), die angeben, wie sich eine jeweils zunehmende Menge an Trainingsdaten auf die Klassifikationsqualität auswirkt. Insgesamt waren hierfür mehrere Hundert Trainings- und Validierungsläufe mit dem Wikipedia-Datenset notwendig, was einen beträchtlichen Zeitaufwand mit sich bringt. Um die Optimierung der Parameter zu beschleunigen, wurde die in Abschnitt 4.4.1 in Formel 4.8 und 4.9 beschriebene, einfache Variante des Scorings verwendet. Diese hat einen beträchtlich geringeren Berechnungsaufwand, da deutlich weniger Iterationen zum Scoring notwendig sind. Das Scoring mittels komplementärem Naïve Bayes (Formel 4.10 und 4.11) ist weitaus rechenaufwendiger, weil durch das Laplace-Smoothing auch Term-Kategorie-Einträge mit einem Wert von Null (die durch die Glättung inkrementiert wird) in die Berechnung mit einfließen. Die im Vorfeld gewonnenen Erfahrungen haben gezeigt, dass für beide Scoring-Verfahren jeweils die Optimalwerte mit identischen Featurekonfigurationen erzielt werden. In der abschließenden Evaluierung, in der die einzelnen hier vorgestellten Strategien miteinander verglichen werden, wird das aufwendigere Scoring verwendet.

Ermittlung der optimalen Featurekombination Der Textklassifikator erlaubt unterschiedliche Konfigurationen für die Featureextraktion. Experimentiert wurde mit zeichen- und wortbasierten n-Grammen und Kombinationen unterschiedlicher Länge. Zeichenbasierte n-Gramme wurden für das Intervall $n = [1, 10]$ extrahiert, wortbasierte n-Gramme für die Längen $n = [1, 5]$. Außerdem wurden jeweils Kombinationen der Länge $[n, m]$, also der Vereinigungsmenge aus n- bis m-Grammen, generiert. Unter Verwendung der Scoring-Funktion aus Formel 4.8 und 4.9, einer initial festgelegten Rastergröße von 5° und des Trainings- und Validierungssets aus der Wikipedia wurde aus 70 möglichen Featurekombinationen die beste Zusammenstellung ermittelt. Die besten Ergebnisse wurden mit zeichenbasierten 6-9-Grammen erzielt, der Fehler im Validierungsset beträgt nach arithmetischem Mittel hier 1.022,72 km, der Median 180,78 km. Die Ergebnisse sind detailliert in den Tabellen B.1 und B.2 zu finden. Die nachfolgenden Betrachtungen bezüglich der Dictionary-basierten Fokusbestimmung nutzen durchweg diese Featurekombination.

Betrachtung unterschiedlicher Rastergrößen Im nächsten Schritt wurde der Fehler nach Median und arithmetischem Mittel im Hinblick auf die unterschiedliche Zellgrößen s untersucht. Eine

Startgröße von 180° wurde dazu mit jedem Schritt halbiert, womit sich die Anzahl der Zellen jeweils vervierfacht. Der geringste Fehler nach arithmetischem Mittel mit 1.028,78 km wird mit einer Zellgröße von 5,63° erreicht, der Median-Fehler beträgt hier 199,19 km. Nach Median hingegen wird die geringste Fehlerdistanz mit 61,52 km mit einer Zellgröße von 0,7° erzielt, wobei hier das arithmetische Mittel für den Fehler bei 1.694,26 km liegt. Die Ergebnisse sind in Abbildung 4.6 dargestellt, eine ausführliche Auflistung der Werte findet sich in Tabelle B.3. Sie verdeutlichen die in Abschnitt 4.4.2 erläuterte Problematik. Bei einem vergleichsweise groben Raster werden insgesamt akkuratere Ergebnisse erzielt, was sich in geringeren Fehlerdistanzen nach arithmetischem Mittel manifestiert. Auf der anderen Seite wird mit einem feineren Raster eher eine höhere Präzision erzielt, welche sich in geringeren Fehlerdistanzen beim Median niederschlägt. Die hier erläuterte Diskrepanz zwischen akkuraten und präzisen Ergebnissen bildet die Motivation für die mehrstufige Strategie.

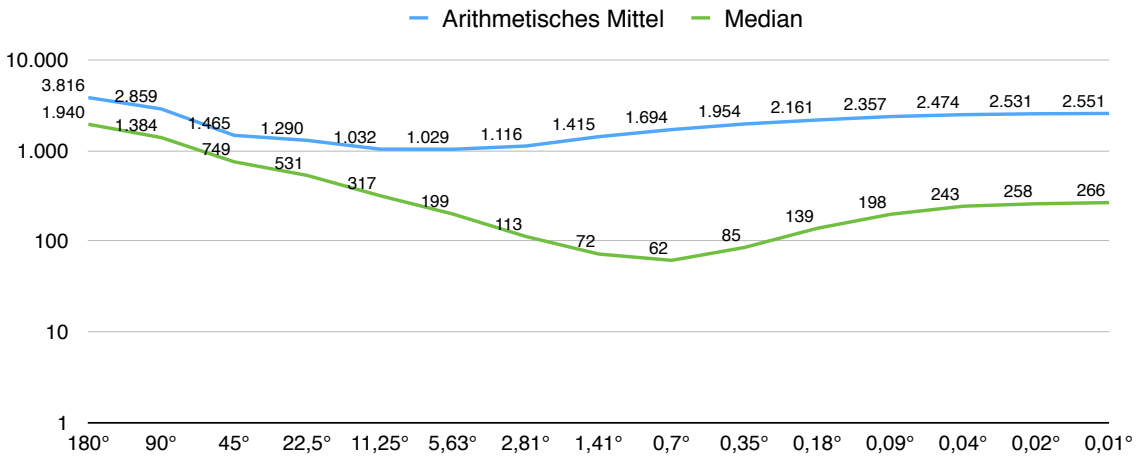


Abbildung 4.6: Arithmetisches Mittel und Median der Fehlerdistanzen in Kilometern bei schrittweise halbierten Zellgrößen mit Dictionary-basierter Strategie und Wikipedia-Datenset (x- und y-Achse sind jeweils logarithmisch skaliert)

Betrachtung von Kombinationen von Rastergrößen Der vorangehende Abschnitt hat gezeigt, dass die erzielten Optima bezüglich Fehlerabweichungen für das arithmetische Mittel und den Median bei unterschiedlichen Größen für die jeweils gewählten Raster liegen (siehe Abbildung 4.6). Während bei dem verwendeten Datenset nach Median der minimale Fehler bei einer Rastergröße von 0,7° erreicht wird, wird der geringste Fehler nach arithmetischem Mittel bei einer Rastergröße von 5,63° erzielt. Dieser Abschnitt betrachtet, ob und inwieweit unter Verwendung der mehrstufigen Dictionary-basierten Strategie (siehe Abschnitt 4.3) zur Fokusbestimmung diese Ausgangswerte verbessert werden können. Getestet wurden jeweils Kombinationen aus Grob- und Feinraster (sodass die generische mehrstufige Strategie hier konkret zweistufig instanziiert wurde) beginnend bei 180°, welches entsprechend der vorhergegangenen Betrachtung, jeweils bis hin zu $180 \cdot 2^{-14} \approx 0{,}01°$

halbiert wurde. Somit wurden insgesamt $\binom{14}{2} = 91$ Kombinationen aus grobem und feinem Raster evaluiert.

Als intuitive Wahl für eine Rasterkombination mag bei Betrachtung der Daten aus der vorhergehenden Untersuchung $S = \{5{,}63°, 0{,}7°\}$ erscheinen, also die Kombination der Zellgrößen, die bei einstufiger Strategie nach arithmetischem Mittel und Median den geringsten Fehler produzieren. Diese Kombination verursacht nach Median eine Abweichung von 57,25 km, nach arithmetischem Mittel eine Abweichung von 948,62 km. Wie Tabelle B.4 zu entnehmen, die die einzelnen Ergebnisse im Detail zeigt, kann aber noch eine weitere Verbesserung erzielt werden. Der minimale Fehler nach Median (46,65 km) wird mit der Kombination $S = \{2{,}81°, 0{,}35°\}$, der geringste Fehler nach arithmetischem Mittel (869,38 km) mit $S = \{11.25°, 1.41°\}$ erreicht. Tabelle 4.4 zeigt die erzielten Verbesserungen durch die mehrstufige im Vergleich zur normalen Dictionary-basierten Strategie.

	Fehler in km	
	$\overline{d}$	$\widetilde{d}$
Dictionary (min. $\overline{d}$; $s = 5{,}63°$)	1.028,78	199,19
Dictionary (min. $\widetilde{d}$; $s = 0{,}7°$)	1.694,26	61,52
Dictionary mehrst. ($S = \{5{,}63°, 0{,}7°\}$)	948,62	57,25
Dictionary mehrst. ($S = \{2{,}81°, 0{,}35°\}$)	1.084,65	**46,65**
Dictionary mehrst. ($S = \{11{,}25°, 1{,}41°\}$)	**869,38**	72,18

Tabelle 4.4: Vergleich für Fehler nach arithmetischem Mittel $\overline{d}$ und Median-Fehler $\widetilde{d}$ zwischen Dictionary-basierter und mehrstufiger Dictionary-basierter Strategie

4.5.3 Optimierung der k-NN-basierten Strategie

Die k-NN-basierte Strategie (siehe Abschnitt 4.4.3) nutzt Apache Lucene[89] als technologische Grundlage. Während der Trainingsphase werden sämtliche Dokumente in einem Lucene-Index gespeichert. Die in Formel 4.13 dargestellte Funktion $\mathsf{findSimilarDocuments}(k, D)$ wird durch eine `MoreLikeThis`-Suche[90] realisiert, welche zu dem gegebenen Dokument D eine gerankte Liste mit ähnlichen Dokumenten im Index findet. Als Ähnlichkeitsmaß wird die standardmäßige `DefaultSimilarity`[91] von Lucene verwendet[92].

89 `https://lucene.apache.org`

90 `https://lucene.apache.org/core/4_7_2/queries/org/apache/lucene/queries/mlt/MoreLikeThis.html`

91 `https://lucene.apache.org/core/4_7_2/core/org/apache/lucene/search/similarities/DefaultSimilarity.html`

92 Die verwendete Implementierung nutzt eine Reihe weiterer Faktoren, sodass hier streng genommen keine Euklidsche Distanz vorliegt, wie in Abschnitt 4.4.3 vorgeschlagen. Experimente haben aber gezeigt, dass in diesem Fall die Lucene-Implementierung der Euklidschen Distanz überlegen ist.

Ermittlung der optimalen Featurekombination Zur Bestimmung der optimalen Featurekombination für die k-NN-basierte Strategie (siehe Abschnitt 4.4.3) wurde analog vorgegangen wie in Abschnitt 4.5.2 beschrieben. k (also die Anzahl ähnlicher Dokumente, die zur Koordinatenermittlung berücksichtigt werden) wurde zunächst auf 1 festgelegt, die Betrachtung unterschiedlicher Werte für k erfolgt im nächsten Schritt. Von den 70 betrachteten Featurekombinationen wurde mit wortbasierten 1-Grammen sowohl im Hinblick auf das arithmetische Mittel als auch den Median-Fehler das beste Ergebnis erzielt. Ersterer betrug 1.226,6 km, letzterer 101,65 km. Die Ergebnisse sind aufgeschlüsselt für jede evaluierte Featurekombination in Tabelle B.5 und B.6 zu finden.

Betrachtung des Parameters k In Abbildung 4.7 ist die Entwicklung des Fehlers nach Median und arithmetischem Mittel bei Werten für $k = [1, 10]$ dargestellt. Während die Werte für den Median bei Erhöhung von k vergleichsweise konstant bleibt, sinkt der Fehler nach arithmetischem Mittel bei Erhöhung von k deutlich. Bemerkenswert ist die Tatsache, dass die stärkste Reduktion im Hinblick auf $\overline{d}$ beim Schritt von $k = 2$ zu $k = 3$ erzielt wird. Dies liegt an der Aggregationsmethode der ermittelten Koordinaten: Der geometrische Median (siehe Abschnitt 2.1.4), mit dem das Zentrum minimaler Distanz ermittelt wird, entspricht bei $k = 2$ Dokumenten (und somit Koordinatenpaaren) dem einfachen räumlichen Mittelpunkt (Abschnitt 2.1.3). Für $k \geq 3$ hingegen ändert sich das Verhalten des geometrischen Medians dahingehend, dass bei Mengen mit Koordinatenpaaren Ausreißer „eliminiert“ werden: Bei drei Koordinatenpaaren beispielsweise, von denen zwei relativ nahe beieinander und eines deutlich entfernt liegt, befindet sich das Zentrum minimaler Distanz bei den beiden beieinanderliegenden Koordinaten (siehe Abbildung 2.2). Übertragen auf die vorliegende k-NN-Strategie bedeutet dies, dass etwaige Fehlprädiktionen gut korrigiert werden können. Der minimale Median-Fehler mit 101,41 km wird bei den abgebildeten Daten bei $k = 1$, der minimale Fehler nach arithmetischem Mittel in Höhe von 963,48 km bei $k = 6$ erzielt. Für den nachfolgenden Vergleich wird $k = 5$ festgelegt, da für größere Werte von k der Fehler nach arithmetischem Mittel weniger stark fällt, der Fehler nach Median jedoch stetig zunimmt.

4.5.4 EINFLUSS DER GRÖSSE DES TRAININGSSETS

Nachfolgend wird untersucht, wie sich die Menge an Trainingsdaten auf die Qualität der Fokusbestimmung auswirkt. Dazu wurden die Dictionary- und die k-NN-basierte Strategie mit jeweils zunehmender Menge an Trianingsdokumenten trainiert und mit dem Validierungsset getestet. Die Menge wurde, beginnend mit einem Dokument, jeweils schrittweise verdoppelt bis jeweils die gesamte Trainingsmenge von 16.231 Dokumenten ausgeschöpft war (der letzte Schritt entspricht somit nicht ganz einer Verdopplung). Insgesamt wurde die beschriebene Prozedur zehnmal mit zufällig ausgewählten Teilmenge zum Training wiederholt, um statistische Schwankungen auszugleichen. Die dargestellten Werte stellen somit den gemittelten Wert aus den zehn Durchläufen dar.

Die in Abbildung 4.8 und 4.9 dargestellten Resultate zeigen, dass sowohl bei der k-NN-basierten als auch bei der Dictionary-basierten-Strategie mit Vergrößerung der Trainingsmenge kontinuierlich

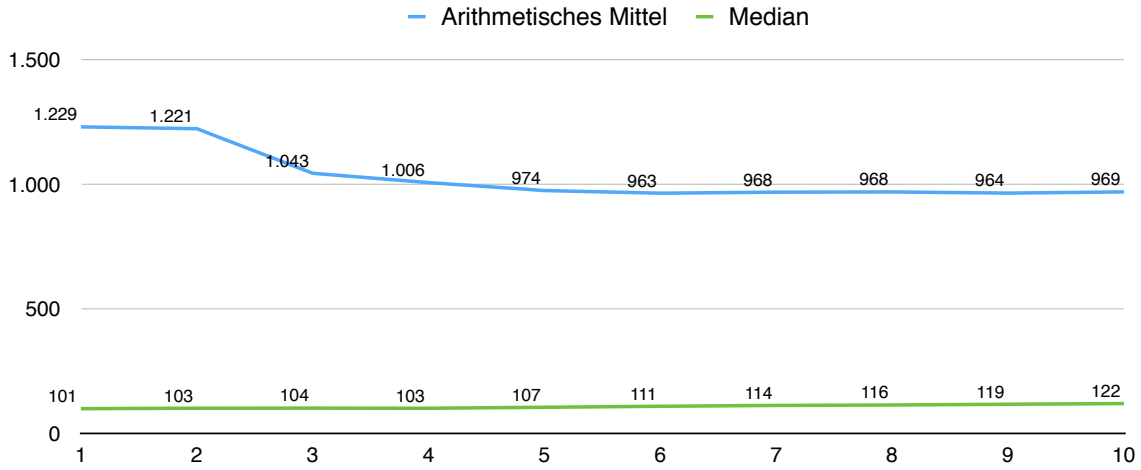


Abbildung 4.7: Arithmetisches Mittel und Median der Fehlerdistanzen in Kilometern bei k-NN-basierter Strategie und Wikipedia-Datenset mit variierten Werten für k

bessere Resultate bezüglich der Abweichung nach arithmetischem Mittel und Median erzielt werden. Im Verlauf flacht die durch Verdopplung der Daten erzielte Verbesserung im Median etwas ab, das arithmetische Mittel fällt jedoch im Hinblick auf den Median noch vergleichbar stark. Die Resultate zeigen, dass auch bei einer weiteren Vergrößerung des Trainingssets über die 16.231 Dokumente hinaus weitere Verbesserungen bezüglich der Vorhersagegenauigkeit zu erwarten sind.

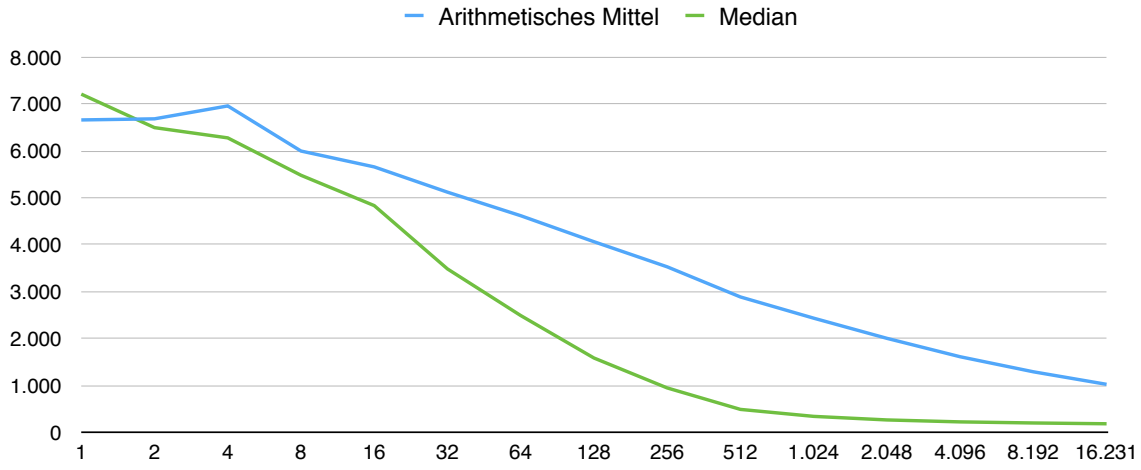


Abbildung 4.8: Entwicklung des arithmetischen Mittels und Medians der Fehlerdistanzen in Kilometern bei Dictionary-basierter Strategie und Wikipedia-Datenset bei Verdopplung der Trainingsmenge mit jedem Schritt

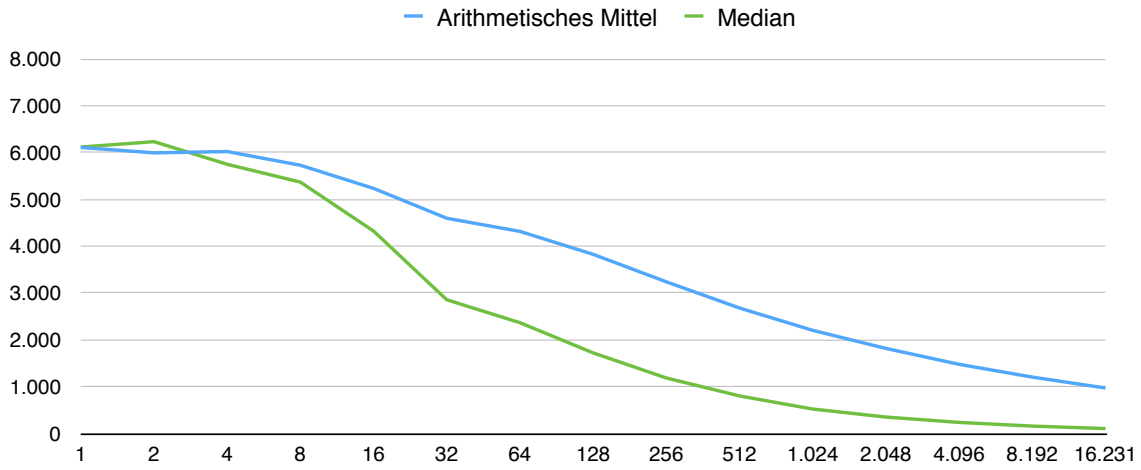


Abbildung 4.9: Entwicklung des arithmetischen Mittels und Medians der Fehlerdistanzen in Kilometern bei k-NN-basierter Strategie und Wikipedia-Datenset bei Verdopplung der Trainingsmenge mit jedem Schritt

4.6 VERGLEICH

Dieses Kapitel vergleicht die vorgestellten Strategien zur Fokusbestimmung miteinander. Diese nutzen jeweils die im vorherigen Abschnitt 4.5 ermittelten Optimalkonfigurationen. Zusätzlich zu den vorangehend bereits ermittelten Fehlerdistanzen, für die entsprechend der Vorgehensweise in Abschnitt 4.5.2 das arithmetische Mittel und der Median in Kilometern ermittelt wird, wird nachfolgend noch die Wahrscheinlichkeit mit Fehlern von höchstens x Kilometern $p(d \leq x)$ für die Werte von $x \in \{1, 10, 100, 1.000\}$ angegeben. Damit wird ein Überblick über die jeweiligen Fehlerintervalle erleichtert, indem abgelesen werden kann, wie viel Prozent der Dokumente beispielsweise mit einem Fehler von höchsten 100 km lokalisiert wurden. Dokumente im Datenset TUD-Loc-2013, in jeweils denen keine Referenz-Fokuslokation vorhanden ist, werden übersprungen. Im Falle, dass eines der hier vorgestellten Verfahren keine Fokus-Lokation bzw. dazugehörige Koordinaten extrahiert, wird ein maximaler Fehler von 20.037,58 km angenommen (entsprechend der maximal möglichen Entfernung zweier Punkte auf der Erdoberfläche ausgehend von einem maximalen Erdumfang von 40.075,16 km am Äquator).

Diese Bewertung ist bewusst streng gewählt, um Extraktionen, bei denen kein Ergebnis erzielt wurde, stark zu bestrafen. In diesem Fall hat der maximal angenommene Fehler von 20.037,58 km einen starken Einfluss auf den ermittelten Fehler nach arithmetischem Mittel. Die hier verwendeten Strategien könnten die erzielten Ergebnisse beispielsweise schönen, indem in jenen Fällen, in denen per Algorithmus keine Fokuslokation bestimmt werden kann, einfach eine Koordinate per Zufallsprinzip (oder jene mit höchster A-priori-Wahrscheinlicheit) bestimmt wird, womit in solchen Fällen der Fehler im Durchschnitt auf die Hälfte des maximal angenommenen reduziert werden könnte.

Dies widerspricht jedoch den hier zugrunde gelegten Prinzipien eines fairen Vergleichs und wird nicht angewendet.

	$p(d \leq x)$, x in km				Fehler in km	
	1	**10**	**100**	**1.000**	$\overline{d}$	$\widetilde{d}$
Position	54	54	60	74	2.273,89	0,15
Einwohnerzahl	12	12	20	40	3.170,43	1.386,38
Frequenz	48	50	62	76	1.676,84	6,69
Mittelpunkt	12	12	16	58	1.692,95	840,85
Min. Distanz	26	28	44	80	1.213,39	137,08
Trust	24	26	30	60	1.855,64	385,98
Machine Learning	**60**	**64**	**74**	**90**	**835,03**	**0**

Tabelle 4.5: Evaluierungsergebnisse für Fokusbestimmung auf TUD-Loc-2013-Testset

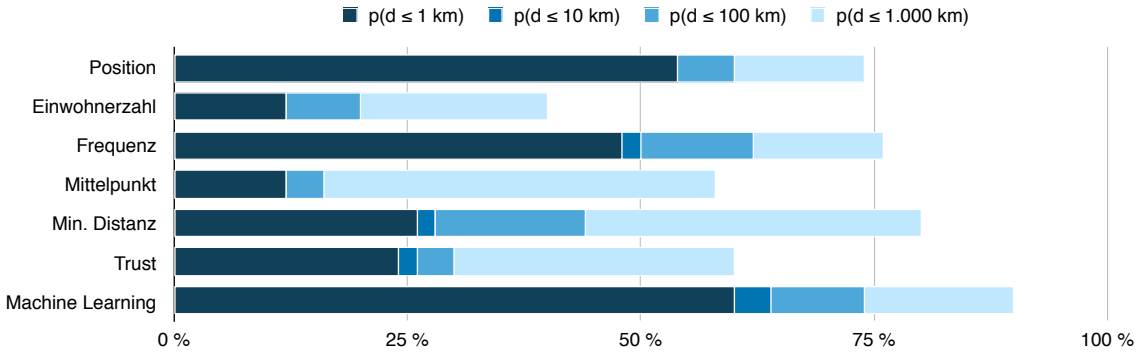


Abbildung 4.10: Kumulierte Fehlerwahrscheinlichkeiten der Ranking-basierten Algorithmen für Fokusbestimmung auf TUD-Loc-2013-Testset

TUD-Loc-2013 Für das Datenset TUD-Loc-2013 wurde keine Evaluierung der Textklassifikations-basierten Methoden vorgenommen, da das Datenset mit insgesamt 152 Dokumenten zu klein ist und hier nicht ausreichend Trainingsdaten zur Verfügung stehen. Die Machine-Learning-basierte Strategie wurde mit dem Trainingsset von TUD-Loc-2013 trainiert. Tabelle 4.5 zeigt die Ergebnisse für die Fokusbestimmung für TUD-Loc-2013. Die Machine-Learning-basierte Strategie schneidet hier in sämtlichen Belangen am besten ab. Der Median-Fehler beträgt 0 km, der Fehler nach arithmetischem Mittel 835,03 km. In Bezug auf den Fehler nach arithmetischem Mittel liegt die Strategie „Minimale Distanz“ mit 1.213,39 km an zweiter Stelle. Nach Median-Fehler hingegen ist die Strategie „Position“ mit 0,15 km zweitplatziert. Abbildung 4.10 stellt die Verteilungen der Fehler dar.

Wikipedia-Datenset In Tabelle 4.6 sind die entsprechenden Ergebnisse für das Wikipedia-Datenset dargestellt. Hier wurden zusätzlich die Strategien mit Textklassifikation evaluiert (unter Verwendung des komplementäre Naïve-Bayes-Scorings in Formel 4.10 und 4.11). Die Ergebnisse

	$p(d \leq x)$, x in km				Fehler in km	
	1	**10**	**100**	**1.000**	$\overline{d}$	$\widetilde{d}$
Position	46,34	62,82	74,23	84,29	1.112,38	1,51
Einwohnerzahl	0,35	3,01	18,4	70,08	1.682,43	358,8
Frequenz	25,33	42,44	60,67	81,73	1.305,28	27,7
Mittelpunkt	9,8	21,84	49,13	83,17	794,32	105,52
Min. Distanz	27,18	51,81	76,66	91,6	542,42	8,7
Trust	6,35	16,31	38,48	82,14	1.002,44	172,79
Machine Learning	**50,59**	**72,4**	**87,26**	**95,01**	**345,49**	**0,95**
Dictionary (s = 0,7°)	1,12	13,64	61,82	82,79	1.153,8	50,4
Dictionary mehrst. ($S = \{5{,}63°, 0{,}7°\}$)	1,12	13,84	63,2	91,74	590,43	50,61
k-NN (k = 5)	1,5	13,49	48,86	81,9	1.009,94	107,49

Tabelle 4.6: Evaluierungsergebnisse für Fokusbestimmung auf Wikipedia-Testset

spiegeln zunächst die Erkenntnisse, welche auch für das TUD-Loc-2013-Datenset gewonnen wurden wieder: Die Machine-Learning-basierte Fokusbestimmung liefert durchweg bessere Resultate als die Heuristiken. Der Fehler nach arithmetischem Mittel liegt hier bei 345,49 km, der Median bei 0,95 km. Die Heuristik „Minimale Distanz“ erzielt den zweitgeringsten Fehler nach arithmetischem Mittel (542,42 km), mit „Position“ wird der zweitbeste Median in Höhe von 1,51 km erreicht.

Vergleichsweise stark abgeschlagen zur Strategie „Machine Learning“ liegen die hier zusätzlich evaluierten Verfahren, die Textklassifikation zur Fokusbestimmung nutzen. Dies mag zunächst zu dem Schluss führen, dass jene Verfahren über keinen praktischen Wert verfügen. Diesen Schluss gilt es allerdings in zweierlei Hinsicht zu relativieren: Die textklassifikationsbasierten Verfahren benötigen keine vorgelagerte Lokationsextraktion und somit keinen Gazetteer im Hintergrund, in dem potentielle Lokationen nachgeschlagen werden müssen. Für einen Beispielsatz wie „We had a ride in the cable cars and went to the bay“ könnte mit den Ranking-basierten Verfahren keine Aussage über den lokalen Fokus getroffen werden, da der Satz keine expliziten Ortsangaben wie „San Francisco“ enthält, welche durch die Lokationsextraktion extrahiert werden können. Die hier vorgestellten Verfahren auf Basis von Textklassifikation können in solchen Situationen ihre Stärken ausspielen, sofern trainierte regionsspezifische Merkmale erkannt werden.

Desweiteren muss bedacht werden, dass die hier verwendete Trainingsmenge von 16.231 Dokumenten immer noch vergleichsweise gering ist. Bei der hier für den Dictionary-basierten Ansatz verwendeten Rastergröße von s = 0,7° entstehen 5.653 nichtleere Zellen, zwischen denen der Textklassifikator unterscheiden muss. Dies entspricht im Durchschnitt also gerade einmal 2,87 Dokumenten pro Rasterzelle, die für das Training zur Verfügung stehen. Dass hier mehr Trainingsdokumente die Resultate weiter verbessern können, wurde bereits in Abschnitt 4.5.4 deutlich. Nachfolgend wird aus diesem Grund noch eine Betrachtung mit deutlich mehr Trainingsdokumenten durchgeführt.

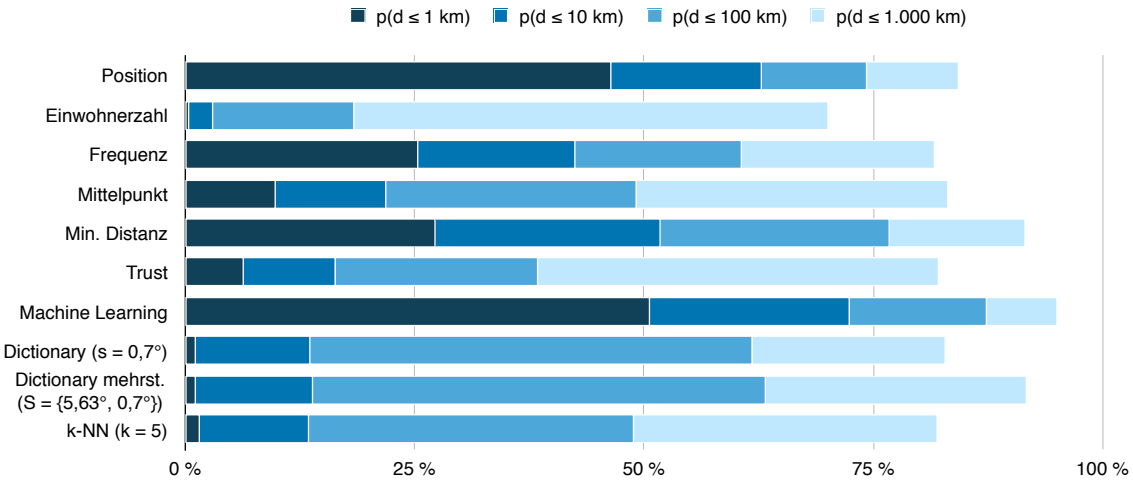


Abbildung 4.11: Kumulierte Fehlerwahrscheinlichkeiten für Fokusbestimmung auf Wikipedia-Datenset

Abbildung 4.11 zeigt die Verteilung der einzelnen Fehlerwahrscheinlichkeiten für das Wikipedia-Datenset. Deutlich wird hier das Phänomen, dass die Genauigkeit unter 1 bzw. 10 km bei den Strategien mit Textklassifikation im Vergleich zu den anderen (vor allem der Strategie „Machine Learning“) extrem zurücksteht. Hier profitieren die Ranking-basierten Verfahren von dem Gazetteer, der genaue Koordinatendaten liefern kann.

Wikipedia komplett Abschließend wurde die Fokusbestimmung mittels Textklassifikation auf der gesamten englischsprachigen Wikipedia evaluiert. Hiermit sind Vergleiche mit der Arbeit von Wing und Baldridge (2011) möglich, deren Experimente ebenfalls auf einem Wikipedia-Dump vorgenommen wurden. Verwendet wird der Dump der englischsprachigen Wikipedia vom 3. Mai 2013[93]. Es galten die gleichen Kriterien wie auch bereits für das in Abschnitt 4.5 beschriebene Wikipedia-Datenset: Betrachtet werden sämtliche Artikel aus dem Haupt-Namespace[94], die über ein `coord`-Tag verfügten welches ein `display`-Attribut mit dem Wert `title` oder `t` hatten. Seiten, deren Namen mit „List of“ begann, wurden hier ebenfalls ignoriert.

Um den Wikipedia-Dump im vorhandenen Evaluierungsframework verarbeiten zu können, wurde das Palladian-Toolkit (Urbansky, Muthmann, Katz und Reichert, 2012) um einen entsprechenden Parser erweitert, der eine direkte Verarbeitung der komprimierten XML-Daten erlaubt, ohne diese dekomprimieren zu müssen (die komprimierte Datei im bzip2-Format ist über 9,8 GB groß). Die Aufteilung in Trainings- und Testset geschieht hier auf Basis der eindeutigen Seiten-IDs im Verhältnis 90:10. Seiten, bei denen die IDs ohne Rest durch zehn teilbar sind, wurden als Test-, die anderen als Trainingsdokumente verwendet. Nach den oben genannten Kriterien standen somit insgesamt

93 `http://dumps.wikimedia.org/backup-index.html`

94 Das heißt, nur Wikipedia-Artikel im eigentlichen Sinne ohne beispielsweise Diskussionsseiten; innerhalb des Dumps haben die Artikelseiten den Namespace-Identifikator 0, siehe `http://en.wikipedia.org/wiki/Wikipedia:Namespace`

560.300 Dokumente mit Koordinaten zur Verfügung, wovon 504.630 auf das Trainings- und 55.670 auf das Testset entfielen.

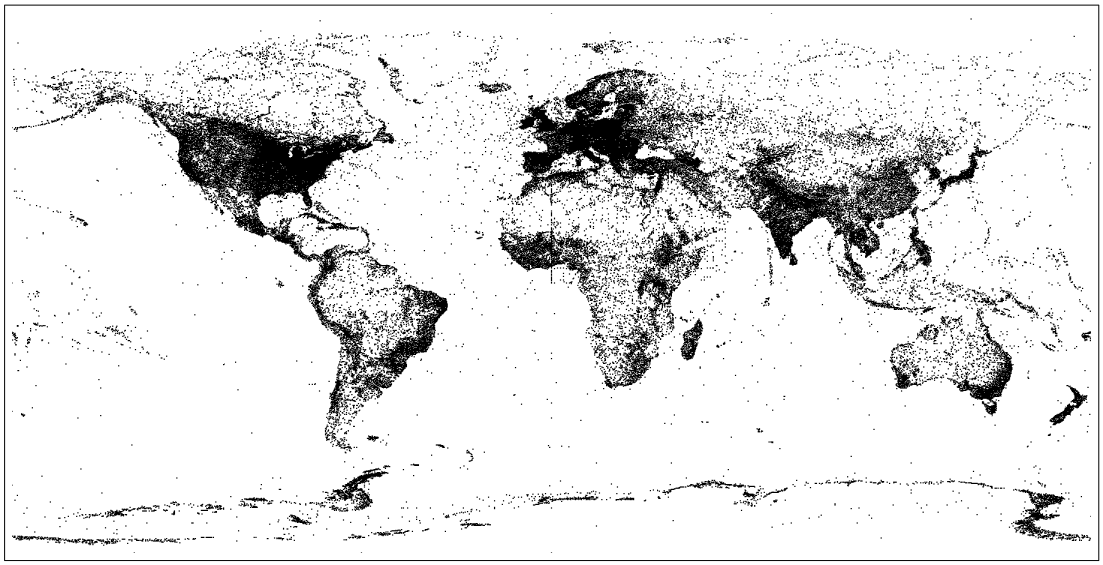


Abbildung 4.12: Belegung der Rasterzellen für Dictionary, welches aus 90 % der Wikipedia-Artikel trainiert wurde; ein schwarzer Punkt entspricht einem vorhandenen Trainingsdokument an der entsprechenden Stelle

In Abbildung 4.12 ist die räumliche Verteilung der Trainingsdokumente visualisiert. Schwarze Punkte in der Abbildung repräsentieren Zellen, in denen Trainingsdokumente vorhanden sind, weiße Flächen hingegen sind unbelegte Zellen. Das Dictionary wurde analog zu Wing und Baldridge (2011) mit einer Rastergröße von 0,1° trainiert und enthält über 35 Mio. unterschiedliche n-Gramme der Länge 6 bis 9. Von den bei der Rastergröße von 0,1° theoretisch möglichen 6,48 Mio. Zellen sind 172.034 nichtleer, was einem Anteil von 2,65 % entspricht. Insgesamt sind in der Term-Kategorie-Matrix des Dictionarys über 267 Mio. Einträge nichtleer. Die durchgeführten Evaluierungen liefen insgesamt über mehrere Tage und es war notwendig, der verwendeten Java-VM einen maximalen Heap-Speicher von 12 GB zuzuweisen.

In Tabelle 4.7 sind die Ergebnisse der Evaluierungen dargestellt. Im Vergleich mit den Ergebnissen, die mit dem kleinen Trainingsset der Wikipedia erzielt wurden (siehe Tabelle 4.6), ist hier durch die große Datenmenge eine enorme Verbesserung der Ergebnisse zu verzeichnen. Die besseren Resultate erzielt die mehrstufige Dictionary-basierte Variante, die einen Median-Fehler von 13,24 km und einen Fehler nach arithmetischem Mittel von 219,21 km produziert. Die k-NN-Strategie verursacht einen minimal höheren Fehler nach Median (14,08 km), jedoch ein deutlich höheres arithmetisches Mittel bezüglich des Fehlers in Höhe von 329,58 km. Die Resultate entsprechen somit weitestgehend denen von Wing und Baldridge (2011), die einen Fehler nach arithmetischem Mittel von 221 km und

einen Median von 11,8 km angeben[95]. Die Angaben zu den kumulierten Fehlerabweichungen fehlen in der Tabelle, da die Autoren dazu leider keine Angaben machen.

	$p(d \leq x)$, x in km				Fehler in km	
	1	**10**	**100**	**1.000**	$\overline{d}$	$\widetilde{d}$
Dictionary mehrst. (s = 0,1°)	4,55	42,49	**84,94**	**97,22**	**219,21**	13,24
k-NN (k = 5)	**8,62**	**43,4**	77,52	93,94	329,58	14,08
Wing und Baldridge (2011)	?	?	?	?	221	**11,8**

Tabelle 4.7: Evaluierungsergebnisse für Fokusbestimmung auf Wikipedia-Dump, zum groben Vergleich Wing und Baldridge (2011), die keine kumulierten Fehlerabweichungen angeben (hier mit ? ausgewiesene Einträge).

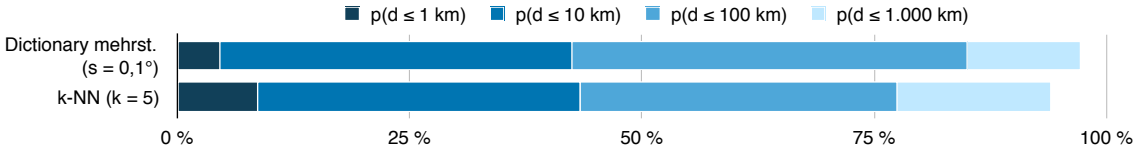


Abbildung 4.13: Kumulierte Fehlerwahrscheinlichkeiten für Fokusbestimmung auf komplettem Wikipedia-Dump

Die Ergebnisse offenbaren außerdem, dass selbst mit der enormen Trainingsmenge die erzielten Vorhersagen weniger akkurat sind, als jene, welche sich mittels Machine-Learning-basiertem Ranking erzielen lassen. Im Vergleich mit den Ergebnissen in Tabelle 4.6, in der beispielsweise die Wahrscheinlichkeit $p(d \leq x)$ für einen Fehler x = 10 km 72,4 % beträgt, liegt diese hier bei deutlich geringeren 42,49 % bzw. 43,4 %. Werden maximale Fehlerdistanzen von x = 1 km gefordert, wird dies mittels Machine-Learning-Methode in 50,59 % der Fälle erreicht, mittels Textklassifikation nur in 4,55 % bzw. 8,62 % der Fälle. Hier offenbaren sich also die Vorteile der genauen Lokationsinformationen, welche im Rahmen der Lokationsextraktion aus dem Gazetteer gewonnen werden können.

Trotz alledem haben die hier vorgestellten Verfahren der Textklassifikation ihre Daseinsberechtigung, gerade im Zusammenhang mit Texten minderer Qualität oder solchen, denen explizite Ortsangaben fehlen. Für zukünftige Untersuchungen erscheinen hier insbesondere kombinierte Verfahren, die die Textklassifikation als Fallback und/oder zusätzliches Feature im Rahmen das Machine-Learning-basierten Rankings nutzen, vielversprechend.

95 Wing und Baldridge (2011) verwendeten einen anderen Datenbestand der Wikipedia vom 4. September 2010, der mittlerweile nicht mehr verfügbar ist. Außerdem werden die Wikipedia-Seiten anders geparst, und beispielsweise Daten aus Infoboxen verarbeitet. Die angegebenen Werte dienen damit nur als grober Richtwert zum Vergleich.

4.7 ZUSAMMENFASSUNG

In diesem Kapitel wurden mehrere Methoden vorgestellt, um den geographischen Fokus von Textdokumenten zu ermitteln. Die einzelnen Strategien gehören den beiden Gruppen „Ranking-basiert“ (Abschnitt 4.2 und 4.3) und „textklassifikationsbasiert“ (Abschnitt 4.4) an. Innerhalb der ersten Gruppe wurden mehrere einfache Heuristiken vorgestellt. Im Zuge der Related-Work-Recherche wurden keine Arbeiten gefunden, die die Fokusbestimmung mittels Machine-Learning-basierter Klassifikation und einer Reihe von lokations- und textspezifischen Features wie sie in Abschnitt 4.3 vorgestellt wurde, durchführen.

Im Rahmen der textklassifikationsbasierten Verfahren wurde eine Dictionary-basierte Strategie diskutiert, welche Koordinaten mittels Raster auf Kategorien abbildet. Diese Strategie ähnelt im Grundsatz der von Wing und Baldridge (2011) präsentierten. Eine deutliche Verbesserung konnte durch die darauf aufbauende, in Abschnitt 4.4.2 vorgestellte, mehrstufige Weiterentwicklung präsentiert werden. Bei mittelgroßen Trainingsmengen kann hiermit der Fehler im Vergleich zur Dictionary-basierten Strategie bei gleichen Trainingsdaten deutlich reduziert werden. In den Resultaten in Tabelle 4.6 beispielsweise reduziert sich der Fehler $\overline{d}$ nach arithmetischem Mittel von 1.153,8 km auf 590,43 km bei quasi gleichbleibendem Median $\widetilde{d}$, was einer Verringerung um 48,83 % entspricht. Desweiteren wurde eine k-NN-Strategie vorgestellt, welche die Fokusbestimmung ebenfalls mittels Textklassifikation, jedoch ohne Rastertransformation vornimmt. Letztere erzielt aber keine bessere Vorhersagegenauigkeit als die mehrstufige Dictionary-basierte Variante.

Die Auswirkungen einzelner Konfigurationsparameter wurden detailliert untersucht und die insgesamt zehn vorgestellten Verfahren mit drei verschiedenen Datensets unterschiedlichen Umfangs evaluiert. Ein Vergleich dieses Umfangs wurde nach aktuellem Stand bisher in keiner verwandten Arbeit vorgenommen.

Die zweite Hypothese dieser Arbeit (siehe Abschnitt 1.4), welche besagt, dass bei Verwendung kleiner Trainingsmengen und eines Machine-Learning-basierten Klassifikators zum Ranking der geographische Fokus von Texten besser bestimmt werden kann als mittels Textklassifikation, wurde unter Verwendung des Datensets „TUD-Loc-2013“ und des Wikipedia-Datensets bestätigt. Die erreichten Resultate können per Textklassifikation nur annähernd erzielt werden, wenn Trainingsdaten in beträchtlich höherem Umfang zur Verfügung stehen, wie hier mittels Training aus dem Wikipedia-Dump gezeigt wurde, der über eine halbe Million Trainingsdokumente bereitstellt.

5 AGGREGATION EREIGNISRELEVANTER NUTZERGENERIERTER INHALTE

In den vorangehenden Kapitel wurde gezeigt, wie einzelne Lokationen und der geographische Fokus aus Nachrichtentexten extrahiert werden kann. In diesem Kapitel werden diese Informationen genutzt, um eine automatisierte Aggregation nutzergenerierter Inhalte für Nachrichtenereignisse vorzunehmen und somit die dritte und vierte Forschungsfrage dieser Arbeit zu adressieren.

Das Kapitel zeigt, wie Nachrichtenereignisse (siehe Abschnitt 1.2) durch eine kontinuierliche Überwachung von konventionellen Nachrichtenquellen wie beispielsweise CNN, Reuters oder BBC und ein Clustering der akquirierten Beiträge erkannt werden. Erkannte Ereignisse induzieren eine zielgerichtete Aggregation auf Quellen für nutzergenerierte Inhalte. Wird beispielsweise das Ereignis „Proteste auf dem Majdan Nesaleschnosti in Kiew“ erkannt, ist das Ziel, passende Suchanfragen zu generieren, mit denen Texte, Bilder und Videos auf Instagram, Twitter, YouTube oder Flickr gefunden werden können, welche die Proteste aus Sicht von Augenzeugen portraitieren. Hiermit werden die Voraussetzungen für die in Abschnitt 1.1 skizzierte Vision einer Social-Media-unterstützten Nachrichtenplattform, die nachfolgend als „NewsSeecr“ bezeichnet wird, geschaffen.

Im Kapitelverlauf werden verschiedene Anfragestrategien vorgestellt, mit denen ein hoher Recall bei der Aggregation ereignisrelevanter Inhalte erzielt werden soll. Um innerhalb der Ergebnisse eine hohe Precision zu erreichen, also irrelevante Resultate auszufiltern, die durch die Recall-priorisierten Aggregationsstrategie gesammelt wurden, werden Klassifikationsmethoden eingesetzt.

5.1 VERWANDTE ARBEITEN

Motiviert durch die ursprüngliche Topic-Detection-and-Tracking-Initiative (J. Allan, 2002) entstanden eine Reihe von Arbeiten, die die dort vorgestellten Konzepte auf nutzergenerierte Inhalte und Social-Media-Plattformen übertrugen.

Ikeda, Fujiki und Okumura (2006) präsentieren eine Methode, um Nachrichtenartikel mit Blogbeiträgen zu verlinken. Sie siedeln das Problem im Bereich „Cross Database Retrieval“ an, extrahieren repräsentative Termvektoren aus den jeweiligen Dokumenten und ermitteln Ähnlichkeiten auf Basis dieser Vektoren.

Das NewsStand-System (Teitler et al., 2008) ähnelt in Teilen dem hier vorgestellten NewsSeecr-Konzept. Beiträge von Nachrichtenplattformen werden mittels Web-Feeds aggregiert und geclustert, um diese ereignisbasiert zusammenzufassen. Auch hier wird eine Extraktion von Geo-Daten vorgenommen, die jedoch ausschließlich dazu dient, dem Anwender die Nachrichtenereignisse auf einer Landkarte zu visualisieren. Nutzergenerierte Inhalte werden in NewsStand nicht berücksichtigt. Mit PhotoStand (Samet, Adelfio, Fruin, Lieberman und Sankaranarayanan, 2013) wird ein System vorgestellt, das auf NewsStand aufbaut und Fotos aus den Artikelseiten extrahiert.

TwitterStand von Sankaranarayanan, Samet, Teitler, Lieberman und Sperling (2009) zielt darauf ab, auf Twitter Nachrichtenereignisse zu identifizieren, indem sie Clusteringverfahren auf Tweet-Nachrichtenströme anwenden. Das gleiche Ziel verfolgt Becker, Naaman und Gravano (2011), die eventspezifische Features vorstellt und diese zur Klassifikation verwendet. Watanabe, Ochi, Okabe und Onai (2011) extrahiert Lokationsinformationen aus Tweets, um lokale Ereignisse zu erkennen. Andere Twitter-zentrierte Arbeiten konzentrieren sich auf die Betrachtung bestimmter Naturkatastrophen, wie beispielsweise Waldbrände (Longueville, Smith und Luraschi, 2009), Überflutungen (Starbird, Palen, Hughes und Vieweg, 2010) oder Erdbeben (Sakaki, Okazaki und Matsuo, 2010). Benson, Haghighi und Barzilay (2011) hingegen extrahieren Beschreibungen von Konzertereignissen aus Tweets, während Nichols, Mahmud und Drews (2012) Sportereignisse anhand von Twitter-Inhalten zusammenfassen. Marcus et al. (2011) stellt einen Visualisierungsansatz vor, der Ereignisse auf Twitter anhand ihrer Aktivitäten darstellt. Die hier genannten Vertreter repräsentieren nur einen kleinen Bruchteil der Arbeiten aus dem Twitter-Umfeld. Die Recherche ergab jedoch, dass keine Arbeiten existieren, die neben Twitter noch andere Quellen in die Aggregation miteinbeziehen und eine quellenübergreifende Betrachtung vorzunehmen, wie es das Ziel der vorliegenden Arbeit ist.

Mit dem Prototyp „Vox Civitas“ stellen Diakopoulos, Naaman und Kivran-Swaine (2010) ein Werkzeug für Journalisten vor, das diese bei der Analyse nutzergenerierter Inhalte für Ereignisse unterstützen soll. Die Arbeit konzentriert sich jedoch auf die Konzeption und Evaluierung des Systems aus Nutzersicht und stellt keine Methoden zur Inhaltsaggregation vor.

Tsagkias, de Rijke und Weerkamp (2011) suchen gezielt nach textuellen Social-Media-Inhalten, um diese in Verbindung mit Nachrichtenereignissen zu bringen. Allerdings wird in der Arbeit von einer simulierten „Laborumgebung“ mit lokalem Datenbestand ausgegangen, der es den Autoren gestattet, ausgefeilte Anfrage- und Rankingmodelle für die Suche zu adaptieren. Das hier vorgestellte NewsSeecr-System hingegen wird im Hinblick auf den Praxiseinsatz konzipiert, bei dem die betrachteten Quellen „Black Boxes“ sind und somit auf das Ranking keinen Einfluss genommen werden kann. Tsagkias et al. (2011) setzen verschiedene Anfragemodelle um, die aus Texten der betrachteten Artikel und Social-Media-Referenzinhalten unter Ausnutzung explizit vorhandener Links erzeugt werden. Um die gerankten Resultatlisten einzelner Anfragen zu kombinieren, werden die von He und Wu (2008) vorgeschlagenen Methoden der „Late Data Fusion“ angewendet.

Psallidas, Becker, Naaman und Gravano (2013) stellen Verfahren zur Erkennung von Ereignissen und zur Aggregation zugehöriger Inhalte auf Social-Media-Plattformen vor. Sie unterscheiden zwischen bekannten Ereignissen, wie beispielsweise dem amerikanischen Super Bowl und unbekannten Ereignissen, die der Definition in dieser Arbeit entsprechen (siehe Abschnitt 1.2). Für das erste Szenario müssen die Autoren auf vorhandenes Wissen über die Ereignisse zurückgreifen, welches aus Kalendern von Event-Plattformen bereitgestellt wird und aus dem Suchanfragen generiert werden, mit denen die gewünschten Inhalte gesucht werden. Für das zweite Szenario zur Erkennung unbekannter Ereignisse schlagen die Autoren ein Clusteringverfahren vor.

Von Assaf, Senart und Troncy (2013) wird mit SNARC ein System zur kontextbasierten Aggregation von Social-Media-Inhalten vorgestellt. Das System ist als Browsererweiterung realisiert und greift auf verschiedene Standard-Web-APIs zu. Die Darstellung beschränkt sich jedoch auf Architekturaspekte, eine Evaluierung aus der Information-Retrieval-Perspektive erfolgt nicht.

Der von der grundlegenden Idee ähnlichste Ansatz zu NewsSeecr ist jener von Tsagkias et al. (2011), in dem ebenfalls eine zielgerichtete Suche nach nutzergenerierten Inhalten vorgenommen wird. Allerdings beschränkt sich erwähnte Arbeit auf textorientierte Daten und es werden keine dedizierten Geo-Anfragen verwendet. Die Autoren konnten die Suche genau auf ihren Anwendungszweck abzustimmen, was im Szenario dieser Arbeit nicht möglich ist. NewsSeecr verwendet vergleichsweise einfache Anfragen, die einen hohen Recall erzielen sollen und wendet anschließend Klassifikationsmethoden an, um die Relevanz der gefundenen Ergebnisse zu gewährleisten. Ähnlich Psallidas et al. (2013) wird innerhalb von NewsSeecr ein Clustering vorgenommen, um Ereignisse zu erkennen. Dies geschieht jedoch nicht mit nutzergenerierten Inhalten, sondern mit aggregierten Nachrichten von konventionellen Nachrichtenplattformen. Erst im Anschluss werden auf Basis erkannter Ereignisse Suchanfragen generiert und an Quellen mit nutzergenerierten Inhalten gestellt. Durch diese Methode ergibt sich folgender Vorteil: Die Erkennung von Ereignissen kann durch Beobachtung vertrauenswürdiger Nachrichtenquellen zuverlässiger und mit geringerem Aufwand erfolgen als dies beispielsweise anhand der Nachrichtenströme von Twitter möglich wäre, da hier nur wenig Textinformation aber viel „Rauschen“ in Form irrelevanter Tweets vorhanden ist.

5.2 DATENQUELLEN

Wie im Rahmen der in Abschnitt 1.5 vorgestellten Architektur beschrieben, unterscheidet NewsSeecr zwischen konventionellen Nachrichtenplattformen und Plattformen für nutzergenerierte Inhalte. Für beide Arten von Datenquellen werden unterschiedliche Zugriffsmethoden verwendet.

Konventionelle Nachrichtenplattformen Web-Feeds im RSS-[96] und Atom-Standard[97] erlauben es Computerprogrammen, Webseiten auf Änderungen und neue Inhalte zu überwachen. Im Gegensatz zu präsentationsorientierten HTML-Seiten genügen Web-Feeds einer strikten Struktur,

96 http://cyber.law.harvard.edu/rss/rss.html

97 http://www.ietf.org/rfc/rfc4287.txt

die eine automatisierte Verarbeitung vereinfachen. Praktisch jede Nachrichtenseite bietet einen Web-Feed an, der mittels regelmäßigem Polling auf neue Inhalte geprüft werden kann. Innerhalb der NewsSeecr-Architektur kommt der Feed Reader aus Palladian (Urbansky, Muthmann, Katz und Reichert, 2012) zum Einsatz, der auf Basis der Arbeiten von Urbansky (2012) und Reichert (2012) optimale Abfrageintervalle für einzelne Web-Feeds aus den tatsächlichen Aktualisierungsintervallen ermittelt und somit ein effizientes Polling gestattet. Der verwendete Prototyp von NewsSeecr nutzt eine manuell zusammengestellte Auswahl von 46 international relevanten Nachrichtenfeeds, die zusammen täglich deutlich über 2.000 Beiträge publizieren.

Plattformen für nutzergenerierte Inhalte Wie in Kapitel 1 dargestellt, werden zu Spitzenzeiten auf Twitter aktuell bis zu 7.000 Nachrichten pro Sekunde versendet. Eine kontinuierliche Auswertung sämtlicher Daten dieser Plattformen stellt sich somit bereits aufgrund der schieren Masse als schwierig dar. Desweiteren stellen die betrachteten Quellen in der Regel keine bzw. keine freien Möglichkeiten bereit, den kompletten Nachrichtenstrom „mitzuschneiden“[98]. Andererseits gehen Untersuchungen davon aus, dass nachrichtenbezogene Inhalte auf Twitter lediglich 8 % der ausgetauschten Gesamt-Tweets ausmachen (Dann, 2010). Aus diesem Grund wird eine Aggregationsstrategie verfolgt, bei der zielgerichtete Suchanfragen an die jeweiligen Quellen gestellt werden. Ermöglicht wird dies dank der in den vergangenen Jahren gewachsenen Popularität von REST-Paradigmen (Fielding, 2000) und die Bestrebungen verbreiteter Web-Plattformen, gut dokumentierte REST-APIs für externe Entwickler und Anwendungen zur Verfügung zu stellen[99].

5.3 CLUSTERING UND EREIGNISERKENNUNG

Die Erkennung von Ereignissen in NewsSeecr gemäß der Definition in Abschnitt 1.2 wird mittels Clustering und einer nachfolgender Klassifikation realisiert. Die umgesetzten Verfahren ähneln durch den Einsatz eines Single-Pass-Clusterings den Konzepten der „New Event Detection“ (J. Allan, Papka und Lavrenko, 1998) im Topic Detection and Tracking (J. Allan, 2002).

Als problematisch erweist sich der Umstand, dass die betrachteten Nachrichtenquellen nicht ausschließlich über aktuelle Ereignisse wie zum Beispiel „Dozens dead as magnitude-7.1 earthquake hits the Philippines“[100] berichten, sondern auch eigene redaktionelle Beiträge wie zeitliche Rückblenden (z. B. „Retro Report: The Clone Named Dolly“[101]), allgemeine Reportagen (z. B. „Why Malala's bravery inspires us“[102]), Essays oder Kolumnen enthalten, die jeweils keinen direkten zeitlichen Bezug zu einem aktuellen Ereignis gemäß der verwendeten Definition haben.

98 Twitter beispielsweise gewährt mittlerweile nur noch über externe Dienstleister einen Zugriff auf den sogenannten „Garden Hose“

99 Beispiel Flickr: `http://www.flickr.com/services/api/`

100 `http://edition.cnn.com/2013/10/14/world/asia/philippines-earthquake/`

101 `http://www.nytimes.com/2013/10/14/booming/the-clone-named-dolly.html`

102 `http://edition.cnn.com/2013/10/10/opinion/fine-malala-essay-contest/index.html`

Durch das thematische Clustering wird ein erster Schritt unternommen, um ereignisrelevante Themen zu identifizieren. Wichtige Ereignisse von internationalem Interesse, wie das oben erwähnte Beispiel zum Erdbeben auf den Philippinen, finden innerhalb kürzester Zeit Beachtung in der Berichterstattung mehrerer Nachrichtenquellen und sollen somit große Cluster generieren, die sämtliche Nachrichten zum Ereignis enthalten. Sehr spezifische Ereignisse von ausschließlich lokaler Relevanz oder die oben charakterisierten redaktionellen Beiträge hingegen werden in der Regel nur von einer einzigen oder wenigen Quellen publiziert und erzeugen somit ein- oder wenigelementige Cluster. Durch die Betrachtung der Clustergröße können Ereigniscluster gut von anderen separiert werden.

Ein weiterer Vorteil, der aus dem Clustering mehrerer Nachrichtenbeiträge gezogen wird, ist die Redundanz für die Informationsextraktion. Obwohl die hier konzipierten Verfahren zur Lokationsextraktion bessere Resultate liefern als vergleichbare Ansätze, sind diese nicht perfekt. Durch die Kombination der Extraktionsergebnisse mehrerer Dokumente pro Ereignis können Fehler ausgeglichen werden.

5.3.1 CLUSTERING

Bei statischen Dokumentmengen bieten sich zum Clustering Methoden wie k-Means (MacQueen, 1967; Steinhaus, 1957) an, bei denen eine feste Anzahl gewünschter Cluster im Vorfeld festgelegt werden muss. Die hier betrachteten kontinuierlichen Nachrichtenströme hingegen verlangen nach einer dynamischeren Methodik. Entsprechend ähnlicher Arbeiten zur Ereigniserkennung (J. Allan et al., 1998) wird hier auf ein Single-Pass-Prinzip (Frakes und Baeza-Yates, 1992) zurückgegriffen, bei dem eingehende Nachrichtenbeiträge jeweils mit bereits bestehenden Clustern verglichen und bei Überschreiten eines festgelegten Schwellwerts dem ähnlichsten Cluster zugewiesen werden. Falls der Schwellwert nicht überschritten wird, wird ein neues Cluster erzeugt. Die Anzahl der tatsächlich erzeugten Cluster ergibt sich also aus der Wahl des Schwellwerts. Höhere Schwellwerte führen intuitiv zu einer größeren Menge kleiner Cluster, wohingegen geringe Schwellwerte in einer kleinen Anzahl vergleichsweise großer Cluster münden.

Der Ablauf des verwendeten Cluster-Algorithmus ist in Abbildung 5.1 dargestellt. Das zu verarbeitende Dokument wird mit d bezeichnet, C gibt die Menge sämtlicher Cluster an, der Ähnlichkeitsschwellwert wird mit *similarityThreshold* bezeichnet. Mittels $\mathsf{similarity}(d,c)$ wird die Ähnlichkeit zwischen den Dokumenten d und c bestimmt. Konkrete Ähnlichkeitsmaße werden nachfolgend vorgestellt. Um zu verhindern, dass ähnliche Dokumente, welche den festgelegten Ähnlichkeitsschwellwert knapp unterschreiten auf zwei Cluster separiert und die Clusterstruktur somit im Zeitverlauf zu stark fragmentiert, wird nach jeder Änderung eines vorhandenen Clusters überprüft, ob das veränderte Cluster den Ähnlichkeitsschwellwert zu einem anderen Cluster unterschreitet. Trifft dies zu, werden beide Cluster mitenander verschmolzen. Diese Idee wurde bereits erfolgreich

```
funct clusterDocument(d, C, similarityThreshold) ≡
  bestCluster := null;
  bestSimilarity := 0;
  for c in C do                                          // Determine most similar cluster.
    similarity := similarity(d,c);
    if similarity > bestSimilarity then
                                    bestSimilarity := similarity;
                                    bestCluster := c;
    fi
  end
  if bestCluster ≠ null ∧ bestSimilarity ≥ similarityThreshold
    then
      bestCluster ← d;                                   // Add document to existing cluster.
      for otherCluster in (C ∖ bestCluster)              // Try to merge cluster into existing one.
        if similarity(bestCluster, otherCluster) ≥ similarityThreshold
          then
            otherCluster := otherCluster ∪ bestCluster;
            C := C ∖ bestCluster;
            break;
        fi
      end
    else C ← newCluster(d);                              // Create new cluster with document.
  fi.
```

Abbildung 5.1: Pseudocode für Single-Pass-Clustering

von Sankaranarayanan et al. (2009) eingesetzt. Zur Ähnlichkeitsbestimmung zwischen zwei Clustern kommt die gleiche Ähnlichkeitsfunktion wie zwischen zwei Dokumenten zum Einsatz.

Aufgrund dessen, dass die betrachteten Nachrichtendokumente jeweils eine starke zeitliche Abhängigkeit vom zugehörigen Ereignis haben, können regelmäßig ältere, inaktive Cluster, denen keine neuen Inhalte mehr hinzugefügt wurden, aus der betrachteten Clustermenge C entfernt werden. Somit können Ressourcen im Hinblick auf Laufzeit und Speicher eingespart werden. Im Rahmen des konzipierten Systems geschieht dies durch die Festlegung einer Maximalmenge von aktiven Clustern. Wird diese Menge überschritten, wird der älteste Cluster aus C entfernt.

5.3.2 Ähnlichkeitsfunktionen

Zum Clustering wird eine geeignete Funktion $\mathsf{similarity}(d,c)$ benötigt, mit der die Ähnlichkeit zwischen Dokumenten bestimmt wird. Nachfolgend werden drei mögliche Ähnlichkeitsfunktionen und eine Kombination vorgestellt.

Textähnlichkeit Für die Textähnlichkeit wird ein Bag-of-Words-Modell genutzt, bei dem die einzelnen Wörter mittels TF-IDF gewichtet werden (siehe Abschnitt 2.2). Im Rahmen der Vorverarbeitung wird der Text in Kleinbuchstaben umgewandelt, anschließend werden jene Wörter entfernt, die nicht aus alphanumerischen Zeichen bestehen, sich auf einer Stoppwortliste mit 590 englischen Stoppwörtern befinden oder kürzer als zwei Zeichen sind. Für Cluster werden die Texte sämtlicher enthaltener Dokumente konkateniert.

Die IDF-Berechnung erfolgt auf Basis eines aus der englischsprachigen Wikipedia generierten Dokumentkorpus mit 642.607 Einträgen. Es wurde außerdem mit zeichen- und wortbasierten n-Grammen und einer Gewichtungen nach TF und IDF experimentiert, die besten Resultate wurden jedoch mit der vorangehend beschriebenen Vorgehensweise erzielt. Die Ähnlichkeitsbestimmung in Formel 5.1 geschieht mittels Kosinus-Ähnlichkeit (Formel 5.2), die auf ein Intervall $[0, 1]$ abbildet.

$$\text{similarity}_{text}(d, c) = \cos(\text{preprocess}(d), \text{preprocess}(c)) \tag{5.1}$$

$$\cos(a, b) = \frac{a \cdot b}{\|a\| \|b\|} \tag{5.2}$$

Zeitliche Ähnlichkeit Der Faktor Zeit stellt ein weiteres Merkmal dar, um zu beurteilen, ob sich zwei Nachrichtenbeiträge auf das selbe Ereignis beziehen. Je näher die Veröffentlichungszeitpunkte zweier Beiträge beieinander liegen, desto höher ist hierfür die Wahrscheinlichkeit. Zur Gewichtung der zeitlichen Differenz in Stunden wird eine Hill-Funktion (Hill, 1910) eingesetzt. Die Hill-Funktion hat einen Wertebereich von $[0, 1]$ und kann als logarithmische Variante einer Sigmoid-Funktion betrachtet werden. Nachfolgend für angenommen, dass gilt $\lambda = 1$ und $h = 72$. Bei einem zeitlichen Abstand von beispielsweise drei Tagen beträgt die Ähnlichkeit somit 0,5. Cluster, die aus mehreren Dokumenten bestehen, werden anhand des Durchschnitts aus den Veröffentlichungszeitpunkten sämtlicher enthaltener Dokumente repräsentiert.

$$\text{similarity}_{time}(d, c) = 1 - \text{hill}(\text{hoursDifference}(d, c)) \tag{5.3}$$

$$\text{hill}(x) = \frac{x^{\lambda}}{h^{\lambda} + x^{\lambda}} \tag{5.4}$$

Räumliche Ähnlichkeit Als drittes Ähnlichkeitsmaß wird die in Formel 5.5 angegebene räumliche Nähe zwischen zwei Dokumenten bestimmt. Verwendet wird die in Kapitel 4 beschriebene Fokusbestimmung, die aus jedem Dokument die Hauptlokation extrahiert. Die Distanz in Kilometern zwischen den Hauptlokationen wird anhand Formel 2.1.2 berechnet. Die Koordinaten von Clustern werden aus dem geometrischen Median (Formel 2.1.4) der Koordinaten der enthaltenen Dokumente ermittelt. Entsprechend der zeitlichen Ähnlichkeit wird auch hier eine Gewichtung mit der Hill-Funktion (Formel 5.4) vorgenommen. Die Parameter der Hill-Funktion wurde wie folgt festgelegt: $\lambda = 1$ und $h = 100$. Für den Fall, dass ein Dokument bzw. Cluster keine Lokation enthält, wird eine Ähnlichkeit von 0 angenommen.

$$\text{similarity}_{geo}(d, c) = 1 - \text{hill}(\text{distance}(\text{focus}(d), \text{focus}(c))) \tag{5.5}$$

Kombinierte Ähnlichkeiten Die drei vorgestellten Ähnlichkeiten können miteinander kombiniert werden. Hierfür bietet sich eine Multiplikation der Einzelwahrscheinlichkeiten wie in Formel 5.6 beispielhaft für die Kombination aus Textähnlichkeit und zeitlicher Ähnlichlichkeit angegeben an.

$$\text{similarity}_{text \cdot time}(d, c) = \text{similarity}_{text}(d, c) \cdot \text{similarity}_{time}(d, c) \tag{5.6}$$

5.3.3 DATENSET

Die Evaluierung von Clusteringverfahren erweist sich als schwierig. Für den hier betrachteten Anwendungsfall existieren keine Referenzdatensets, die für einen Vergleich verwendet werden könnten. Auf der anderen Seite bringt die manuelle Erzeugung eines umfassenden und repräsentativen Datensets einen erheblichen Aufwand mit sich, da hier jede denkbare Dokument-Cluster-Paarung manuell bewertet werden müsste. In der Praxis erweist sich eine solche Vorgehensweise als unpraktikabel.

Um trotzdem tendenzielle Aussagen über die vorgestellten Ähnlichkeitsfunktionen treffen und eine Optimierung der Algorithmen vornehmen zu können, wurde auf automatischem Wege ein Datenset von der Nachrichtenplattform Wikinews[103] erzeugt. Wikinews gehört ebenso wie die Wikipedia zum Wikimedia-Projekt[104] und ist eine Wiki-basierte Plattform, auf der jeder Nutzer Artikel schreiben und bearbeiten kann. Die Artikel auf Wikinews zeichnen sich dadurch aus, dass sie neben dem eigentlichen Nachrichtentext zumeist noch über eine Sektion „Sources“ mit einer Liste von Links auf externe Nachrichtenartikel zum jeweiligen Thema, die Hintergrundinformationen liefern und Primärquellen darstellen. Abbildung 5.2 zeigt beispielsweise die Quellen für den Artikel „Iran to reduce nuclear enrichment in exchange for sanctions reduction“[105].


Sources [edit]

- "Iran welcomes nuclear deal which Israel calls 'mistake'" — *BBC News Online*, November 25, 2013
- "Iran nuclear deal: Key points" — *BBC News Online*, November 25, 2013
- "Netanyahu calls Iranian deal 'historic mistake'" — *Ynetnews*, November 24, 2013
- Michal Margalit. "Obama to Netanyahu: US will remain committed to Israel" — *Ynetnews*, November 24, 2013
- Julian Borger. "Iran nuclear agreement: Q&A" — *The Guardian*, November 24, 2013
- Saeed Kamali Dehghan. "Iran's leaders and public celebrate Geneva nuclear deal" — *The Guardian*, November 24, 2013
- Dan Roberts. "Obama admits Israel has good reason for scepticism over Iran nuclear deal" — *The Guardian*, November 24, 2013
- Harriet Sherwood. "Israel condemns Iran nuclear deal as 'historic mistake'" — *The Guardian*, November 24, 2013
- "Iran nuclear deal: David Cameron praises 'important first step'" — *BBC News Online*, November 24, 2013

Abbildung 5.2: Quellen innerhalb des englischen Wikinews-Artikels „Iran to reduce nuclear enrichment in exchange for sanctions reduction“

Es wurde die Annahme getroffen, dass jeder Wikinews-Artikel innerhalb des zu erzeugenden Datensets ein Referenzcluster darstellt und die jeweils verlinkten externen Quellen die zu clusternden

103 http://www.wikinews.org
104 http://www.wikimedia.de/wiki/Hauptseite
105 http://en.wikinews.org/wiki/Iran_to_reduce_nuclear_enrichment_in_exchange_for_sanctions_reduction

Dokumente repräsentieren. Erstellt wurde das Datenset unter Zuhilfenahme der MediaWiki-API[106] und des MediaWiki-Parsers aus Palladian (Urbansky, Muthmann, Katz und Reichert, 2012). Für den Zeitraum vom 1. Januar 2010 bis zum 31. Dezember 2012 wurden von der englischsprachigen Seite sämtliche Wikinews-Artikel bezogen. Aus den resultierenden 3.299 Artikeln wurden jene entfernt, die den Bestandteil „Wikinews Shorts“ im Titel enthielten, da dies zusammengefasste Kurznachrichten mit unterschiedlichen Themen sind[107].

Aus den verbleibenden Artikeln wurden die Links aus der „Sources“-Sektion extrahiert, die auf externe Seiten verwiesen und die folgenden Bereinigungsschritte vorgenommen: Kurze URLs, die lediglich auf Startseiten verwiesen, wurden ausgefiltert, außerdem wurden Links auf YouTube-Videos und PDF-Dateien entfernt. Ignoriert wurden ferner sämtliche Links zu Artikeln von „The New York Times“ und „The Times“[108], da diese nur bei vorhandenem Abonnement aufgerufen werden können. Aus den Webseiten der übrigen URLs wurde mittels Inhaltsextraktor aus Palladian der Hauptinhalt extrahiert. Nicht mehr erreichbare Seiten, deren Aufruf in einem HTTP-Status-Code ≥ 400 resultierten, wurden ignoriert.

Das erstellte Datenset besteht aus 2.957 Referenzclustern mit 8.511 Dokumenten. Die Veröffentlichungszeitpunkte der Dokumente wurden aus den Datumsangaben der zugehörigen Wikinews-Artikel extrahiert (siehe Abbildung 5.2). Es zeigte sich, dass die Anzahl der publizierten Wikinews-Artikel über den betrachteten Zeitraum nicht gleichverteilt ist und über die drei Jahre jeweils abgenommen hat. Das Jahr 2010 umfasst 1.637 Wikinews-Artikel mit 4.190 dazugehörigen Nachrichtendokumenten, Jahr 2011 besteht aus 824 Artikeln mit 2.446 Dokumenten, im Jahr 2012 sind 496 Artikel mit 1.875 Dokumenten vorhanden. Das Trainings- und Validierungsset wird deshalb durch das Jahr 2010 repräsentiert, das Testset zusammen durch die Jahre 2011 und 2012.

5.3.4 EXPERIMENTE UND OPTIMIERUNG

Unter Verwendung des Trainings-/Validierungssets wird nachfolgend der optimale Ähnlichkeitsschwellwert für die einzelnen Ähnlichkeitsmaße bestimmt, außerdem wird evaluiert, mit welcher Ähnlichkeitsfunktion sich die besten Clustering-Ergebnisse erzielen lassen. Betrachtet werden die einzelnen in Abschnitt 5.3.2 beschriebenen Ähnlichkeitsmaße und mögliche Kombinationen.

BCubed-Metrik Zur Evaluierung kommt die BCubed-Metrik (Bagga und Baldwin, 1998) zum Einsatz. Verglichen wird jeweils zwischen dem Referenzclustering des Datensets und den mittels der hier beschriebenen Verfahren erzeugten Clusterings. Amigó, Gonzalo, Artiles und Verdejo (2009), die in einer umfangreichen Untersuchung verschiedene Evaluierungsmaße für Clusteringverfahren untersucht haben, zeigen, dass BCubed die einzige Metrik ist, die sämtliche der vier Qualitätsbedingungen Homogenität, Vollständigkeit, „Rag Bag“ (wird einem bereits unsauberen Cluster ein unpassendes

106 http://www.mediawiki.org/wiki/API:Main_page

107 Beispiel: http://en.wikinews.org/wiki/Wikinews_Shorts:_May_5,_2012

108 http://www.nytimes.com, http://www.timesonline.co.uk, http://www.thetimes.co.uk

Dokument hinzugefügt, wird dies weniger bestraft als wenn dies bei einem sauberen Cluster passiert), sowie Größe-Quantität-Tradeoff (ein geringer Fehler in einem großen Cluster ist einer großen Anzahl Fehlern in kleinen Clustern vorzuziehen) berücksichtigt. BCubed ermittelt zunächst für jedes Dokument d im Datenset D die BCubed-Precision (siehe Formel 5.7) und Recall (entsprechend der Precision unter Vertauschung der *Reference*- und *Result*-Mengen). cluster(S, d) bezeichnet hierbei das Cluster aus Menge S, welches das Dokument d enthält (Carpenter, 2011).

$$\mathrm{b}^3\mathrm{Precision}(d, \mathit{Reference}, \mathit{Result}) = \frac{|\,\mathrm{cluster}(\mathit{Result}, d) \cap \mathrm{cluster}(\mathit{Reference}, d)\,|}{|\,\mathrm{cluster}(\mathit{Result}, d)\,|} \tag{5.7}$$

Anschließend werden aus BCubed-Precision und -Recall die Durchschnittswerte für das gesamte Clustering ermittelt. Nachfolgend geschieht dies dokumentweise gemäß Formel 5.8 (analog für den Recall). Aus BCubed-Precision und -Recall wird das kombinierte BCubed-F1-Maß als harmonisches Mittel wie in Abschnitt 2.4 beschrieben berechnet.

$$\mathrm{b}^3\mathrm{Precision}(\mathit{Reference}, \mathit{Result}) = \sum_{d \in D} \frac{\mathrm{b}^3\mathrm{Precision}(d, \mathit{Reference}, \mathit{Result})}{|\,D\,|} \tag{5.8}$$

Optimale Schwellwerte und Evaluierungsergebnisse Für die Bestimmung des optimalen Ähnlichkeitsschwellwerts *similarityThreshold* wurde dieser für jede Ähnlichkeitsfunktion im Intervall $[0, 1]$ in Schritten von 0,05 erhöht und der bestmögliche BCubed-F1-Wert bestimmt. In Tabelle 5.1 sind die Ergebnisse dargestellt. „Baseline“ repräsentiert ein „triviales“ Clustering, bei dem für jedes Dokument ein einelementiges Cluster erzeugt wird, die BCubed-Precision beträgt somit 100 % bei einem Recall von 41,68 % und somit einem B-Cubed-F1 von 58,84 %. Außerdem sind in Tabelle 5.1 die erzielten Resultate für den jeweils identischen Schwellwert auf dem ungesehenen Testset zu sehen.

Im Hinblick auf das BCubed-F1-Maß wird auf dem verwendeten Datenset das beste Clustering mit der kombinierten Ähnlichkeit „text · time“ erzielt. Auf dem Trainings-/Validierungsset erzielt diese Kombination einen BCubed-F1-Wert von 84,19 %. Von den Einzelähnlichkeiten wird mit „text“ der höchste F1-Wert von 73,65 % erreicht. Die Hinzunahme des Faktors Zeit bringt somit, wie zu erwarten, eine deutliche Verbesserung. Überraschenderweise kann mit den Geo-Kombinationen „text · time · geo“ keine weitere Verbesserung erzielt werden, der F1-Wert liegt hier mit 80,05 % unter jenem für „text · time“. Die Ähnlichkeiten „time“ und „geo“ alleine schneiden mit 34,69 bzw. 40,93 % schlechter ab als die Baseline. Auch auf dem Testset wird mittels „text · time“ das beste Ergebnis erreicht, die Werte sind hier insgesamt jedoch schlechter, was sich bereits anhand der Baseline zeigt, die im BCubed-F1 über zehn Prozentpunkte unter dem Trainings-/Validierungsset liegt.

Fazit Aufgrund der hier dargestellten Ergebnisse liegt die Annahme nahe, dass die Einbeziehung des Faktors „Räumliche Ähnlichkeit“ für das Clustering keine Vorteile bringt. Dies widerspricht jedoch den praktischen Erfahrungen, die im Rahmen des NewsSeecr-Prototyps gewonnen wurden. Wie in Abschnitt 5.2 dargestellt, werden durch das NewsSeecr-System täglich über 2.000 Nachrichtenbeiträge aggregiert. Mit dem vorangehend verwendeten Datenset zur Evaluierung kann

		Training/Valdierung			Test		
Ähnlichkeit	**t**	**B^3-Pr.**	**B^3-Rc.**	**B^3-F1**	**B^3-Pr.**	**B^3-Rc.**	**B^3-F1**
Baseline		100	41,68	58,84	100	32,15	48,66
text	0,5	79,07	68,93	73,65	75,86	60,7	67,44
time	0,85	22,16	69,9	34,69	36,39	65,62	46,81
geo	0,95	30,66	61,54	40,93	27,87	51,01	36,05
text · time	0,25	90,91	78,4	**84,19**	91,38	70,27	**79,45**
text · geo	0,35	77,91	64,98	70,86	76,4	54,23	63,43
time · geo	0,6	80,52	66,16	72,64	82,7	54,23	65,5
text · time · geo	0,1	87,14	74,03	80,05	87,17	65,55	74,83

Abkürzungen: t = Ähnlichkeitsschwellwert,
B^3-Pr./-Rc./-F1 = BCubed-Precision/-Recall/-F1-Maß

Tabelle 5.1: Resultate beim Clustering mit höchstem F1-Maß auf dem Trainingsset und die dazugehörigen Schwellwerte sowie die Ergebnisse auf dem Testset; alle Werte in Prozent, Bestwert für F1-Maß jeweils fett hervorgehoben

dieses Nachrichtenaufkommen nicht adäquat simuliert werden, da hier pro Tag gerade einmal 8 Nachrichtenbeiträge in durchschnittlich 2,7 Clustern vorhanden sind.

Die deutlich stärkere Steuung von Quellen innerhalb des realen NewsSeecr-Systems führt dazu, dass häufig innerhalb eines gemeinsamen Zeitraums Nachrichtenmeldungen gleichen Typs aggregiert werden, die sich jedoch auf unterschiedliche tatsächliche Ereignisse beziehen (beispielsweise Meldungen zum Thema „Autounfall“, die sich auf verschiedene Orte beziehen jedoch identisches Vokabular wie „crash“, „car“, „driver“ enthalten). Werden hier nur die Ähnlichkeitsfaktoren Text und Zeit berücksichtigt, entstehen häufig Cluster, die Beiträge verschiedener Ereignisse ähnlichen Typs an unterschiedlichen Orten umfassen.

In der Praxis hat sich gezeigt, dass durch die Verwendung der Kombination „text · time · geo“ diesem Problem wirkungsvoll vorgebeugt werden kann. Eine Evaluierung ist jedoch aufgrund der in Abschnitt 5.3.3 beschriebenen Einschränkungen schwierig.

5.3.5 EREIGNISERKENNUNG

Im Anschluss werden die einzelnen Cluster in die Kategorien „Ereignis“ bzw. „Nicht-Ereignis“ klassifiziert. Hier sollen jene Cluster für nachfolgende Verarbeitungsschritte ausgefiltert werden, die keinen Ereignisbezug gemäß der Definition in Abschnitt 1.2 aufweisen. Die Ereigniserkennung wird als Textklassifikationsproblem adressiert. Dies begründet sich in der Tatsache, dass Nachrichten, die der verwendeten Definition genügen, oft eindeutige Terme enthalten, wie beispielsweise „protests“

oder „earth quake“, wohingegen Terme wie „announce“ oder „proposal“ negative Indikatoren sind.

Um den Textklassifikator zu trainieren und Aussagen bezüglich der erzielten Genauigkeit treffen zu können, wurden aus dem NewsSeecr-Datenbestand aus einem Zeitraum von über sechs Monaten 2.939 Cluster ausgewählt, deren Größe zehn Nachrichten überschritt. Aus jedem Cluster wurde zufällig eine Nachricht als Repräsentant ausgewählt und manuell einer der Kategorien „Ereignis“, „Nicht-Ereignis“ oder „unklar“ zugeordnet. Auf die erste Kategorie entfielen 693 Einträge (23,58 %), auf die zweite 1.835 (62,44 %), als unklar wurden 411 Einträge (13,98 %) bewertet.

Die Liste wurde chronologisch in zwei gleichgroße und disjunkte Mengen zum Training bzw. zum unbesehenen Testen aufgeteilt. Jene Dokumente, die im Rahmen der manuellen Annotation mit „unklar“ klassifiziert wurden, wurden entfernt. Andererseits wurde die Dokumentmenge so erweitert, dass sämtliche Dokumente, die sich im gleichen Cluster eines annotierten Dokuments befanden in die Trainings- und Testmengen aufgenommen wurden. Somit umfasste das Traingsset 28.455, das Testset 28.177.

Verwendet wird der Dictionary-basierte Textklassifikator aus dem Palladian-Toolkit (Urbansky, Muthmann, Katz und Reichert, 2012), der bereits im Rahmen der geographischen Fokusbestimmung in Abschnitt 4.4.1 zum Einsatz kam. Unter Verwendung des Traingssets wurden ähnlich der Vorgehensweise in Abschnitt 4.5.2 unterschiedliche Einstellungen zur Feature-Extraktion evaluiert (Wort- und Zeichen-basierte n-Gramme und Kombinationen). Die optimalen Ergebnisse wurden mittels dem in Abschnitt 4.4.1 beschriebenen komplementären Naïve-Bayes-Scorings und der Featurekombination mit zeichenbasierten 3-4-Grammen erzielt. Diese Konfiguration erzielt bei der Klassifikation des ungesehenen Testsets eine Accuracy (siehe Abschnitt 2.4) von 86,95 %. Im Vergleich zur trivialen Baseline, die die häufigste Klasse im Testset vorhersagt („Nicht-Ereignis“) und damit eine Accuracy von 70,86 % erzielt, entspricht dies einer Verbesserung um 16,09 Prozentpunkte.

5.4 ANFRAGEGENERIERUNG UND SUCHE

Die Anfragegenerierung ermittelt in regelmäßigen Zeitintervallen, ob durch die vorangegangenen Verarbeitungsschritte neue Ereigniscluster erkannt wurden. Ist dies der Fall, werden auf jene Cluster, die eine festgelegte Mindestgröße überschreiten, eine Reihe von Informationsextraktionsschritten angewendet, die die Eingabedaten für die zu erzeugenden Suchanfragen liefern. Der Ablauf ist in Abbildung 5.3 dargestellt. Die dargestellte „Relevanzfilterung“ im vorletzten Verarbeitungsschritt wird im letzten Abschnitt dieses Kapitels besprochen.

5.4.1 INFORMATIONSEXTRAKTION

Der folgende Ablauf extrahiert in drei Schritten Lokationsinformationen, deskriptive Schlüsselwörter und -phrasen sowie den Zeitpunkt des Ereignisses für jedes Cluster.

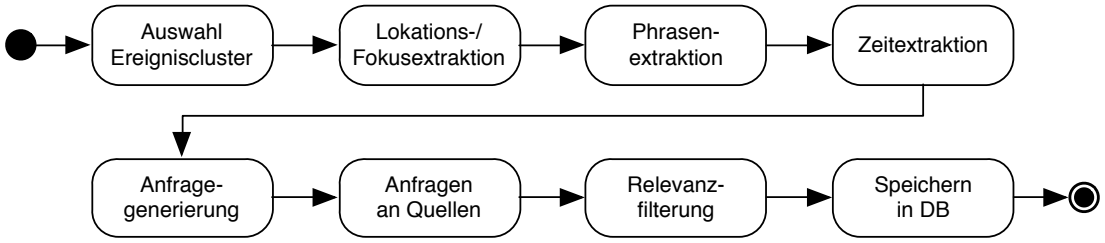


Abbildung 5.3: Ablauf der Anfragegenerierung und Suche

Lokationsextraktion Für diesen Schritt wird auf die in Kapitel 3 vorgestellte Lokationsextraktion und die in Kapitel 4 beschriebene Fokusbestimmung zurückgegriffen. Der Text aller Nachrichtendokumente des betrachteten Clusters wird dazu konkateniert und unter Verwendung des vorgestellten Machine-Learning-basierten Verfahren Lokationsvorkommen L extrahiert. Anschließend wird mittels Fokusbestimmung die Hauptlokation focus(L) extrahiert[109].

Um Fehlextraktionen zu eliminieren, werden aus L jene Lokationen entfernt, die in weniger als 25 % der Dokumente des betrachteten Clusters vorkommen. Außerdem werden solche Lokationen ausgeschlossen, die über 500 km von der ermittelten Hauptlokation focus(L) des Clusters entfernt liegen. Die Menge wird ferner auf maximal zehn Lokationen, sortiert nach Auftrittswahrscheinlichkeit reduziert. Anschließend werden aus L zwei Mengen erzeugt, die Menge C mit Koordinaten und die Menge P mit Ortsnamen.

Die Menge C enthält die Koordinaten sämtlicher punktbasierter Lokationen, für die eine Koordinatensuche vorgenommen wird. Gemäß der hier verwendeten Lokationstypen (siehe Abschnitt 3.4.2) sind dies `CITY` und `POI`. Bei Lokationen anderen Typs, beispielsweise `CONTINENT` oder `COUNTRY` mit großen räumlichen Ausmaßen ist eine koordinatengetriebene Suche hingegen nicht sinnvoll. Um später unnötige Suchanfragen zu vermeiden, werden Koordinaten in C, die weniger als 1 km voneinander entfernt sind, auf einen Punkt reduziert.

In P hingegen werden die Namen sämtlicher extrahierter Lokationen aus L aufgenommen. Verwendet wird der primäre Name aus dem Gazetteer (siehe Abschnitt 3.6). Namen von Ländern, die innerhalb der Datenbank anhand ihrer offiziellen Bezeichnung aufgeführt sind (z. B. „Swiss Confederation“) werden mittels Heuristik auf den kürzesten und somit gebräuchlichsten englischen Alternativnamen reduziert (z. B. „Switzerland“).

Phrasenextraktion Ziel der Phrasenextraktion ist, deskriptive n-Gramme aus dem Clusterinhalt zu extrahieren. Dazu wird der gesamte Text sämtlicher enthaltener Nachrichtendokumente konkateniert und die folgenden Ablaufschritte angewendet:

109 Die nachfolgenden Experimente nutzen noch die einfache Heuristik „Frequenz“, die in Abschnitt 4.2 vorgestellt wurde. Die Machine-Learning-basierte Fokusbestimmung, die im Rahmen dieser Arbeit als beste der verglichenen Strategien ermittelt wurde, war zum Zeitpunkt der Experimente noch in der Entwicklung.

1. Tokenisierung des Texts
2. Filterung von Tokens auf Stoppwortliste, die kürzer als zwei Zeichen sind oder nicht ausschließlich aus alphanumerischen Zeichen bestehen
3. Stemming der Tokens mit Snowball-Stemmer (Porter, 2001) und Aufbewahrung der ursprünglichen Version, um diese wieder zurücktransformieren zu können
4. Filterung von Tokens mit TF-IDF < 0,5, um allgemeine, aussagelose Bestandteile wie „president“ zu entfernen, einzigartige Terme wie „phailin“ und „cyclone“ jedoch zu erhalten[110]
5. Erzeugung von Phrasen aus n-Grammen direkt aufeinanderfolgender Tokens
6. Scoring der Phrasen nach Termfrequenz innerhalb des Dokuments multipliziert mit Anzahl der Tokens: $\text{score}(t,d) = \text{tf}(t,d) \mid \text{tokenize}(t) \mid$
7. Eliminierung von Teilphrasen, die in längeren Phrasen mit gleicher Auftrittshäufigkeit im Dokument vorkommen
8. Zurücktransformation der gestemmten Tokens

Phrasen, die bereits in den Resultaten der vorangehenden Lokationsextraktion waren, werden verworfen. Pro Cluster werden jeweils maximal zehn Phrasen sortiert nach Scoring für jede n-Gramm-Länge im Intervall $[1, 5]$ ermittelt. Um zu verhindern, dass zu generelle Phrasen als Suchbegriffe extrahiert werden, wurde ferner ein Schwellwert von 0,25 als Mindestscore festgelegt.

Zeitextraktion Der Zeitpunkt eines Ereignisses wird näherungsweise durch die Publikationsdaten der im Cluster enthaltenen Nachrichtendokumente determiniert. Ermittelt wird der Medianzeitpunkt über sämtliche Dokumente. Relevant ist dieser Extraktionsschritt nur, wenn Anfragen für ältere Cluster generiert werden, wie es beispielsweise in der nachfolgenden Evaluierung der Fall war. Im normalen Betrieb von NewsSeecr wird hingegen immer mit dem momentanen Zeitpunkt gearbeitet, da hier ausschließlich Inhalte für aktuelle Ereignisse gesucht werden. Da die Zeit nur dazu dient, ein Intervall zur Einschränkung der durchgeführten Suchanfragen zu bestimmen, ist eine genauere Bestimmung des tatsächlichen Ereigniszeitpunkts nicht notwendig.

5.4.2 Anfragegenerierung

Auf Grundlage der im vorangegangenen Schritt extrahierten Informationen werden multifacettierte Suchanfragen generiert. Die hier verwendeten Suchanfragen genügen der folgenden EBNF[111] und sind jeweils und-verknüpft:

```
MultifacetQuery = PhrasePlace, ∧, TimePeriod ;

PhrasePlace = Phrase | Place | Phrase, ∧, Place ;

Phrase = OneGram | TwoGram | ThreeGram | FourGram | FiveGram ;
```

110 Für die Bestimmung der IDF wird die gleiche Vorgehensweise angewendet wie in Abschnitt 5.3.2 beschrieben.
111 Extended Backus–Naur Form

```
Place = PlaceName | PointRadius ;
TimePeriod = StartTime, ∧, EndTime ;
PointRadius = Coordinate, ∧, Radius ;
```

Die Komponente `timePeriod` ist nur als ergänzender Bestandteil für die Suche sinnvoll und wird nicht alleinstehend genutzt. Durch explizite Beschränkung auf einen angegebenen Zeitraum kann eine signifikate Einschränkung der gelieferten Suchresultate erzielt werden. Da die hier gesuchten ereignisbezogenen Inhalte in unmittelbarem zeitlichen Zusammenhang zu dem zugehörigen Ereignis stehen müssen, wird durch diese Beschränkung eine signifikante Menge irrelevanter Inhalte ignoriert. Die Startzeit wird nachfolgend auf drei Tage vor den im vorangegangenen Schritt extrahierten Zeitpunkt festgelegt, die Endzeit entsprechend drei Tage nach dem Zeitpunkt, sodass das zeitliche Intervall jeweils sechs Tage umfasst. Entsprechend muss bei Koordinaten im Rahmen des Anfragebestandteils `PointRadius` ein räumlicher Suchradius `Radius` angegeben werden. Dieser wird auf 10 km um die zugehörige Koordinate `Coordinate` gesetzt. Mit dem Ziel, bei der Suche einen hohen Recall zu erreichen, sind beide dieser Werte bewusst hoch angesetzt.

Basierend auf dem vorangehend dargestellten Schema können die folgenden Kombinationen generiert werden:

- **n-G+T:** n-Gramm-Phrase und Zeitraum
- **P+T:** Ortsname und Zeitraum
- **n-G+P+T:** n-Gramm-Phrase, Ortsname und Zeitraum
- **C+T:** Koordinate mit Radius und Zeitraum
- **n-G+C+T:** n-Gramm-Phrase, Koordinate mit Radius und Zeitraum

Für das Ereignis „Erdbeben auf Bohol, Philippinen“ im Oktober 2013 sehen drei beispielhafte Instanziierungen für Suchanfragen der Kategorien 3-G+T, P+T und C+T wie folgt aus:

```
query1 = text:"7.2 magnitude earthquake" ∧ after:2013-10-12 ∧ before:2013-10-18
query2 = text:"Cebu City" ∧ after:2013-10-12 ∧ before:2013-10-18
query3 = coordinate:9.82,124.19 ∧ radius:10 ∧ after:2013-10-12 ∧ before:2013-10-18
```

Die Anfragegenerierung kombiniert sämtliche möglichen Suchkombinationen auf Basis der extrahierten Informationen. Die Suchanfragen an die betrachteten Quellen für nutzergenerierte Inhalte werden durch individuelle Implementierungen auf die spezifischen REST-Aufrufe abgebildet und die erhaltenen Ergebnisse wiederum in ein einheitliches systeminternes Datenformat übersetzt. Als Grundlage wurden die `WebSearcher` aus dem Palladian-Toolkit (Urbansky, Muthmann, Katz und Reichert, 2012) verwendet, die jedoch ursprünglich nur eine textbasierte Suche erlaubten. Basierend auf den hier beschriebenen Anforderungen wurde deshalb innerhalb des Toolkits das Konzept „Multifacet REST Searcher“ umgesetzt. Hiermit werden Klassen von Such-API-Adaptern abstrahiert, die neben einer reinen textbasierten Suche unterschiedliche miteinander kombinierbare Facetten, wie die hier verwendeten Zeiträume und Koordinaten unterstützen.

5.5 AUSWERTUNG DER ANFRAGEN UND SUCHE

Ziel der Betrachtungen in diesem Abschnitt ist, die Qualität der vorgestellten Anfragekombinationen zu untersuchen. Konkret werden nachfolgend Aussagen zu er Ergiebigkeit und der Precision der einzelnen Anfragetypen auf verschiedenen Quellen getroffen.

Für die durchgeführten Untersuchungen wurden 54 Ereigniscluster auf der NewsSeecr-Datenbank im Zeitraum von Februar 2013 bis Februar 2014 als Benchmarkdatenset ausgewählt, die jeweils mehr als 50 Nachrichtenbeiträgen enthielten. Anhand der im vorherigen Abschnitt beschriebenen Vorgehensweise wurden für sämtliche Cluster 5.843 Suchanfragen in den beschriebenen Kombinationen generiert. Betrachtet werden die folgenden Quellen für nutzergenerierte Inhalte: Flickr, YouTube, Topsy[112] und Instagram. Topsy ist eine Suchmaschine, die Tweets im Gegensatz zu Twitter selbst über einen längeren Zeitraum archiviert[113]. Twitter hingegen bietet, wie sich herausstellte, per Suche lediglich Zugriff auf die letzten Monate und deckt somit nicht sämtliche der hier betrachteten Ereignisse ab. Leider gibt Topsy keine Auskunft, nach welchen Kriterien Tweets archiviert werden und wie hoch der Prozentsatz tatsächlich archivierter Tweets ist. Durchgeführte Analysen legen nahe, dass sich Topsy auf eine Auswahl stark populärer Twitter-Accounts und -Tweets beschränkt, weshalb die hier präsentierten Ergebnisse nur als grobe Simulation einer echten Twitter-Suche betrachtet werden sollten.

Bei der Suche nach Tweets über Topsy werden für die jeweilige Anfrage gegebenenfalls auch Retweets gefunden, also die Weiterverbreitung eines Tweets unter dem eigenen Nutzernamen. Da dies zu einer erheblichen Menge inhaltlicher Duplikate führte, wurden Retweets weitestgehend regelbasiert ausgefilter. Dazu wird die Twitter-Konvention genutzt, dass Retweets mittels „`RT @nutzername`“ gekennzeichnet werden. Da dies jedoch nicht verpflichtend ist, enthalten die Daten von Topsy trotz Filterung einen gewissen Anteil an Duplikaten.

Von den betrachteten Quellen unterstützt nur Flickr sämtliche der beschriebenen Anfragekombinationen. Topsy erlaubt keine koordinatenbasierte Suche, bei YouTube ist eine Suche mittels Koordinaten nicht in Kombination mit einem Zeitintervall möglich[114]. Instagram hingegen unterstützt von den verwendeten Kombinationen ausschließlich C+T und keine Textsuche, Vergleiche zwischen unterschiedlichen Anfragestrategien können hier also nicht durchgeführt werden. Gleichwohl kann ein Vergleich der koordinatenbasierten Suche zur Quelle Flickr gezogen werden.

Die einzelnen Quellen erlauben ein Blättern („Paging“) in den Suchresultaten. Somit besteht theoretisch die Möglichkeit, sämtliche vorhandenen Ergebnisse für eine Suchanfrage zu beziehen. Da

112 `http://topsy.com`

113 Der kostenlose API-Zugriff wurde Mitte April 2014 eingestellt, die hier beschriebenen Experimente wurden vor diesem Zeitpunkt durchgeführt.

114 Version 2.0 der YouTube-API erlaubt eine Suche auf Basis von Koordinaten, unterstützt jedoch nicht die Angabe eines Zeitintervalls, wohingegen Version 3.0 die Beschränkung auf ein Zeitintervall gestattet, jedoch nicht die Verwendung von Geokoordinaten; `https://code.google.com/p/gdata-issues/issues/detail?id=4234`

die einzelnen Quellen jedoch in der Regel Nutzungskontingente für API-Zugriffe festlegen[115], die die Anzahl von Anfragen innerhalb eines Zeitraums limitieren, muss hier eine sinnvolle Grenze festgelegt werden. Es wird davon ausgegangen, dass die relevanten Treffer jeweils an den vorderen Positionen in den Ergebnislisten rangieren. Eine Untersuchung dieser Annahme anhand der hier betrachteten Daten wird nachfolgend vorgenommen. Pro durchgeführter Suchanfrage wird die Anzahl maximal akquirierter Ergebnisse zunächst auf 100 festgelegt.

5.5.1 AUSBEUTE

Die hier verwendeten Quellen stellen „Black Boxes" dar. Angaben darüber, wie viele tatsächlich relevante Ergebnisse für eine Suchanfrage in der Quelle vorhanden sind, können nicht gemacht werden. Mithin ist eine Bestimmung des Recalls (siehe Abschnitt 2.4) nicht möglich.

Der hier verwendete Term „Ausbeute" quantifiziert die Ergiebigkeit des Typs einer Suchanfrage, indem die mittlere Anzahl erzielter Resultate pro Suchanfrage ermittelt wird. Da pro Anfrage maximal 100 Resultate bezogen wurden, stellt eine Ausbeute von 100 das erzielbare Maximum dar. In Tabelle 5.2 sind die Anzahlen der erzielten Treffer für die einzelnen Anfragetypen aufgeschlüsselt. Die Anzahlen der angegebenen Ergebnisse zählen Duplikate mit, die innerhalb des angegebenen Typs mit unterschiedlichen konkreten Suchanfragen gefunden wurden.

Mit dem Anfragetyp 1-G+T (also der Kombination aus 1-Gramm und Zeit) wird auf jeder der betrachteten Quellen mit Werten zwischen 95,04 und 100 % jeweils die höchste Ausbeute erzielt. Die zweithöchste Ausbeute auf Flickr erzielt mit 79,71 % die Kombination P+T (Ortsname und Zeit), wohingegen auf YouTube und Topsy mit 2-Grammen und Zeit (2-G+T) die zweithöchste Ausbeute von 92,55 bzw. 96,2 % erzielt wird. An dritter Stelle rangiert bei Flickr die Kombination C+T aus Koordinate und Zeit, bei YouTube und Topsy die bereits genannte Kombination P+T. Instagram liefert bei der einzig unterstützten Suchanfrage vom Typ C+T 822 Resultate, was einer Ausbeute von 6,18 % entspricht.

Spezifischere Anfragen mit längeren n-Grammen führen, wie zu erwarten, zu geringerer Ausbeute. Ebenso sinkt, wenig überraschend, bei der Hinzunahme des Ortsnamens die Ausbeute im Vergleich zu den Anfragen ohne dieses Kriterium (n-G+T versus n-G+P+T). Besonders stark im Vergleich zu den anderen beiden Quellen zeigt sich dies bei der Quelle Topsy. Die Kombinationen mit n-Grammen und Koordinaten, welche bei den betrachteten Quellen nur von Flickr unterstützt wurden, liefern durchweg eine geringe Ausbeute.

Eine hohe Ausbeute bedingt jedoch keine hohe Precision. Gerade allgemeine Anfragen, die eine hohe Ausbeute liefern, beinhalten im Allgemeinen viele irrelevante Resultate. Dieser Aspekt wird nachfolgend untersucht.

115 Flickr beispielsweise gestattet 3.600 HTTP-Anfragen pro Stunde.

Anfragen		Flickr		YouTube		Topsy		Instagram	
Typ	**#**	**# R.**	**A.**	**# R.**	**A.**	**# R.**	**A.**	**# R.**	**A.**
1-G+T	407	38.683	95,04	40.600	99,75	40.700	100	–	–
2-G+T	157	6.536	41,63	14.530	92,55	15.103	96,2	–	–
3-G+T	96	1.403	14,61	5.710	59,48	5.949	61,97	–	–
4-G+T	41	208	5,07	1.458	35,56	2.383	58,12	–	–
5-G+T	12	36	3	231	19,25	412	34,33	–	–
P+T	265	21.123	79,71	24.099	90,94	24.533	92,58	–	–
1-G+P+T	1.871	73.853	39,47	157.207	84,02	14.302	7,64	–	–
2-G+P+T	660	13.072	19,81	46.222	70,03	2.833	4,29	–	–
3-G+P+T	432	3.335	7,72	15.573	36,05	496	1,15	–	–
4-G+P+T	171	471	2,75	3.395	19,85	5	0	–	–
5-G+P+T	41	79	1,93	701	17,1	0	0	–	–
C+T	133	6.902	51,89	–	–	–	–	822	6,18
1-G+C+T	906	6.813	7,52	–	–	–	–	–	–
2-G+C+T	330	828	2,51	–	–	–	–	–	–
3-G+C+T	215	246	1,14	–	–	–	–	–	–
4-G+C+T	87	1	0,01	–	–	–	–	–	–
5-G+C+T	19	0	0	–	–	–	–	–	–
Gesamt	5.843	173.589	29,71	309.726	74,58	106.716	25,7	822	6,18

Abkürzungen: R. = Resultate, A. = Ausbeute

Tabelle 5.2: Statistiken zu Suchanfragen und Anzahl der erzielten Ergebnisse sowie Ausbeute auf Benchmarkdatenset mit 54 Clustern

5.5.2 PRECISION

Zur Ermittlung der Precision wurden repräsentativ 11.088 Resultate aus dem Benchmarkdatenset ausgewählt. Ziel war, so viele Ergebnisse zu annotieren, dass die Precision pro Anfragetyp mit einem Konfidenzintervall (Witten, Frank und Hall, 2011) von unter 5 % bei einem Konfidenzlevel von 90 % angegeben werden kann. Für einzelne Anfragetypen, die nur eine geringe Ausbeute lieferten und bei denen sich Resultate aus den betreffenden Anfragen auf einige wenige Cluster beschränkten, stellt das gewonnene Datenset nicht genügend Resultate zur Verfügung, weshalb hier keine ausreichend verlässlichen Angaben zur Precision möglich sind (das Konfidenzintervall liegt somit deutlich über 5 %).

Annotation Pro Kombination aus Nachrichtencluster, Anfragetyp und Quelle wurden maximal zehn Einträge nach Zufallsprinzip zur manuellen Annotation bestimmt. Ziel der Annotation war die Bewertung, ob das jeweilige Suchresultat relevante Informationen bezüglich des betrachteten Nachrichtenclusters enthält. Es galten die folgenden Annotationsrichtlinien:

- Bilder müssen das jeweilige Ereignis vor Ort dokumentieren, um als relevant bewertet zu werden. Dieses Kriterium ist nicht erfüllt, wenn Bilder Reaktionen auf Ereignisse an einem anderen Ort darstellen, beispielsweise Solidaritätskundgebungen in New York für Proteste in Kiew.
- Für Videos gilt grundsätzlich Gleiches wie für Bilder. Es wurden Videos als relevant angenommen, die zumindest zu einem Drittel der Spielzeit das zugehörige Ereignis portraitieren. Diese Entscheidung rührt daher, dass viele der Videos, besonders jene, die von Fernsehsendern eingestellt werden, Vor-Ort-Berichte mit Studiomoderation kombinieren.
- Textdokumente (hier Tweets) müssen Informationen über das jeweilige Ereignis beinhalten oder auf solche verlinken (wie beispielsweise in „US senators urge Egypt to release ousted president Mohammed Morsi and other political prisoners http://fb.me/2N8Itm5A8“[116]).

Nicht sämtliche 11.088 der Resultate wurden zweifelsfrei als „relevant“ bzw. „irrelevant“ annotiert. Teilweise konnte der Inhalt der Resultate nicht verstanden werden, da dieser in einer fremden Sprache war. Dies traf vor allem bei Inhalten von YouTube zu, von denen aufgrund dessen 311 Resultate (8,3 %) als „unklar“ markiert wurden und somit in den nachfolgenden Betrachtungen ignoriert werden. Von den Inhalten von Flickr traf dies auf 120 (2,96 %), bei Topsy auf 109 (3,34 %) der Ergebnisse zu. Manche der Resultate waren zum Zeitpunkt der Annotation nicht mehr auf den jeweiligen Plattformen verfügbar, entweder, weil die Inhalte durch die Nutzer selbst oder den jeweiligen Plattformbetreiber entfernt wurden (vor allem bei YouTube geschieht dies häufig bei Urheberrechtsverletzungen). Dies betraf bei YouTube 43 Resultate (1,19 %), bei Flickr 153 (3,77 %) und bei Topsy 10 (0,31 %) Resultate. Diese sind nachfolgend ebenfalls nicht berücksichtigt. Annotiert wurden 3.783 Resultate von Flickr, 3.391 von YouTube, 3.148 von Topsy und 209 von Instagram.

Die Ergebnisse der manuellen Annotation und die daraus ermittelten Werte für die Precision sind in Tabelle 5.3 und Abbildung 5.4 dargestellt. Die nachfolgenden Interpretationen der Ergebnisse berücksichtigen jeweils nur solche Anfrage-Quellen-Kombinationen, bei denen die Precision mit einem Konfidenzintervall von höchstens 5 % ermittelt werden konnte.

Flickr Bei der Quelle Flickr wird die höchste Precision von 72,7 % mit dem Anfragetyp 2-G+P+T erzielt. Grundsätzlich steigert die Hinzunahme eines der Kriterien „Ortsname“ oder „Koordinate“ die Precision jeweils deutlich. Die Kombination 1-G+T bespielsweise erzielt eine Precision von lediglich 10,38 %, die bei Hinzunahme der Ortsnamen (1-G+P+T) auf 41,06 %, bei Hinzunahme der Koordinaten (1-G+C+T) auf 55,12 % gesteigert wird. Die 2-Gramm-basierte Anfrage 2-G+T, welche eine Precision von 41,57 % erzielt, kann durch die Erweiterung um Ortsnamen (2-G+P+T) auf 72,7 % gesteigert werden. Insgesamt erzielen bei Flickr Anfragekombinationen mit Koordinaten eine höhere Precision als jene mit Ortsnamen. Die Ursache kann darin gesucht werden, dass Ortsnamen mehrdeutig sind. Die Anfrage P+T mit Ortsname und Zeit übertrifft mit einer Precision von 24,66 % jene mit Koordinate und Zeit (C+T), welche 21,16 % Precision erzielt.

116 https://twitter.com/theworldobserve/status/367705641407614976

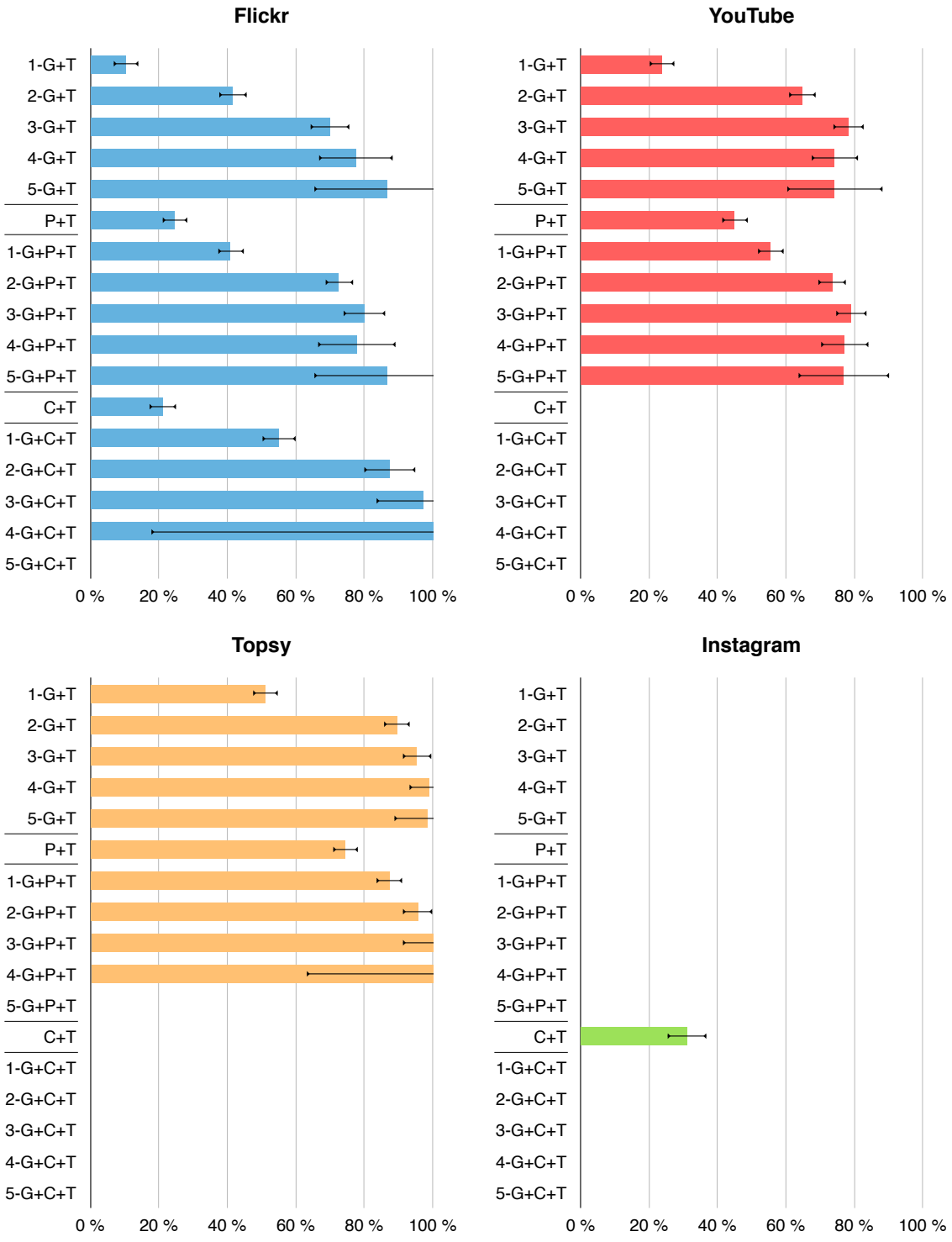


Abbildung 5.4: Precision der einzelnen Suchanfragen für Flickr, YouTube, Topsy und Instagram auf manuell annotiertem Benchmarkdatenset (Konfidenzintervall für ein Konfidenzlevel von 90 %)

	Flickr		YouTube		Topsy		Instagram	
Anfragetyp	**Pr.**	±	**Pr.**	±	**Pr.**	±	**Pr.**	±
1-G+T	**10,38**	3,61	**23,76**	3,66	**51,22**	3,57	–	–
2-G+T	**41,57**	4,01	**64,81**	3,96	**89,53**	3,8	–	–
3-G+T	70,09	5,62	**78,22**	4,4	**95,42**	4,15	–	–
4-G+T	77,59	10,8	74,32	6,76	99,01	5,79	–	–
5-G+T	86,67	21,24	74,29	13,9	98,57	9,83	–	–
P+T	**24,66**	3,64	**45,1**	3,72	**74,62**	3,59	–	–
1-G+P+T	**41,06**	3,71	**55,58**	3,77	**87,37**	3,68	–	–
2-G+P+T	**72,7**	4,1	**73,54**	3,98	**95,6**	4,31	–	–
3-G+P+T	80	5,97	**79,14**	4,4	100	8,58	–	–
4-G+P+T	77,78	11,19	77,3	6,93	100	36,78	–	–
5-G+P+T	86,67	21,24	76,92	13,17	–	–	–	–
C+T	**21,16**	3,97	–	–	–	–	31,10	5,69
1-G+C+T	**55,12**	4,73	–	–	–	–	–	–
2-G+C+T	87,5	7,51	–	–	–	–	–	–
3-G+C+T	97,22	13,71	–	–	–	–	–	–
4-G+C+T	100	82,25	–	–	–	–	–	–
5-G+C+T	–	–	–	–	–	–	–	–
Mittel	**43,97**	1,34	**59,69**	1,41	**82,81**	1,47	31,10	5,69

Abkürzungen: Pr. = Precision, ± = Konfidenzintervall für ein Konfidenzlevel von 90 %

Tabelle 5.3: Statistiken zur Precision der einzelnen Suchanfragen auf manuell annotiertem Benchmarkdatenset; fett sind jene Precision-Werte die im Konfidenzintervall ≤ 5 angegeben werden können, die gemittelten Precision-Werte ergeben sich durch Gleichgewichtung der einzelnen Anfragetypen

YouTube Bei der Quelle YouTube liefert der Anfragetyp 3-G+P+T mit 79,14 % die höchste Precision, die jedoch nur geringfügig höher liegt als das Pendant 3-G+T ohne Ortsname, welches eine Precision von 78,22 % erreicht. Bei den Anfragekombinationen mit kürzeren n-Grammen ist der Gewinn an Precision durch die Hinzunahme des Kriteriums „Ort“ höher; bei 1-Grammen steigt diese von 23,76 % auf 55,58 %, bei 2-Grammen von 64,81 % auf 73,54 %.

Topsy Es fällt auf, dass bei Tweets von Topsy durchweg eine deutlich höhere Precision als bei Flickr und YouTube erzielt wird. Dies lässt sich teilweise dadurch erklären, dass Topsy auf Twitter-spezifische Popularitätsindikatoren wie Retweets und Nutzer-Follower für das Ranking zurückgreifen kann und deshalb mutmaßlich ein Ranking erzielt, das Nachrichtenereignisse gut abzubilden vermag. Ferner ist dies ein Anzeichen für die im zu Beginn geäußerte Vermutung, dass Topsy nur einen kleinen Anteil populärer Tweets abdeckt. Auch hier wird mit dem zusätzlichen Kriterium „Ort“

durchweg eine höhere Precision erzielt als für die Anfragen, die nur aus n-Gramm und Zeit bestehen. Ferner zeigt sich bei Topsy, dass mit der Kombination P+T im Vergleich zu den anderen Quellen eine relativ hohe Precision von 74,62 % erzielt wird.

Instagram Bei Instagram beträgt die Precision für die einzig unterstützte Anfrage vom Typ C+T 31,1 % (65 von 209 annotierten Resultaten wurden mit „relevant“ bewertet, das Konfidenzintervall beträgt somit 5,69 %).

5.5.3 SCHÄTZUNG DER ANZAHL KORREKTER RESULTATE

Wie bereits einleitend dargestellt, können keine Angaben zum Recall gemacht werden, da die Anzahl der tatsächlich in den jeweiligen Quellen vorhandenen Resultate unbekannt ist, womit insbesondere auch kein F1-Maß ermittelt werden kann, welches einen konsolidierten Vergleich der einzelnen Anfragetypen gestatten würde. Stattdessen wird nachfolgend eine Schätzung der absoluten Anzahl korrekter Resultate pro Anfragetyp vorgenommen. Diese Anzahl ermittelt sich wie in Formel 5.9 durch Multiplikation der auf dem manuell annotierten Subset ermittelten Precision und der Anzahl der insgesamt gefundenen eindeutigen Resultate im jeweiligen Subset.

$$numEstimatedCorrect = SubsetPrecision \cdot |Results| \tag{5.9}$$

Die Anzahl der geschätzten korrekt extrahierten Resultate stellt Abbildung 5.5 dar. Die Ergebnisse zeigen bei den Quellen Flickr und YouTube, dass mit den Anfragevarianten n-G+P+T jeweils signifikant mehr korrekte Resultate gefunden werden als mit den Varianten n-G+T ohne Ortskomponente. Die koordinatenbasierten Anfragen, die nur von Flickr unterstützt werden, liefern absolut gesehen weniger korrekte Resultate als die entsprechenden anderen Anfragen. Dies ist der Tatsache geschuldet, dass nur vergleichbar wenige Bilder auf Flickr mit expliziten Geo-Tags versehen sind. Im Gegensatz zu Flickr und YouTube werden bei der Quelle Topsy mit den n-Gramm-Orts-Kombinationen schlechtere Resultate als die Varianten ohne Ortskomponenten erzielt.

5.5.4 RANKINGSCHWELLWERT

Eingangs wurde bereits angegeben, dass zur Erstellung des Benchmarkdatensets pro Suchanfrage maximal 100 Resultate akquiriert wurden. Dieser Wert wurde bewusst hoch gewählt. Um zu bestimmen, ob dieser Schwellwert auch geringer gewählt werden kann, kann ermittelt werden, wie hoch der kumulierte Anteil relevanter Resultate an einem bestimmten Rank ist. Abbildung 5.6 stellt diesen Zusammenhang auf Grundlage des manuell annotierten Benchmarkdatensets dar.

Die durchgezogenen Linien entsprechen dem kumulierten Anteil an Rankingposition R relevanter Ergebnisse im Hinblick auf sämtliche relevanten Einträge an Position $R = 99$. Sie werden mit TP@R (True Positive@R) bezeichnet (siehe Abbildung 2.4). Die gepunktete Linie gibt den Verlauf der

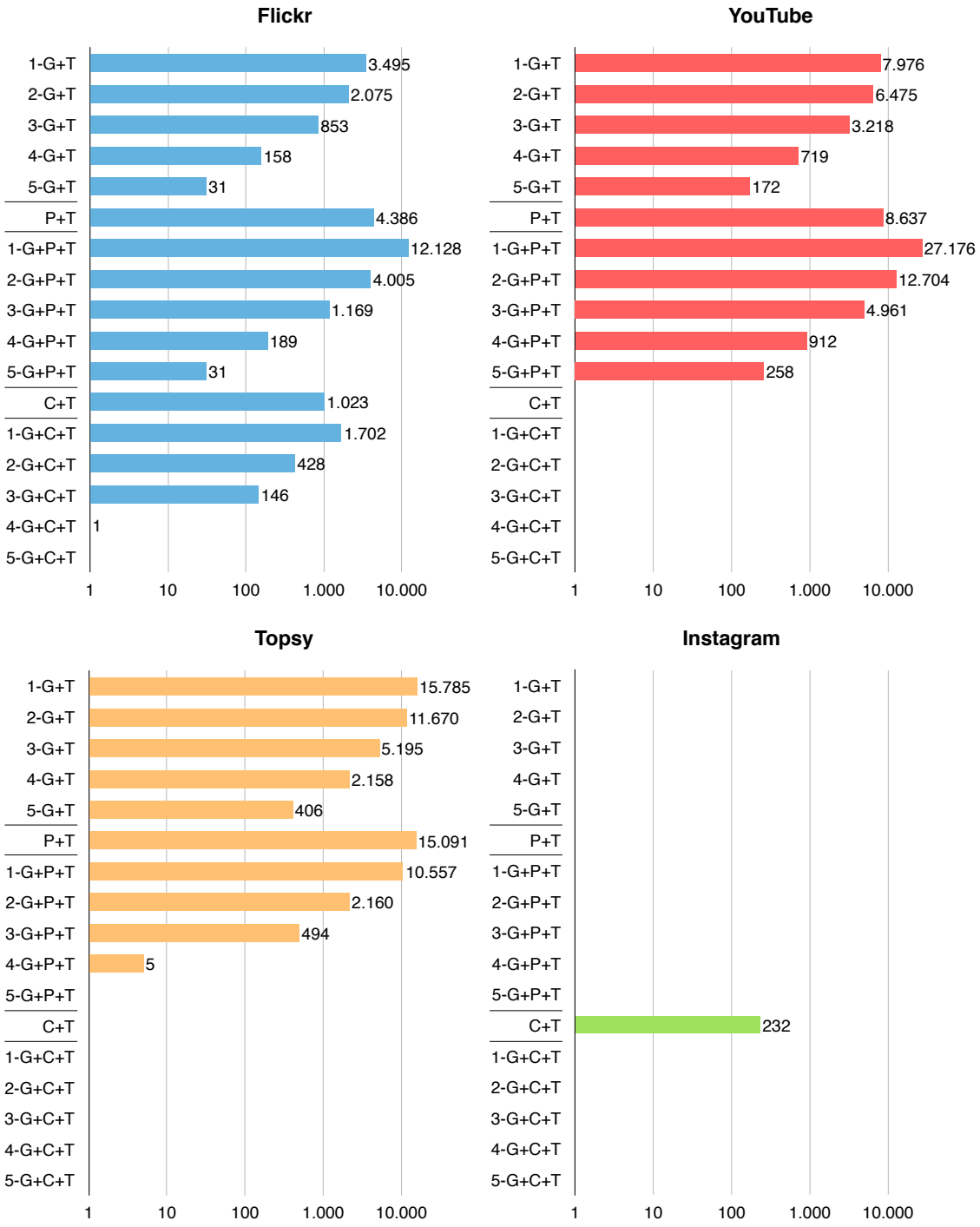


Abbildung 5.5: Geschätzte Anzahl korrekter Resultate für Flickr, YouTube, Topsy und Instagram unter Verwendung des manuell annotierten Benchmarkdatensets

Precision@R an, sie zeigt, wie hoch an einer Rankingposition R der Anteil relevanter Resultate an den Gesamtresultaten an gleicher Position ist.

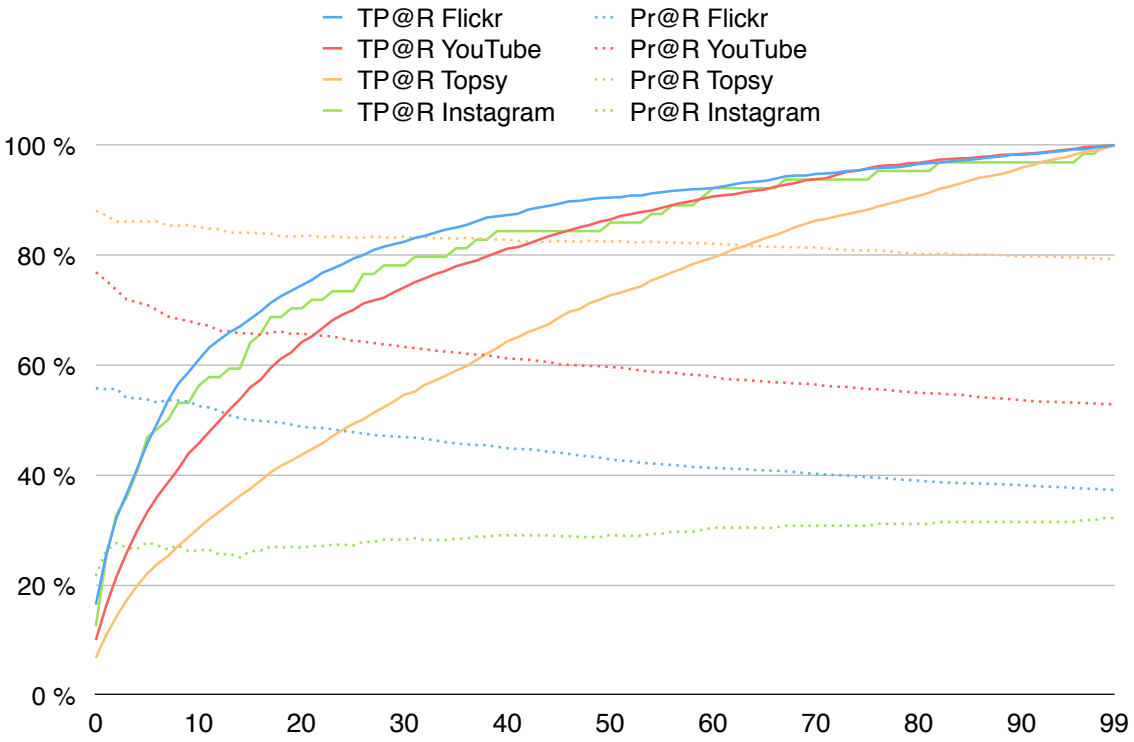


Abbildung 5.6: Verteilung der Resultate auf die Rankingpositionen $R = [0, 99]$

Die Stufen für die Werte von Instagram rühren daher, dass hier deutlich weniger Daten zur Verfügung standen als bei den anderen Quellen (siehe Tabelle 5.2). Da identische Resultate in der Regel durch verschiedene Anfragen und somit auch mit unterschiedlichen Ranks gefunden werden, wurde für mehrfach vorhandene Resultate für diese Analyse jeweils der höchste (also zahlenmäßig geringste) Rank angenommen.

Der Verlauf der kumulierten relevanten Resultate zeigt ein beschränktes Wachsum, bei der der Anteil zusätzlich gewonnener relevanter Ergebnisse mit steigendem Rank abnimmt. Mit dem Ziel, beispielsweise 85 % der ursprünglich relevanten Ergebnisse zu erhalten, kann bei Flickr die Anzahl der Resultate pro Suchanfrage auf 35, bei Instagram auf 50, bei YouTube auf 47 und bei Topsy auf 68 reduziert werden. Dadurch kann die Anzahl notwendiger API-Aufrufe verringert werden. Auf der anderen Seite zeigt der Kurvenverlauf der Precision@R für Flickr, YouTube und Topsy, dass sich bei Reduzierung des Rankingschwellwerts die Precision innerhalb der Ergebnisse erhöht. Bei Flickr beispielsweise beträgt bei einem Schwellwert von 35 die erzielte Precision 45,77 % im Gegensatz zur Precision@99 von 37,29 %.

Bei Topsy ist der Anstieg von TP@R deutlich weniger ausgeprägt als bei den anderen Quellen. Dies manifestiert sich auch im Verlauf der Precision@R, die mit steigendem Rankingschwellwert

weniger abnimmt als bei Flickr und YouTube. Bei Instagram hingegen ist ein leichter Ansteig der Precision@R zu verzeichnen, der auf die nur geringe hier betrachtete Datenmenge zurückzuführen ist.

5.6 RELEVANZFILTERUNG

In Abschnitt 5.5.2 wurde die Precision der unterschiedlichen Suchanfragen auf den betrachteten Quellen untersucht. Bei Gleichgewichtung der einzelnen Anfragen betrug diese 31,1 % für Instagram, 43,97 % für Flickr, 59,69 % für YouTube und 82,81 % für Topsy (siehe Tabelle 5.3). Abgesehen von Topsy sind diese Werte vergleichsweise schlecht und würden in realen Einsatzszenarien an fehlender Nutzerakzeptanz scheitern. Wie in Abbildung 5.4 gezeigt, sind die Unterschiede zwischen den einzelnen Anfragetypen beträchtlich. Eine Möglichkeit, die Gesamtprecision zu steigern, wäre folglich eine Beschränkung auf eine ausgewählte Teilmenge von Anfragetypen, die eine hohe Precision erzielen. Wie in Abbildung 5.5 dargestellt, geht dies jedoch deutlich zu Lasten der Absolutanzahl gefundener Resultate. Eine weitere Möglichkeit, die Ergebnisse akkurater zu machen, besteht, wie im vorhergenden Abschnitt 5.5.4 gezeigt, darin, den Rankingschwellwert so zu justieren, dass weniger, aber dafür prozentual mehr relevante Resultate akquiriert werden.

Eine dritte Möglichkeit zur Steigerung der Precision wird nachfolgend vorgestellt. Die Relevanzfilterung wird hier als Klassifikationsproblem betrachtet und mittels Machine Learning adressiert. In diesem Zuge werden nachfolgend eine Reihe von Features für die Relevanzklassifikation vorgestellt. Mittels gelerntem Random-Forest-Klassifikator werden akquirierte Inhalte in Bezug auf Ereigniscluster als „relevant“ oder „irrelevant“ klassifiziert.

Die nachfolgenden Betrachtungen werden mit den manuell annotierten Daten durchgeführt, die bereits in Abschnitt 5.5.2 Verwendung fanden. Nach Konsolidierung identischer Ergebnisse, die von mehreren Anfragetypen geliefert wurden, stehen 3.391 Resultate von Flickr und 3.177 von YouTube, 3.087 von Topsy und 211 von Instagram für die Experimente zur Verfügung. Die Datensets werden jeweils auf Ereignisbasis in zwei annähernd gleich große disjunkte Trainings- und Testsets aufgeteilt.

5.6.1 FEATURES

Die hier verwendeten Features beziehen sich einerseits auf die verwendeten Suchanfragen (nachfolgend als „Anfragespezifische Features“ bezeichnet), andererseits kommen eine Reihe verschiedener Ähnlichkeitsmaße zwischen dem zu klassifizierenden Inhalt und dem Bezugscluster zum Tragen (nachfolgend „Ähnlichkeitsfeatures zwischen Cluster und Resultat“). In Tabelle 5.4 findet sich die Zusammenfassung der nachfolgend vorgeschlagenen Features.

Bezeichnung	Typ	Beschreibung
Anfragespezifische Features		
`numQueries`	num.	Anzahl der Anfragen, mit denen das Resultat gefunden wurde
`numQueryTypes`	num.	Anzahl der Anfragetypen, mit denen das Resultat gefunden wurde
`foundWith(C)`	bin.	Resultat wurde mit Anfragebestandteil `C` gefunden (z. B. „1-G“)
`minRank`	num.	Minimaler Rank des Resultats
`maxRank`	num.	Maximaler Rank des Resultats
Ähnlichkeitsfeatures zwischen Cluster und Resultat		
`nGramOverlap(n)`	num.	Overlap-Koeffizient aus Zeichen-n-Grammen
`tokenOverlap`	num.	Overlap-Koeffizient aus Tokens
`minTimeDistance`	num.	Minimaler zeitlicher Abstand
`maxTimeDistance`	num.	Maximaler zeitlicher Abstand
`medianTimeDistance`	num.	Median der zeitlichen Abstände
`meanTimeDistance`	num.	Arithmetisches Mittel der zeitlichen Abstände
`mainGeoDistance`	num.	Geographische Distanz zur Hauptlokation
`minGeoDistance`	num.	Minimale geographische Distanz
`maxGeoDistance`	num.	Maximale geographische Distanz
`medianGeoDistance`	num.	Median der geographischen Distanzen
`meanGeoDistance`	num.	Arithmetisches Mittel der geographischen Distanzen

Abkürzungen: bin. = binär, num. = numerisch

Tabelle 5.4: Features für die Klassifikation zur Relevanzbestimmung

Anfragespezifische Features Die anfragespezifischen Features werden aus den Suchanfragen ermittelt, mit denen ein Resultat gefunden wurde. Resultate können durch mehrere Anfragen gefunden werden. Dies wird einerseits durch das Feature `numQueries` ausgedrückt, welches angibt, wie viele Anfragen zu dem betreffenden Resultat geführt haben. Hingegen gibt `numQueryTypes` die Anzahl unterschiedlicher Anfragetypen an, die das Resultat gefunden haben (für ein Ergebnis, das mit den Typen 2-G+T und P+T gefunden wurde, ist der Featurewert somit 2). Das Feature `foundWith(C)` gibt an, ob das Resultat mit einer Anfragekomponente C gefunden wurde (ein Resultat, das mit dem Anfragetyp 2-G+T gefunden wurde, verfügt folglich die beiden binären Features `foundWith(2-G)` und `foundWith(T)`). Der minimale und maximale Rank eines Resultats in sämtlichen akquirierten Ergebnislisten wird anhand von `minRank` und `maxRank` angegeben.

Ähnlichkeitsfeatures zwischen Cluster und Resultat Die zweite Gruppe der extrahierten Features bildet die Ähnlichkeit zwischen dem betrachteten Ereigniscluster und dem den entsprechenden Resultat ab. Es wird die Textähnlichkeit zwischen dem Nachrichtentext des Clusters, in Form der Konkatenation sämtlicher enthaltener Nachrichtendokumente und dem Text des Resultats ermittelt, der aus der Verknüpfung der Felder „Titel“, „Beschreibung“ und „Tags“ repräsentiert wird. Es wurde mit unterschiedlichen Ähnlichkeitsmetriken experimentiert. Neben dem ausgewählten Overlap-Koeffizienten (siehe Formel 5.10) wurden auch die Jaccard- (Jaccard, 1901), Sørensen–Dice-

Koeffizienten (Dice, 1945; Sørensen, 1948) evaluiert. Aufgrund der signifikanten Unterschiede der Textlängen wurden mit dem Overlap-Koeffizienten, der die Größe der Schnittmenge zweier Mengen durch die Größe der kleineren Menge teilt, die besten Ergebnisse erzielt.

$$\text{overlap}(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)} \tag{5.10}$$

Für die Bestimmung der Textähnlichkeit anhand des Overlap-Koeffizienten werden zeichenbasierte n-Gramme für $n = [3, 8]$ sowie einfache Tokens extrahiert.

Die zeitliche Entfernung zwischen einem Resultat und dem Ereigniscluster wird durch die vier Features `minTimeDistance`, `maxTimeDistance`, `medianTimeDistance` und `meanTimeDistance` ausgedrückt. Für diese wird jeweils der Absolutwert der Zeitdifferenz in Stunden zwischen dem Veröffentlichungsdatum des Resultats und den Veröffentlichungsdaten sämtlicher Clusterinhalte betrachtet.

Die geographische Distanz in Kilometern wird in Form des Abstands zwischen der ermittelten Fokuslokation des Clusters und den Koordinaten des Resultats in Form von `mainGeoDistance` ermittelt (siehe Abschnitt 2.1.2). Außerdem werden die vier Features `minGeoDistance`, `maxGeoDistance`, `medianGeoDistance` und `meanGeoDistance` bestimmt, für die sämtliche Clusterlokationen gemäß der Kriterien in Abschnitt 5.4.1 berücksichtigt werden. Für den Fall, dass das betrachtete Resultat keine Koordinatenangaben enthält, wird eine maximale Distanz von 20.037,58 km angenommen (siehe Abschnitt 4.6). Zukünftig bietet es sich hier an, auch auf die Texte der akquirierten Inhalte die hier beschriebenen Lokationsextraktionsmethoden anzuwenden.

5.6.2 Evaluierung der Features

Durch Instanziierung entstehen insgesamt 27 Features. Das Ranking gemäß Information Gain (siehe Abschnitt 2.3.2) ist in Abbildung 5.7 zu sehen.

Als deskriptive Features erweisen sich hier vor allem die textbasierten Ähnlichkeiten, die die ersten sechs Positionen innerhalb des Information-Gain-Rankings einnehmen. Ebenfalls durchgeführt wurde eine Backward Feature Elimination (siehe Abschnitt 2.3.4), um die Featuremenge ohne negative Auswirkungen auf die Accuracy so weit wie möglich zu reduzieren. Selektiert wurden zehn der 27 Features, die in Abbildung 5.8 aufgelistet sind. Analog zu den anderen Klassifikationsaufgaben in dieser Arbeit kam auch hier ein Random-Forest-Klassifikator (siehe Abschnitt 2.3.3) zum Einsatz, mit dem 100 Decision Trees erzeugt wurden. Entsprechend der Beobachtungen aus den vorangehenden Kapiteln zeigt sich auch hier wieder, dass die selektierten Features über das gesamte Spektrum des Information-Gain-Rankings verteilt sind.

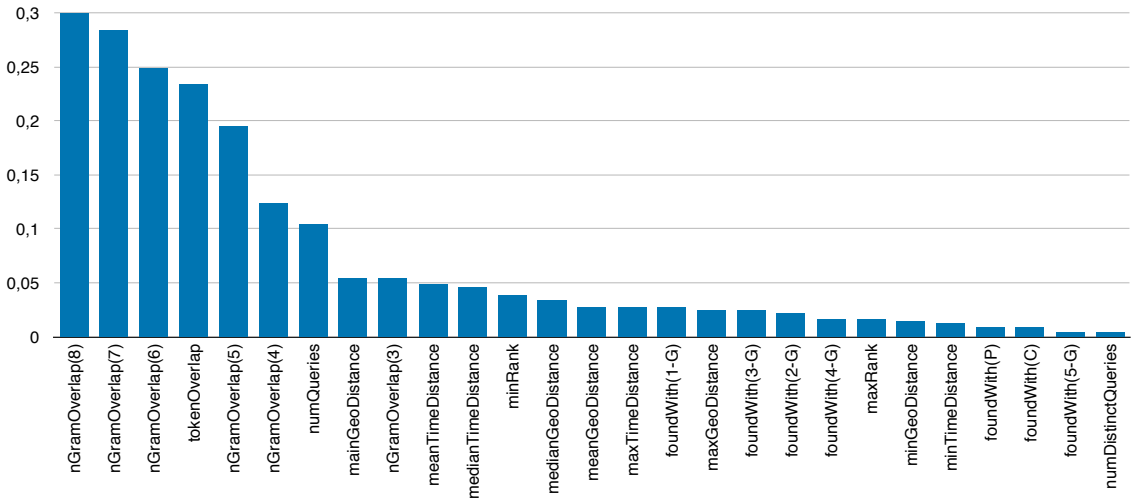


Abbildung 5.7: Information Gain der Features ermittelt auf Traingsset

1. `nGramOverlap(8)`
2. `nGramOverlap(3)`
3. `numQueries`
4. `meanTimeDistance`
5. `nGramOverlap(4)`
6. `maxTimeDistance`
7. `foundWithPlace`
8. `maxRank`
9. `medianGeoDistance`
10. `maxGeoDistance`

Abbildung 5.8: Selektierte Features für die Relevanzklassifikation nach Backward Feature Elimination

5.6.3 EVALUIERUNG DER KLASSIFIKATION

Der aus den Trainingsdaten und mit den zehn ausgewählten Features trainierte Klassifikator wurde zur Relevanzklassifikation des bis hierhin noch ungesehenen Testsets verwendet. Tabelle 5.5 zeigt die Resultate aufgeschlüsselt auf die vier betrachteten Quellen jeweils im Vergleich zur Ausgangssituation ohne Filterung. Der angegebene Recall bezieht sich hier auf die Gesamtmenge des Testsets und ist in der angegebenen Ausgangssituation mithin 100 %, die Accuracy entspricht im Ausgangsfall der Precision. Die minimalen Abweichungen im Vergleich zu den in Tabelle 5.3 unter „Mittel“ aufgeführten Werte für die Precision erklären sich dadurch, dass diese Mengen hier jeweils in Trainings- und Testmenge aufgeteilt und die Werte hier für die Testmenge ermittelt wurden.

Für alle vier Quellen konnte durch die Relevanzklassifikation die Precision in der Ergebnismenge gesteigert werden, ohne überdurchschnittlich den Recall zu beeinträchtigen, was sich in dem jeweils gesteigerten F1-Maß manifestiert. Gerade bei den Quellen Flickr und YouTube mit geringer Ausgangsprecision fällt das Delta in Anbetracht des F1-Maßes in Höhe von 26,57 bzw. 13,65 % deutlich aus. Bei Topsy hingegen, wo bereits in der Ausgangssituation eine vergleichsweise hohe Precision erreicht wird, fällt der Zugewinn in Hinsicht auf den F1-Wert geringer aus. Eine weitere

	Ausgangspunkt				Klassifikator				
Quelle	**Pr.**	**Rc.**	**F1**	**Acc.**	**Pr.**	**Rc.**	**F1**	**Acc.**	**Δ F1**
Flickr	39,74	100	56,88	39,74	78,70	88,81	83,45	86	+26,57
YouTube	58,47	100	73,79	58,49	74,66	90,55	81,84	76,5	+8,05
Topsy	82,05	100	90,14	82,05	94,24	93,79	94,01	90,2	+3,87
Instagram	31,48	100	47,89	31,48	88,89	47,06	61,54	81,48	+13,65

Abkürzungen: Pr. = Precision, Rc. = Recall, Acc. = Accuracy, Δ = Verbesserung in Bezug zur Ausgangsbasis

Tabelle 5.5: Klassifikationsergebnisse zur Relevanzbestimmung im Vergleich mit der Baseline, sowie erzielte Verbesserung im F1-Maß

Verbesserung wäre hier denkbar, indem für jede Quelle ein separater Klassifikator trainiert wird. Experimentell zeigte sich jedoch, dass die weiteren Verbesserungen vergleichsweise gering ausfielen. Desweiteren steht diese Vorgehensweise der Idee einer flexiblen zukünftigen Skalierung des Systems durch Hinzunahme weiterer Quellen im Wege, da hier zunächst Trainingsdaten erzeugt werden müssten.

5.6.4 SCHWELLWERTANALYSE

Die in Tabelle 5.5 gezeigte Auswertung der Klassifikation zeigt nur die Momentanaufnahme, die bei Wahl des Klassifikationsschwellwerts von 0,5 erzielt wird. Durch Verschiebung dieses Schwellwerts können die Ergebnisse mehr in Richtung Precision oder Recall optimiert werden. In Abbildung 5.9 ist die Entwicklung von Precision, Recall und F1-Maß auf dem kombinierten Datenset mit den vier Quellen Flickr, YouTube, Topsy und Instagram dargestellt. Ein Schwellwert von 0 entspricht dem Ausgangspunkt, das heißt, es wird keine Filterung nach Relevanz vorgenommen, was, bezogen auf das betrachtete Datenset, ein Recall von 100 % bedeutet.

Während das Optimum für das F1-Maß für einen Schwellwert von 0,5 erreicht wird, kann unter Verlust an Recall durch Erhöhung des Schwellwerts die Precision gesteigert werden. Beispielsweise wird durch die Wahl eines Schwellwerts von 0,8 eine Precision von über 90 % bei einem Recall von 56,61 % erzielt. Damit wird zwar ein beträchtlicher Anteil relevanter Inhalte „geopfert“, jedoch liegt innerhalb des NewsSeecr-Systems die Priorität auf Korrektheit anstatt hoher Quantitäten an Ergebnissen. Angesichts der großen Mengen verfügbarer nutzergenerierter Inhalte sind Abstriche in Hinblick auf den Recall zu verschmerzen.

5.7 ZUSAMMENFASSUNG

In diesem Kapitel wurde gezeigt, wie Nachrichtenereignisse durch das NewsSeecr-System automatisiert erkannt werden können. Es wurde beschrieben, wie Suchanfragen für nutzergenerierte Inhalte

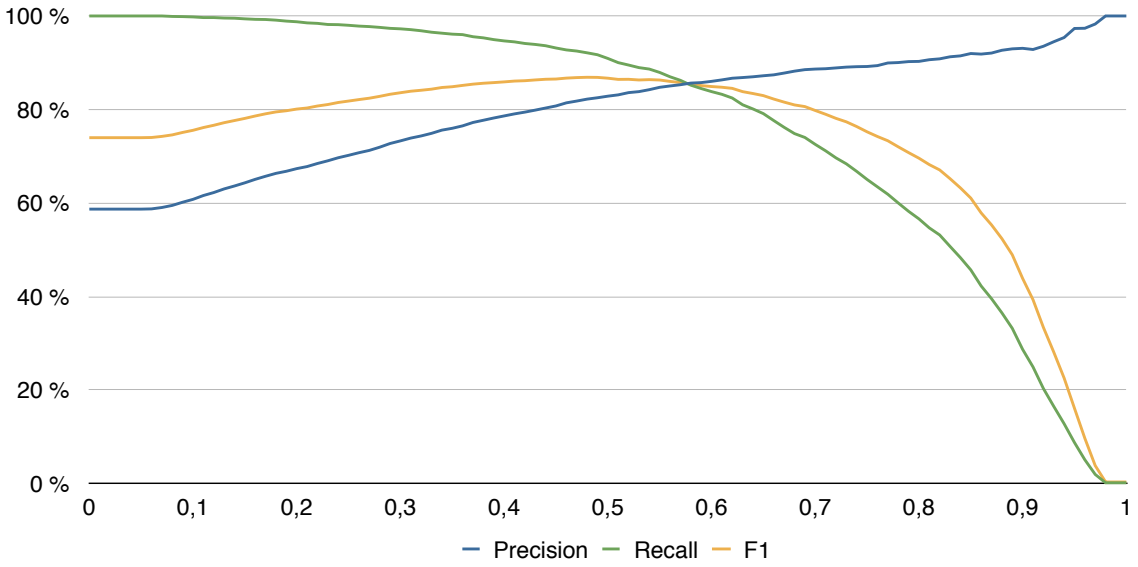


Abbildung 5.9: Precision, Recall und F1-Maß für Schwellwerte im Intervall [0,1] bei der Klassifikation

erzeugt werden, um relevante Beiträge für Nachrichtenereignisse zu aggregieren. Entsprechend des Schwerpunktes dieser Arbeit wurden verschiedene geobasierte Anfragetypen vorgestellt. Die relevanten Komponenten des Gesamtsystems wurden mittels Standardmaßen des Information Retrievals und realistischen Datensets evaluiert.

Die dritte Hypothese dieser Arbeit, die besagte, dass mittels lokationsbasierter Suchanfragen bei der Aggregation ereignisrelevanter nutzergenerierter Inhalte eine höhere Anzahl von Ergebnissen und eine bessere Precision erzielt werden kann, wurde in Abschnitt 5.5 untersucht. Die Ergebnisse, die anhand eines umfangreichen manuell annotierten Datensets über 54 Nachrichtenereignisse gewonnen wurden, zeigten, dass die erzielte Precision durch Suchanfragen mit Lokationsbestandteilen durchweg deutlich über jener liegt, die durch Suchanfragen ohne Lokationsbestandteile erzielt werden. Die absolute Anzahl jeweils gefundener korrekter Resultate ist bei Verwendung lokationsspezifischer Anfragen im direkten Vergleich bei zwei von drei Quellen deutlich höher als bei Anfragen ohne Lokationskomponenten. Verdeutlicht werden soll dies anhand des folgenden Beispiels: Mit den Anfragetypen 3-G+T bzw. 2-G+P+T wird auf Flickr eine Precision von 70,09 bzw. 72,7 % erzielt. Die geschätzte Anzahl gefundener Resultate für das betrachtete Szenario beträgt in erstem Fall jedoch nur 853, wohingegen mit der lokationsbasierten Anfrage 4.005 Ergebnisse gefunden werden. Bei der Quelle Topsy sind hingegen keine Vorteile durch die geobasierten Suchanfragen im Hinblick auf die geschätzte Anzahl korrekter Resultate zu erkennen. Gründe hierfür wurden in Abschnitt 5.5 erläutert.

Mit dem Ziel, die Precision der Suchergebnisse zu steigern, wurde innerhalb von Abschnitt 5.6 eine Machine-Learning-basierte Relevanzfilterung vorgestellt. Unter Verwendung anfrage- und ähnlich-

keitsspezifischer Features und Trainingsdaten wurde ein Klassifikator trainiert, mit dem die Precision der Resultate deutlich verbessert wurde und nur ein vergleichbar geringer Verlust an Recall in Kauf genommen werden musste, was sich in dem gesteigerten F1-Maß im Vergleich zur ungefilterten Ergebnismenge manifestierte. Somit wurde die vierte These dieser Arbeit verifiziert.

6 FAZIT

Dieses Kapitel schließt die vorliegende Arbeit ab. Es wird zunächst eine zusammenfassende Retrospektive über die zentralen Kapitel gegeben. Die in Abschnitt 1.4 vorgestellten Forschungsfragen, welche in dieser Arbeit beantwortet wurden, werden nachfolgend noch einmal aufgegriffen. Neben den Antworten auf die Forschungsfragen werden die Kernbeiträge der Arbeit besprochen und abschließend ein Ausblick mit offenen Problemen gegeben, der mögliche Impulse für zukünftige Forschungsbestrebungen geben soll.

6.1 ZUSAMMENFASSUNG

In Kapitel 1 wurde zunächst die Motivation hinter der vorliegenden Arbeit dargelegt. Der Aspekt „User Generated Content“ im Nachrichtenumfeld bildete den Rahmen für die Arbeit. Mit der gegebenen Definition für den Begriff „Ereignis“ und den Forschungsfragen und Hypothesen wurde der Weg in Richtung geographischer Informationsextraktion geebnet. Das Kapitel schloss mit einer Architekturübersicht über das angestrebte NewsSeecr-Gesamtsystem.

Kapitel 2 gab eine Einführung in die notwendigen Grundlagen und Methoden, die in dieser Arbeit zum Einsatz kamen. Es wurden verwendete geographische Konzepte, Konzepte des Information Retrievals und verwendete Mechanismen des Machine Learnings beschrieben. Abschließend erfolgte eine grundlegende Vorstellung von Evaluierungsmethoden, die für die gesamte Arbeit von Relevanz sind.

Kapitel 3 nahm Bezug auf den geographischen Fokus dieser Arbeit und widmete sich der Extraktion von Lokationen aus unstrukturierten englischen Textdokumenten. Es wurden verwandte Arbeiten und in der Arbeit genutzte Evaluierungsansätze vorgestellt. Der nachfolgende Überblick und die Auswertung ergab, dass die in den verwandten Arbeiten verwendeten Datensets entweder nicht frei verfügbar oder von unzureichender Qualität sind. Aus diesem Grund wurde die Erstellung und die Eigenschaften des Datensets „TUD-Loc-2013“ ausführlich diskutiert. Das Kapitel stellte zwei im Rahmen der Arbeit konzipierte Verfahren zur Lokationsextraktion vor; eines, welches auf einer Reihe von Heuristiken basiert und eines, welches das Problem der Erkennung und Disambiguierung mittels Machine Learning adressiert. In einem abschließenden Vergleich wurden die beiden Verfahren gegen eine Reihe von State-of-the-Art-Systemen evaluiert.

Das daran anschließende Kapitel 4 thematisiert die sogenannte „Fokusbestimmung". Nach einer Diskussion der verwandten Arbeiten wurden insgesamt zehn Strategien zur Fokusbestimmung vorgestellt. Dies umfasst zunächst eine Reihe einfacher Heuristiken, welche bereits in verschiedenen verwandten Arbeiten verwendet wurden. Anschließend wurde eine Machine-Learning-basierte Strategie präsentiert, welche auf einer Reihe konzipierter Features aufbaut. Eine andere Herangehensweise an das Problem wenden die anschließend vorgestellten textklassifikationsbasierten Verfahren zur Fokusbestimmung an. Die erste Strategie ähnelt State-of-the-Art-Ansätzen. Dort identifizierte Schwächen wurden im Rahmen der konzipierten mehrstufigen dictionary-basierten Strategie adressiert. Ein neuer Ansatz ist ferner die vorgestellte k-NN-basierte Strategie. Die Realisierung und Optimierung der vorgestellten Strategien wurde detailliert beschrieben. Anschließend erfolgte eine Evaluierung der zehn Verfahren unter Verwendung von drei Datensets unterschiedlichen Ausmaßes.

Kapitel 5 greift die geschaffenen Konzepte aus den vorangehenden Kapiteln auf und integriert diese in das NewsSeecr-Gesamtsystem. Zunächst werden die Aspekte des Nachrichtenclusterings, mit dem Ziel, relevante Nachrichtenereignisse zu erkennen behandelt. Das Clustering wurde unter Verwendung eines Datensets evaluiert. Daran schließt die Beschreibung der fokussierten Suche nach nutzergenerierten Inhalten an. Es wird beschrieben, wie die verwendeten Anfragen generiert und die dafür notwendigen Informationen extrahiert werden. Für eine exemplarische Auswahl von Nachrichtenclustern wurde ein Datenset erzeugt und manuell annotiert, um die erzielten Resultate zu quantifizieren. Da, wie zu erwarten, die Precision der Ergebnisse vergleichsweise schlecht war, wurde im Anschluss diskutiert, wie mittels per Machine Learning trainierter Klassifikation bei moderaten Einbußen des Recalls die Gesamtprecision deutlich erhöht werden kann.

6.2 BEANTWORTUNG DER FORSCHUNGSFRAGEN

Einleitend wurden vier Forschungsfragen formuliert und daraus Hypothesen abgeleitet. Im Rahmen der Arbeit wurden Konzepte entwickelt und evaluiert, um diese Hypothesen zu verifizieren. Nachfolgend wird auf die einzelnen Forschungsfragen eingegangen.

Welche Methoden eignen sich, um aus unstrukturierten Texten geographische Entitäten zu extrahieren?
Die aus dieser Frage abgeleitete These besagt, dass mittels Machine Learning ein Mechanismus zur Lokationsextraktion aus unstrukturierten englischen Texten geschaffen werden kann, der eine bessere Extraktionsqualität als vergleichbare Systeme erreicht. In Kapitel 3 wurde dieser Mechanismus vorgestellt und unter Verwendung von drei Datensets gegen sechs andere Dienste bzw. Systeme zur Extraktion geographischer Entitäten evaluiert. Die Resultate zeigen, dass die in dieser Arbeit vorgestellte Machine-Learning-basierte Lokationsextraktion State-of-the-Art-Ansätze bezüglich der Erkennung und der Disambiguierung insgesamt deutlich übertrifft. Bei dem in dieser Arbeit vorgestellten TUD-Loc-2013-Datensatz wird bei der Erkennung ein F1-Wert von 77,52 % erzielt.

Dies übertrifft den zweitbesten Ansatz von Alchemy, welcher ein F1-Maß von 74,01 % erzielt. Bei der Lokationsdisambiguierung wird durch das hier vorgestellte Verfahren ein F1-Wert von 72,43 % erreicht, was deutlich über den 62,64 % liegt, die von der Yahoo API erreicht werden.

Welche Methoden eignen sich, um für unstrukturierte Texte den geographischen Fokus zu bestimmen?
Die These, dass mittels Machine Learning ein Ranking extrahierter Lokationen vorgenommen werden kann, um den geographischen Fokus zu bestimmen wurde im Rahmen von Kapitel 4 evaluiert. Die Ergebnisse, welche unter Verwendung des TUD-Loc-2013- und des Wikipedia-Datensets ermittelt wurden, bestätigen diese These. Im Hinblick auf den Median- und den Durchschnittsfehler wurden mit dem vorgestellten Machine-Learning-basierten Ranking die besten Resultate erzielt.

Wie können mit hohem Recall ereignisrelevante Informationen in Quellen für nutzergenerierte Inhalte gefunden werden?
Aus dieser Forschungsfrage wurde die These abgeleitet, dass Suchanfragen mit lokationsspezifischen Bestandteilen bei der Aggregation nachrichtenrelevanter nutzergenerierter Inhalte eine höhere Ausbeute und bessere Precision liefern als Anfragen ohne jene Bestandteile. In Kapitel 5 wurden dazu verschiedene Anfragetypen vorgestellt und für die Suche nach User Generated Content für eine Auswahl an 54 Referenzclustern in den vier Quellen Flickr, YouTube, Topsy und Instagram genutzt. Die Ergebnisse wurden in einem aufwendigen manuellen Annotationsprozess bezüglich ihrer Relevanz für das jeweilige Cluster-Thema bewertet. Eine der betrachteten Quellen (Instagram) unterstützt im Rahmen der dargestellten Bedingungen ausschließlich die Suche mit Koordinaten, weshalb die These implizit erfüllt ist. Bei den Quellen Flickr und YouTube wurden mit geobasierten Anfragen deutlich mehr Resultate gefunden. Bei Topsy war dies nicht der Fall, die Ursachen dafür wurden diskutiert. Die Hypothese ist somit nicht vollumfänglich erfüllt, trotzdem zeigen die Ergebnisse, dass lokationsspezifische Suchanfragen signifikante Verbesserungen bezüglich der Anzahl gefundener Resultate und der erzielten Precision erlauben.

Wie kann für die gefundenen Informationen eine hohe Precision im Hinblick auf die Relevanz für das Ereignis erreicht werden?
Die aus der vorangehend untersuchten Forschungsfrage gewonnenen Resultate zeigen, dass die Gesamtprecision, die bei der Suche nach ereignisrelevanten nutzergenerierten Inhalten in den betrachteten Quellen erzielt wurde, vergleichsweise gering ist. Die im Vorfeld aufgestellte Hypothese besagte, dass diese niedrige Precision durch mittels Machine Learning trainierten Klassifikationsmechanismen verbessert werden kann, ohne dabei zu viele tatsächlich relevante Inhalte zu „opfern". Die Methodik zur Klassifikation und das vorangegangene Feature Mining sind in Abschnitt 5.6 beschrieben. Die erzielten Ergebnisse zeigen, dass durchweg eine Verbesserung des F1-Werts erzielt werden kann.

6.3 WEITERE BEITRÄGE

Neben den grundlegenden Kernaspekten dieser Arbeit, die sich in Antworten auf die aufgestellten Forschungsfragen finden, wurde eine Reihe weiterer wissenschaftlicher und praktischer Beiträge geschaffen, die nachfolgend vorgestellt werden.

Geographische Fokusbestimmung In Kapitel 4 wurden zehn Strategien zur geographischen Fokusbestimmung für Textdokumente vorgestellt. Während die beschriebenen Heuristiken und die Dictionary-basierte Strategie im Wesentlichen auf State-of-the-Art-Ansätzen beruhen, wurden hier drei neue oder deutlich verbesserte Strategien vorgestellt. Es wurde gezeigt, dass mit der mehrstufigen Dictionary-basierten Fokusbestimmung (Abschnitt 4.4.2) bei Trainingssets mittlerer Größe eine signifikante Verbesserung im Vergleich zu einem State-of-the-Art-Ansatz mit einfachem Raster erzielt werden kann. In keiner anderen bisherigen Arbeit wurde bis dato eine umfangreiche Gesamtevaluierung durchgeführt, die unterschiedliche Strategien unter Verwendung verschiedener Datensets miteinander vergleicht.

Palladian Ein Großteil der hier beschriebenen Methoden und Algorithmen wurde innerhalb des Palladian Toolkits implementiert (Urbansky, Muthmann, Katz und Reichert, 2012). Palladian ist eine für die Forschung frei nutzbare Java-basierte Software-Bibliothek[117], die mit dem Ziel entwickelt wurde, Algorithmen des Text Minings, Machine Learnings, der Informationsextraktion und der Suche der Forschung verfügbar zu machen. Die Beiträge und Verbesserungen, die während der Dissertation geschaffen wurden, umfassen vor allem die beschriebenen geobasierten Extraktionsmechanismen, die Textklassifikation und den Zugriff auf verschiedene Web-APIs zur Suche. Die Qualität und Stabilität der Implementierung geht dabei weit über die üblichen wissenschaftlichen Prototypen hinaus, sodass eine unmittelbare Nachnutzung der hier präsentierten Algorithmen möglich wird und somit zukünftige Forschungen unter Verwendung von Palladian auf einem deutlich höheren Niveau als zuvor beginnen können.

Palladian KNIME Nodes Mit dem Ziel, die Funktionalitäten aus Palladian einer breiteren Nutzergruppe – insbesondere Nicht-Entwicklern – zugänglich zu machen, wurden durch den Autor dieser Arbeit die Palladian KNIME Nodes entwickelt[118]. KNIME (Berthold et al., 2007) ist ein workflowbasiertes, grafisches Werkzeug für Data Mining und Analyse, welches ursprünglich hauptsächlich im Bio- und Life-Science-Bereich eingesetzt wurde, aber mittlerweile auch in anderen Domänen Fuß fasst. Die Palladian KNIME Nodes bieten eine Reihe relevanter Funktionen von Palladian über eine nutzerfreundliche Oberfläche an und erfreuen sich wachsender Beliebtheit[119]. Seit nunmehr über drei Jahren findet ein stetiger Kontakt und Austausch mit der KNIME-Community im Rahmen verschiedener Konferenzen statt.

117 http://palladian.ws
118 http://tech.knime.org/community/palladian
119 http://tech.knime.org/community-contributions-downloads

TUD-Loc-2013 Wie die Recherche in Abschnitt 3.4 ergab, existierten keine frei verfügbaren Datensets guter Qualität zur Evaluierung von Lokationsextraktionsverfahren. Mit TUD-Loc-2013, dessen Erstellung in Abschnitt 3.4.2 ausführlich beschrieben ist, wird diese Lücke geschlossen. Das Datenset wurde mit beträchtlichem Aufwand komplett manuell annotiert und ist über die Open-Research-Plattform Areca frei verfügbar[120], womit zukünftige Arbeiten auf diesem Bereich mit den hier präsentierten Ergebnissen verglichen werden können.

NewsSeecr-API Funktionalitäten des hier vorgestellten Systems NewsSeecr sind mittels REST-API[121] nutzbar. Dies umfasst die vorgestellte Lokationsextraktion und Fokusbestimmung (Kapitel 3 und 4). Außerdem bietet die API Zugriff auf aggregierte Nachrichtenereignisse und Inhalte wie in Kapitel 5 beschrieben.

6.4 AUSBLICK

Die mit dieser Arbeit verbundene Forschung hat eine Reihe von Fragen aufgeworfen, die außerhalb des Fokus lagen und deshalb hier nicht beantwortet werden konnten. Diese Fragen liefern jedoch wertvolle Impulse für zukünftige Forschungsbestrebungen und sollen deshalb nachfolgend diskutiert werden.

Die Frage nach der Glaubwürdigkeit der betrachteten Informationen wurde von Beginn an ausgespart. Wohlwissend, dass gerade aufgrund der wachsenden Popularität von nutzergenerierten Inhalten neue Möglichkeiten der gezielten Beeinflussung und der bewussten Streuung von Falschinformationen entstehen, müssen zukünftige Arbeiten deshalb untersuchen, wie die Glaubwürdigkeit von Quellen, Nutzern und einzelnen Inhalten ermittelt und bewertet werden kann.

Bei bestimmten Arten von Nachrichtenereignissen kann eine Betrachtung der zugrundeliegenden Dynamik hochinteressante Einblicke liefern. Die vorgestellten Konzepte zur Extraktion von Lokationen können in Kombination mit einer stärkeren Berücksichtigung des zeitlichen Aspekts die Grundlage liefern, um Entwicklungen von Ereignissen genauer auszuwerten. Beispielsweise könnte analysiert werden, welche Regionen zu welchem Zeitpunkt von Naturkatastrophen wie Bränden oder Erdbeben betroffen sind und wie diese sich ausbreiten. Zukünftige Entwicklungen im Bereich der Computer Vision können in diesem Zusammenhang herangezogen werden, um Inhalte von Bildern und Videos einzuordnen, ohne dass dafür Textinformationen vorhanden sein müssen.

Auch wenn die hier vorgenommenen Betrachtungen primär mit dem Fokus auf die Domäne „Nachrichten“ erfolgten, sind die grundlegenden Konzepte auf andere Domänen übertragbar. Lokationsextraktion und die zielgerichtete Aggregation nutzergenerierter Inhalte ist nicht nur für den Nachrichtenbereich interessant. Gerade im Social-Media-nahen Umfeld werden Quellen und Plattformen für nutzergenerierte Inhalte durch Nutzer früh adaptiert und intensiv genutzt. Im Veranstaltungsbereich, beispielsweise für Konzerte oder Konferenzen, könnten Inhalte, die von Teilnehmern erstellt

120 http://areca.co/21/TUD-Loc-2013-location-extraction-and-toponym-disambiguation-dataset
121 https://www.mashape.com/qqilihq/newsseecr

und veröffentlicht wurden, aggregiert und für eine (semi)automatisierte Berichterstattung genutzt werden.

Die Flut an nutzergenerierten Inhalten wird auch in Zukunft weiter wachsen. Gleichzeitig zeigen aktuelle Trends, dass sich nach der Ära Facebook gerade jüngere Nutzer zunehmend spezialisierten sozialen Netzwerken zuwenden (Matthews, 2014). Im Zuge der steigenden Informationsmenge und der gleichzeitigen Fluktuation und Streuung von Inhalten auf unterschiedliche Plattformen wird der semantischen Verknüpfung von Informationen in den kommenden Jahren eine große Bedeutung zukommen.

A ABBILDUNGEN DER LOKATIONSTYPEN

Feature Class	Type	Zieltyp
A	PCL	COUNTRY
A	PCLF	COUNTRY
A	PCLH	COUNTRY
A	PCLI	COUNTRY
A	PCLIX	COUNTRY
A	PCLS	COUNTRY
A		UNIT
H		LANDMARK
L	AREA	REGION
L	COLF	REGION
L	CONT	CONTINENT
L	RGN	REGION
L	RGNE	REGION
L	RGNH	REGION
L	RGNL	REGION
L		POI
P		CITY
R		POI
S		POI
T		LANDMARK
U	BDLU	REGION
U	PLNU	REGION
U	PRVU	REGION
U		LANDMARK
V		POI

Tabelle A.1: Typabbildungen für GeoNames

Ursprungstyp	Zieltyp
Admin	UNIT
Admin2	UNIT
Admin3	UNIT
Airport	POI
Colloquial	UNDETERMINED
Continent	CONTINENT
Country	COUNTRY
County	UNIT
Drainage	LANDMARK
Estate	POI
HistoricalCounty	COUNTRY
HistoricalTown	CITY
Island	LANDMARK
LandFeature	LANDMARK
LocalAdmin	UNIT
Ocean	LANDMARK
POI	POI
Postal Code	ZIP
Sea	LANDMARK
State	UNIT
Suburb	UNIT
Supername	REGION
Time Zone	UNDETERMINED
Town	CITY
Zip	ZIP

Tabelle A.2: Typabbildungen für Yahoo! BOSS Geo Services

Ursprungstyp	Zieltyp
city	CITY
country	COUNTRY
facility	POI
geographicfeature	LANDMARK
region	REGION
stateorcounty	UNIT

Tabelle A.3: Typabbildungen für AlchemyAPI

Ursprungstyp	Zieltyp
continent	CONTINENT
city	CITY
country	COUNTRY
facility	POI
naturalfeature	LANDMARK
provinceorstate	UNIT
region	REGION

Tabelle A.4: Typabbildungen für OpenCalais

Ursprungstyp	Zieltyp
ADDRESS	UNDETERMINED
BRIDGE	POI
BUILDING	POI
AIRPORT	POI
CANAL	LANDMARK
CITY	CITY
CONTINENT	CONTINENT
COUNTRY	COUNTRY
COUNTY	UNIT
EDUCATIONAL_ORG	POI
GULF	LANDMARK
HEMISPHERE	REGION
HOSPITAL	POI
INTERNATIONAL_REGION	REGION
ISLAND	LANDMARK
LAKE	LANDMARK
LAND_REGION	REGION
LOCATION	UNDETERMINED
MEDICAL_FACILITY	POI
MOUNTAIN	LANDMARK
MOUNTAINRANGE	LANDMARK
OCEAN	LANDMARK
PLANET	REGION
RESTAURANT	POI
RIVER	LANDMARK
ROAD	STREET
SEA	LANDMARK
SETTLEMENT	REGION
STADIUM	POI
UNIVERSITY	POI
US_STATE	UNIT
VALLEY	LANDMARK
WATER_BODY	LANDMARK
WORLDHERITAGESITE	LANDMARK

Tabelle A.5: Typabbildungen für Extractiv

B OPTIMIERUNG DER FOKUSBESTIMMUNG

Features	Modell-größe	$p(d \leq x)$, x in km				Fehler in km	
		1	10	100	1.000	$\overline{d}$	$\widetilde{d}$
1-W	180.091	0,07	1,49	21,32	73,26	1.964,82	221,86
1…2-W	563.350	0,09	1,45	22,06	74,83	1.772,06	214,74
1…3-W	857.033	0,09	1,4	22,01	74,23	1.779,51	218,12
1…4-W	1.064.272	0,09	1,39	21,6	73,54	1.824,1	223,76
1…5-W	1.209.646	0,09	1,39	21,3	72,96	1.853,3	229,32
2-W	841.046	0,09	**1,5**	**22,33**	**75,45**	**1.750,86**	**214,35**
2…3-W	1.464.377	0,09	1,32	21,79	74,7	1.751,86	218,48
2…4-W	1.929.966	0,09	1,28	21,33	74,03	1.784,39	227,37
2…5-W	2.275.668	0,09	1,28	20,94	73,44	1.814,88	232,41
3-W	1.244.195	0,09	1,31	21,72	74,45	1.754,96	222,99
3…4-W	1.958.037	0,09	1,21	20,74	72,98	1.828,83	236,3
3…5-W	2.439.405	0,09	1,2	20,27	72,07	1.885,44	242,9
4-W	1.344.824	**0,1**	1,37	21,26	73,79	1.789,32	228,57
4…5-W	2.055.026	**0,1**	1,29	20,32	71,88	1.891,03	242,54
5-W	1.275.745	**0,1**	1,43	21,36	74,34	1.759,87	227,73

Tabelle B.1: Evaluierung der wortbasierten Featurekombinationen für Dictionary-basierte Strategie

Features	Modell-größe	$p(d \leq x)$, x in km				Fehler in km	
		1	10	100	1.000	$\overline{d}$	$\widetilde{d}$
3-C	58.154	0,04	0,32	7,67	37,81	4.469,62	3.729,6
3…4-C	227.348	0,06	0,78	14,6	55,11	3.410,66	463,01
3…5-C	576.290	0,09	1,38	21,74	73,51	2.124,4	219,56
3…6-C	1.037.169	0,11	1,74	24,88	82,11	1.463,64	193,58
3…7-C	1.512.621	**0,14**	1,87	25,93	85,53	1.218,99	187,27
3…8-C	1.982.922	0,12	1,98	26,08	85,91	1.123,16	186,28
3…9-C	2.439.505	0,12	1,89	25,19	84,51	1.189,76	191,17
3…10-C	2.884.298	**0,14**	1,82	24,32	83,26	1.266,74	197,31
4-C	228.756	0,07	0,78	14,33	54,77	3.370,02	491,1
4…5-C	645.433	0,09	1,29	21,19	72,33	2.176,29	225,21
4…6-C	1.181.752	0,1	1,68	25,03	82,09	1.421,73	193,65
4…7-C	1.720.275	**0,14**	1,93	26,14	85,59	1.205,97	185,45
4…8-C	2.216.405	0,12	1,95	26,34	86,64	1.074,22	183,56
4…9-C	2.688.631	0,12	1,97	26,07	86,02	1.100,55	185,47
4…10-C	3.136.940	0,12	1,86	25,07	84,45	1.181,81	191,91
5-C	632.308	0,09	1,21	19,61	67,73	2.443,12	248,65
5…6-C	1.305.814	0,09	1,65	24,24	80,15	1.560,46	197,44
5…7-C	1.936.342	0,12	1,87	26,12	85,19	1.147,3	184,95
5…8-C	2.492.256	**0,14**	1,97	26,71	**86,99**	1.046,04	181,14
5…9-C	2.971.503	0,12	1,92	26,41	86,75	1.031,98	183,06
5…10-C	3.418.146	0,12	1,95	25,87	85,63	1.085,43	187,01
6-C	1.228.392	0,1	1,47	22,13	74,43	1.914,57	214,4
6…7-C	2.066.330	0,1	1,77	25,59	83,39	1.277,84	189,52
6…8-C	2.722.183	0,12	1,91	26,58	85,91	1.077,04	182,17
6…9-C	3.260.990	0,12	**2**	**26,84**	86,97	**1.022,72**	**180,78**
6…10-C	3.697.156	0,11	1,95	26,4	86,26	1.035,11	183,77
7-C	1.884.737	0,11	1,68	23,58	77,73	1.624,68	203,76
7…8-C	2.827.513	0,11	1,9	26,25	84,63	1.145,03	185,47
7…9-C	3.482.155	0,12	1,92	26,66	86	1.059,58	182,68
7…10-C	3.980.710	0,12	1,96	26,57	86,3	1.051,98	182,69
8-C	2.550.083	0,11	1,74	24,16	79,28	1.504,88	201,07
8…9-C	3.570.958	0,11	1,89	26,23	84,63	1.119,99	186,81
8…10-C	4.199.038	0,11	1,89	26,45	85,43	1.077,18	184,42
9-C	3.217.793	0,11	1,74	24,45	80,1	1.407,98	199,06
9…10-C	4.274.135	0,11	1,9	25,92	84,24	1.144,45	188,94
10-C	3.847.557	0,11	1,74	24,29	80,85	1.351,23	199,39

Tabelle B.2: Evaluierung der zeichenbasierten Featurekombinationen für Dictionary-basierte Strategie

Grid-größe	Belegte Zellen	$p(d \leq x)$, x in km 1	10	100	1.000	Fehler in km $\overline{d}$	$\widetilde{d}$
180	2	0	0	0,36	32,18	3.815,53	1.939,72
90	8	0	0,01	1,38	35,88	2.859,13	1.384,23
45	32	0	0,04	4,31	60,85	1.464,5	748,95
22,5	111	0,02	0,22	5,11	75,34	1.289,58	530,51
11,25	293	0,02	0,54	13,25	**87,54**	1.032,22	317,12
5,63	709	0,12	1,96	22,78	86,85	**1.028,78**	199,19
2,81	1.599	0,22	4,52	45,38	84,36	1.116,44	112,8
1,41	3.179	0,59	8,91	**58,48**	79,36	1.415,46	72,26
0,7	5.653	0,92	12,78	55,92	75,43	1.694,26	**61,52**
0,35	8.704	1,38	16,28	51,4	71,73	1.953,99	85,35
0,18	11.652	1,87	**18,24**	46,86	68,83	2.160,67	138,87
0,09	13.660	**2**	17,43	43,05	66,26	2.357,45	197,65
0,04	14.814	1,9	15,89	40,57	64,6	2.473,7	242,83
0,02	15.483	1,89	15,02	39,57	63,83	2.530,6	258,46
0,01	15.860	1,75	14,72	39,28	63,56	2.551,27	265,93

Tabelle B.3: Evaluierungsergebnisse für Fokusbestimmung mit Wikipedia-Datenset und Dictionary-basierter Strategie bei schrittweise halbierten Grid-Größen

Gridgröße		$p(d \leq x)$, x in km				Fehler in km	
Grob	**Fein**	**1**	**10**	**100**	**1.000**	$\overline{d}$	$\widetilde{d}$
22,5	11,25	0	0	0,12	0,84	1.109,31	334,74
22,5	5,63	0	0,02	0,21	0,84	1.023,01	211,64
11,25	5,63	0	0,02	0,22	**0,88**	939,14	202,03
22,5	2,81	0	0,04	0,43	0,84	975,2	121,3
11,25	2,81	0	0,04	0,44	**0,88**	883,52	116,47
5,63	2,81	0	0,04	0,43	0,87	975,78	121,3
22,5	1,41	0,01	0,08	0,57	0,84	972,59	76,55
11,25	1,41	0,01	0,09	0,59	**0,88**	**869,38**	72,18
5,63	1,41	0,01	0,09	0,58	0,87	954,2	75
2,81	1,41	0,01	0,09	0,6	0,84	1.094,41	69,9
22,5	0,7	0,01	0,12	0,57	0,83	982,95	62,34
11,25	0,7	0,01	0,13	0,59	0,87	870,1	55,15
5,63	0,7	0,01	0,13	0,6	0,87	948,62	57,25
2,81	0,7	0,01	0,13	**0,62**	0,84	1.086,16	50,44
1,41	0,7	0,01	0,13	0,6	0,79	1.407,31	52,05
22,5	0,35	0,01	0,16	0,54	0,83	995,14	70,48
11,25	0,35	0,01	0,17	0,58	0,87	875,71	56,45
5,63	0,35	0,01	0,17	0,58	0,87	950,54	56,26
2,81	0,35	0,01	0,18	**0,62**	0,84	1.084,65	**46,65**
1,41	0,35	0,01	0,18	0,6	0,79	1.404,85	47,23
0,7	0,35	0,01	0,17	0,56	0,75	1.691,8	56,6
22,5	0,18	0,02	0,19	0,51	0,82	1.009,62	88,29
11,25	0,18	0,02	0,2	0,55	0,87	882,72	67,16
5,63	0,18	0,02	0,2	0,57	0,87	954,06	63,61
2,81	0,18	0,02	0,22	0,61	0,84	1.084,57	49,2
1,41	0,18	0,02	0,21	0,6	0,79	1.404,47	48,84
0,7	0,18	0,02	0,21	0,56	0,75	1.691	56,82
0,35	0,18	0,02	0,2	0,51	0,72	1.953,16	84,75
22,5	0,09	0,02	0,19	0,49	0,82	1.021,05	106,95
11,25	0,09	0,02	0,2	0,54	0,87	889,44	77,03
5,63	0,09	0,02	0,2	0,55	0,87	956,74	70,67
2,81	0,09	**0,03**	0,22	0,61	0,84	1.085,72	52,2
1,41	0,09	0,02	**0,22**	0,6	0,79	1.404,52	49,83
0,7	0,09	0,02	**0,22**	0,56	0,75	1.690,85	56,55
0,35	0,09	0,02	0,21	0,51	0,72	1.953,07	85,11
0,18	0,09	0,02	0,19	0,47	0,69	2.160,58	138,87

Tabelle B.4: Evaluierungsergebnisse für zweistufige Fokusbestimmung mit Wikipedia-Datenset für Kombinationen von s im Intervall $[22{,}5°, 0{,}09°]$

Features	$p(d \leq x)$, x in km				Fehler in km	
	1	**10**	**100**	**1.000**	$\overline{d}$	$\widetilde{d}$
1-W	**2,01**	**16**	**49,83**	**79,36**	**1.226,6**	**101,65**
1…2-W	1,81	15,9	49,12	77,06	1.369,5	106,27
1…3-W	1,69	15,2	46,45	73,9	1.650,97	130,25
1…4-W	1,65	14,79	45,18	71,63	1.821,68	148,7
1…5-W	1,75	15,09	45,11	71,33	1.844,53	148,12
2-W	1,76	14,1	45,19	74,56	1.570,21	140,58
2…3-W	1,45	13,59	42,52	70,59	1.939,83	176,26
2…4-W	1,38	13,51	41,29	67,97	2.115,64	200,15
2…5-W	1,4	13,5	40,89	67,67	2.132,38	214,47
3-W	1,39	12,76	40,45	68,72	2.059,86	211,03
3…4-W	1,16	12,63	38,98	66	2.273,43	245,08
3…5-W	1,22	12,56	38,58	64,99	2.343,85	256,57
4-W	1,15	11,98	38,02	65,69	2.266,12	252,4
4…5-W	1,02	11,84	37,36	64	2.413,76	272,55
5-W	0,94	11,23	36,25	62,88	2.683,88	302,92

Tabelle B.5: Evaluierung der wortbasierten Featurekombinationen für k-NN-basierte Strategie

Features	$p(d \leq x)$, x in km				Fehler in km	
	1	**10**	**100**	**1.000**	$\overline{d}$	$\widetilde{d}$
3-C	1,4	13,38	41,25	70,54	1.904,73	192
3…4-C	1,5	13,91	40,95	68,08	2.132,64	208,24
3…5-C	1,48	13,94	39,07	64,41	2.401,33	267,49
3…6-C	1,31	13,29	36,33	60,91	2.713,13	358,42
3…7-C	1,38	13,36	35,97	60,86	2.746,35	353,21
3…8-C	1,28	12,53	35,75	60,35	2.837,07	367,29
3…9-C	1,27	12,53	35,2	59,66	2.851,07	388,65
3…10-C	1,12	12,39	35,01	59,72	2.802,76	391,3
4-C	1,7	15,31	45,17	**73,18**	**1.753,53**	143,95
4…5-C	1,63	14,52	41,19	66,63	2.233,36	211,68
4…6-C	1,57	14,06	38,25	63,04	2.528,53	298,56
4…7-C	1,26	13,54	36,52	61,34	2.710,31	344,86
4…8-C	1,28	13,22	36,07	61,22	2.728,01	345,85
4…9-C	1,24	12,73	35,7	60,51	2.794,52	355,38
4…10-C	1,27	12,64	35,24	60,18	2.815,7	382,23
5-C	**1,82**	**15,81**	**46,28**	72,66	1.809,15	**136,37**
5…6-C	1,52	14,33	41,26	66,23	2.263,04	213,65
5…7-C	1,5	14,3	38,56	62,82	2.515,48	288,63
5…8-C	1,24	13,35	36,27	61,41	2.741,36	344,14
5…9-C	1,34	13,32	36,39	61,38	2.713,67	336,38
5…10-C	1,33	12,85	36,22	61,51	2.725,18	339,97
6-C	1,8	15,66	45,74	71,64	1.895,25	143,67
6…7-C	1,39	14	40,71	65,2	2.373,85	233,54
6…8-C	1,44	13,82	38,67	62,51	2.537,02	283,46
6…9-C	1,33	13,53	37,48	62,35	2.666,82	310,48
6…10-C	1,44	13,48	37,42	62,08	2.685,85	316,68
7-C	1,79	15,27	45,05	70,16	1.978,18	151,67
7…8-C	1,43	13,75	40,29	64,94	2.388,45	238,42
7…9-C	1,42	13,64	38,67	62,55	2.603,15	278,81
7…10-C	1,33	13,42	38,22	62,41	2.630,98	292,86
8-C	1,58	14,74	43,55	69,47	2.023,99	171,41
8…9-C	1,4	13,54	40,35	64,69	2.445,28	243,07
8…10-C	1,32	13,45	38,8	62,77	2.578,72	278,34
9-C	1,47	14,22	42,27	68,14	2.137,72	191,93
9…10-C	1,37	13,32	39,65	64,81	2.410,19	250,27
10-C	1,33	13,68	41,2	67,17	2.211,95	213,22

Tabelle B.6: Evaluierung der zeichenbasierten Featurekombinationen für k-NN-basierte Strategie

PUBLIKATIONSVERZEICHNIS

ZENTRALE PUBLIKATIONEN

1. Philipp Katz und Alexander Schill: **To Learn or to Rule: Two Approaches for Extracting Geographical Information from Unstructured Text**; AusDM / Australasian Data Mining Conference; Canberra, Australien; 11/2013

2. Philipp Katz, Marius Feldmann, Torsten Lunze, Sebastian Sprenger und Alexander Schill: **Authoring Processing Chains for Stream-based Internet Information Retrieval Systems**; BIS 2012, 15th International Conference on Business Information Systems (Akzeptanzrate der Konferenz: 30 %[122]); Vilnius, Litauen; 05/2012

3. Philipp Katz: **Causal Relation Detection for Activities from Heterogeneous Sources**; 11th International Conference on Web Engineering (ICWE 2011); Paphos, Zypern; 06/2011

WEITERE PUBLIKATIONEN

1. Torsten Lunze, Philipp Katz, Dirk Röhrborn und Alexander Schill: **Recommending in an Enterprise Social Media Stream without Explicit User Feedback**; 16. Workshop GeNeMe '13 / Gemeinschaften in Neuen Medien, Dresden; 10/2013

2. Torsten Lunze, Philipp Katz, Dirk Röhrborn und Alexander Schill: **Stream-Based Recommendation for Enterprise Social Media Streams**; BIS 2013, 16th International Conference on Business Information Systems; Poznań, Polen; 06/2013

3. Philipp Katz, Sebastian Sprenger, Marius Feldmann, Eduardo Miranda und Alexander Schill: **A Methodology for Authoring Information Aggregation and Retrieval Workflows in Scalable Infrastructures**; CriMiCo 2012; Sevastopol, Ukraine; 09/2012

4. Christian Knauer, David Urbansky, Johannes Meinecke, Daniel Schuster, Philipp Katz und Alexander Schill: **Semi-Automatic Semantic Lifting of XML to a Target Ontology**; The Joint International Symposium on Natural Language Processing and Agriculture Ontology Service (SNLP-AOS); Bangkok, Thailand; 02/2012

122 http://www.sti2.org/conference-series/conferences-on-business-information-systems

5. Sandro Reichert, David Urbansky, Klemens Muthmann, Philipp Katz, Matthias Wauer und Alexander Schill: **Feeding the World: A Comprehensive Dataset and Analysis of a Real World Snapshot of Web Feeds**; iiWAS2011; Ho Chi Minh City, Vietnam; 12/2011
6. Philipp Katz, Marius Feldmann und Alexander Schill: **Causal Relation Detection for Activities in a Software Development Scenario**; CriMiCo 2011; Sevastopol, Ukraine; 09/2011
7. Christian Liebing, Marius Feldmann, Jan Mosig, Philipp Katz und Alexander Schill: **Tool Support for a Hybrid Development Methodology of Service-based Interactive Applications**; 11th International Conference on Web Engineering (ICWE 2011); Paphos, Zypern; 06/2011
8. Philipp Katz, Torsten Lunze, Marius Feldmann, Dirk Röhrborn und Alexander Schill: **System Architecture for handling the Information Overload in Enterprise Information Aggregation Systems**; BIS 2011, 14th International Conference on Business Information Systems; Poznań, Polen; 06/2011

VORTRÄGE, WORKSHOPS, SONSTIGES

1. David Urbansky, Klemens Muthmann, Philipp Katz und Sandro Reichert: **TUD Palladian Overview**; 10/2011; verfügbar unter `http://palladian.ws`
2. Philipp Katz: **Vortrag: Palladian**; 5th KNIME UGM 2012; Zürich, Schweiz; 02/2012; `http://www.knime.org/UGM2012`
3. Eduardo Miranda, Philipp Katz und David Urbansky: **Vortrag und Workshop: Internet Information Retrieval with Palladian and KNIME**; 6th KNIME UGM 2013; Zürich, Schweiz; 03/2013; `http://www.knime.org/ugm2013`

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

LITERATURVERZEICHNIS

Alex, B. und Grover, C. (2010). Labelling and Spatio-Temporal Grounding of News Events. In *Proceedings of the Workshop on Computational Linguistics in a World of Social Media at NAACL 2010*.

Alexopoulos, P. und Ruiz, C. (2012). Optimizing Geographical Entity and Scope Resolution in Texts using Non-Geographical Semantic Information. In *SEMAPRO 2012, The 6th International Conference on Advances in Semantic Processing* (S. 65–70).

Allan, J. (2002). *Topic Detection and Tracking: Event-based Information Organization*. Norwell, MA, USA: Kluwer Academic Publishers.

Allan, J., Papka, R. und Lavrenko, V. (1998). On-line New Event Detection and Tracking. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (S. 37–45).

Allan, S. und Thorsen, E. (Hrsg.). (2009). *Citizen Journalism: Global Perspectives*. Peter Lang.

Amigó, E., Gonzalo, J., Artiles, J. und Verdejo, F. (2009, August). A comparison of Extrinsic Clustering Evaluation Metrics based on Formal Constraints. *Information Retrieval*, *12* (4), 461–486.

Amitay, E., Har'El, N., Sivan, R. und Soffer, A. (2004). Web-a-Where: Geotagging Web Content. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (S. 273–280).

Andogah, G. (2010). *Geographically Constrained Information Retrieval* (Dissertation). Rijksuniversiteit Groningen.

Andogah, G., Bouma, G., Nerbonne, J. und Koster, E. (2008). Geographical Scope Resolution. In *The Workshop Programme Methodologies and Resources for Processing Spatial Language*.

Assaf, A., Senart, A. und Troncy, R. (2013). SNARC - An Approach for Aggregating and Recommending Contextualized Social Content. In P. Cimiano, M. Fernández, V. Lopez, S. Schlobach und J. Völker (Hrsg.), *The Semantic Web: ESWC 2013 Satellite Events*. Springer Berlin Heidelberg.

Baba, Y., Ishikawa, F. und Honiden, S. (2010). Extraction of Places Related to Flickr Tags. In *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence* (S. 523–528).

Bagga, A. und Baldwin, B. (1998). Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational*

Linguistics and 17th International Conference on Computational Linguistics (Bd. 1, S. 79–85).

Bayes, T. (1763). *An Essay towards solving a Problem in the Doctrine of Chances.*

Becker, H., Naaman, M. und Gravano, L. (2011). Beyond Trending Topics: Real-World Event Identification on Twitter. In *Proceedings of the 5th International Conference on Weblogs and Social Media.*

Benson, E., Haghighi, A. und Barzilay, R. (2011). Event Discovery in Social Media Feeds. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Bd. 1, S. 389–398).

Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., … Wiswedel, B. (2007). KNIME – The Konstanz Information Miner. In C. Preisach, H. Burkhardt, L. Schmidt-Thieme und R. Decker (Hrsg.), *Studies in Classification, Data Analysis, and Knowledge Organization* (S. 319–326). Springer Berlin Heidelberg.

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, *24* (2), 123–140.

Breiman, L. (2001, Januar). Random Forests. *Machine Learning*, *45* (1), 5–32.

Breiman, L., Friedman, J., Stone, C. J. und Olshen, R. A. (1984). *Classification and Regression Trees*. Chapman and Hall.

Buscaldi, D. und Rosso, P. (2008a, März). A conceptual density-based approach for the disambiguation of toponyms. *International Journal of Geographical Information Science*, *22* (3).

Buscaldi, D. und Rosso, P. (2008b). Map-based vs. Knowledge-based Toponym Disambiguation. In *Proceedings of the 2nd International Workshop on Geographic Information Retrieval* (S. 19–22).

Campelo, C. E. C. und de Souza Baptista, C. (2008). Geographic Scope Modeling for Web Documents. In *Proceedings of the 2nd International Workshop on Geographic Information Retrieval.*

Carpenter, B. (2011). *ClusterScore (LingPipe API).* Zugriff auf `http://alias-i.com/lingpipe/docs/api/com/aliasi/cluster/ClusterScore.html`

Chen, E. (2011, April). *Choosing a Machine Learning Classifier.* Zugriff am 29.12.2013 auf `http://blog.echen.me/2011/04/27/choosing-a-machine-learning-classifier/`

Cheng, Z., Caverlee, J. und Lee, K. (2010). You Are Where You Tweet: A Content-Based Approach to Geo-locating Twitter Users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (S. 759–768).

Cieri, C., Graff, D., Liberman, M., Martey, N. und Strassel, S. (2000). Large, Multilingual, Broadcast News Corpora For Cooperative Research in Topic Detection And Tracking: The TDT-2 and TDT-3 Corpus Efforts. In *Proceedings of Language Resources and Evaluation Conference.*

Cohen, W. W. (1996). Learning Trees and Rules with Set-valued Features. In *Proceedings of the 13th National Conference on Artificial Intelligence* (Bd. 1, S. 709–716).

Collins, M. (2002). Ranking Algorithms for Named-Entity Extraction: Boosting and the Voted Perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (S. 489–496).

Crandall, D., Backstrom, L., Huttenlocher, D. und Kleinberg, J. (2009). Mapping the World's Photos. In *Proceedings of the 18th International Conference on World Wide Web* (S. 761–770).

Crowdtap. (2014). *Social Influence: Marketing's New Frontier.* Zugriff auf `http://corp.crowdtap.com/socialinfluence`

da Graça Martins, B. E. (2008). *Geographically Aware Web Text Mining* (Dissertation). Universidade de Lisboa, Faculdade de Ciências, Departamento de Informática.

Dann, S. (2010, 12). *Twitter content classification* (Bd. 15) (Nr. 2). Zugriff am 24.10.2013 auf `http://firstmonday.org/ojs/index.php/fm/article/view/2745/2681`

de la Briandais, R. (1959). File Searching Using Variable Length Keys. In *Papers presented at the the March 3–5, 1959, Western Joint Computer Conference* (S. 295–298).

Diakopoulos, N., Naaman, M. und Kivran-Swaine, F. (2010). Diamonds in the Rough: Social Media Visual Analytics for Journalistic Inquiry. In *2010 IEEE Symposium on Visual Analytics Science and Technology (VAST)* (S. 115–122).

Dice, L. R. (1945, Juli). Measures of the Amount of Ecologic Association Between Species. *Ecology*, *26* (3), 297–302.

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. In *Multiple Classifier Systems* (Bd. 1857, S. 1–15). Springer Berlin Heidelberg.

Ding, J., Gravano, L. und Shivakumar, N. (2000). Computing Geographical Scopes of Web Resources. In *Proceedings of the 26th International Conference on Very Large Data Bases* (S. 545–556).

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S. und Weischedel, R. (2004). The Automatic Content Extraction (ACE) Program Tasks, Data, and Evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation.*

Eisenstein, J., O'Connor, B., Smith, N. A. und Xing, E. P. (2010). A Latent Variable Model for Geographic Lexical Variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (S. 1277–1287).

Elomaa, T. und Kääriäinen, M. (2001, Januar). An Analysis of Reduced Error Pruning. *Journal of Artificial Intelligence Research*, *15*, 163–187.

Etzioni, O., Cafarella, M., Downey, D., Shaked, A.-M. P. T., Soderland, S., Weld, D. S., … Yates, A. (2005). Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *Artificial Intelligence*, *165*, 91–134.

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures* (Dissertation). University of California, Irvine.

Flint, L. N. (1917). *Newspaper Writing in High Schools.* 76 Fifth Avenue, New York City: Noble and Noble.

Frakes, W. B. und Baeza-Yates, R. (1992). *Information Retrieval: Data Structures and Algorithms*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Gale, W. A., Church, K. W. und Yarowsky, D. (1992). One Sense Per Discourse. In *Proceedings of the Workshop on Speech and Natural Language* (S. 233–237).

Garbin, E. und Mani, I. (2005). Disambiguating Toponyms in News. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (S. 363–370).

GeoMidpoint.com. (2007). *Geographic Midpoint Calculation Methods.* Zugriff am 10.10.2013 auf http://www.geomidpoint.com/calculation.html

Graham, M., Hale, S. A. und Gaffney, D. (2013). *Where in the World are You? Geolocation and Language Identification in Twitter.*

Grishman, R. und Sundheim, B. (1996). Message Understanding Conference – 6: A Brief History. In *Proceedings of the 16th International Conference on Computational Linguistics* (Bd. 1, S. 466–471).

Grover, C., Tobin, R., Byrne, K. und Woollard, M. (2009, 11). *Use of the Edinburgh Geoparser in the GeoDigRef and Embedding GeoCrossWalk Projects.*

Guyon, I. und Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Han, B., Cook, P. und Baldwin, T. (2013, August). A Stacking-based Approach to Twitter User Geolocation Prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (S. 7–12).

Hauff, C. und Houben, G.-J. (2012). Geo-Location Estimation of Flickr Images: Social Web Based Enrichment. In R. Baeza-Yates et al. (Hrsg.), *Advances in Information Retrieval* (Bd. 7224). Springer Berlin Heidelberg.

Hays, J. und Efros, A. A. (2008). IM2GPS: estimating geographic information from a single image. In *CVPR*.

He, D. und Wu, D. (2008). Toward a Robust Data Fusion for Document Retrieval. In *International Conference on Natural Language Processing and Knowledge Engineering.*

Hill, A. V. (1910). The possible effects of the aggregation of the molecules of hæmoglobin on its dissociation curves. *The Journal of Physiology*.

Ho, T. K. (1998, August). The Random Subspace Method for Constructing Decision Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20* (8), 832–844.

Hong, L., Ahmed, A., Gurumurthy, S., Smola, A. und Tsioutsiouliklis, K. (2012). Discovering Geographical Topics In The Twitter Stream. In *Proceedings of the 21st International Conference on World Wide Web* (S. 769–778).

Hooper, R. und Paice, C. (2005). *What is Stemming?* Zugriff am 07.01.2014 auf http://www.comp.lancs.ac.uk/computing/research/stemming/general/index.htm

Ikeda, D., Fujiki, T. und Okumura, M. (2006). Automatically Linking News articles to Blog entries. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs.*

Jaccard, P. (1901). Distribution de la flore alpine dans le Bassin des Dranses et dans quelques régions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles*, *37*, 241–272.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*, 11–21.

Jurrat, N. (2011). *Mapping Digital Media: Citizen Journalism and the Internet.* Zugriff am 30.05.2014 auf http://www.opensocietyfoundations.org/reports/mapping-digital-media-citizen-journalism-and-internet

Kelm, P., Schmiedeke, S. und Sikora, T. (2011). A Hierarchical, Multi-modal Approach for Placing Videos on the Map using Millions of Flickr Photographs. In *Proceedings of the 2011 ACM Workshop on Social and Behavioural Networked Media Access* (S. 15–20).

Kruskal, J., Joseph B. (1956). On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, *7* (1).

Kullback, S. und Leibler, R. A. (1951, März). On Information On Sufficiency. *The Annals of Mathematical Statistics*, *22* (1), 79–86.

Laere, O. V., Schockaert, S. und Dhoedt, B. (2010a). Combining Multi-Resolution Evidence for Georeferencing Flickr Images. In *Proceedings of the 4th International Conference on Scalable Uncertainty Management* (S. 347–360). Springer-Verlag.

Laere, O. V., Schockaert, S. und Dhoedt, B. (2010b). Towards Automated Georeferencing of Flickr Photos. In *Proceedings of the 6th Workshop on Geographic Information Retrieval* (S. 5:1–5:7).

Laere, O. V., Schockaert, S. und Dhoedt, B. (2011). Finding locations of Flickr resources using language models and similarity search. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval* (S. 48:1–48:8).

Leidner, J. L. (2006). An evaluation dataset for the toponym resolution task. *Computers, Environment and Urban Systems*, *30* (4), 400–417.

Leidner, J. L. (2007). *Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names* (Dissertation). Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh.

Leidner, J. L., Sinclair, G. und Webber, B. (2003). Grounding spatial named entities for information extraction and question answering. In *Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References* (Bd. 1, S. 31–38).

Li, H., Srihari, R. K., Niu, C. und Li, W. (2002). Location Normalization for Information Extraction. In *Proceedings of the 19th International Conference on Computational Linguistics* (Bd. 1, S. 1–7).

Li, H., Srihari, R. K., Niu, C. und Li, W. (2003). InfoXtract location normalization: a hybrid approach to geographic references in information extraction. In *Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References* (S. 39–44).

Lieberman, M. D. und Samet, H. (2011). Multifaceted Toponym Recognition for Streaming News. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval* (S. 843–852).

Lieberman, M. D. und Samet, H. (2012). Adaptive Context Features for Toponym Resolution in Streaming News. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (S. 731–740).

Lieberman, M. D., Samet, H. und Sankaranarayanan, J. (2010). Geotagging with Local Lexicons to Build Indexes for Textually-Specified Spatial Data. In *Proceedings of the 26th International Conference on Data Engineering* (S. 201–212).

Longueville, B. D., Smith, R. S. und Luraschi, G. (2009). "OMG, from here, I can see the flames!": a use case of mining Location Based Social Networks to acquire spatio-temporal data on forest

fires. In *Proceedings of the 2009 International Workshop on Location Based Social Networks* (S. 73–80).

MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (Bd. 1, S. 281–297).

Mahmud, J., Nichols, J. und Drews, C. (2013). *Home Location Identification of Twitter Users.*

Mani, I., Doran, C., Harris, D., Hitzeman, J., Quimby, R., Richer, J., … Clancy, S. (2009). *SpatialML: Annotation Scheme, Resources, and Evaluation* (Bericht). MITRE.

Manning, C. D., Raghavan, P. und Schütze, H. (2009). *An Introduction to Information Retrieval*. Cambridge University Press.

MapQuest. (2013). Zugriff am 18.12.2013 auf `http://www.mapquest.com`

Marcus, A., Bernstein, M. S., Badar, O., Karger, D. R., Madden, S. und Miller, R. C. (2011). Twitinfo: Aggregating and Visualizing Microblogs for Event Exploration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.*

Markert, K. und Nissim, M. (2002). Towards a Corpus Annotated for Metonymies: the Case of Location Names. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation.*

Martins, B. und Silva, M. J. (2005). A Graph-Ranking Algorithm for Geo-Referencing Documents. In *Proceedings of the fifth IEEE International Conference on Data Mining* (S. 741–744).

Matthews, C. (2014). *More Than 11 Million Young People Have Fled Facebook Since 2011.* Zugriff auf `http://blog.globalwebindex.net/facebook-teens`

McDonald, D. D. (1996). Internal and External Evidence in the Identification and Semantic Categorization of Proper Names. In *Corpus Processing for Lexical Acquisition* (S. 21–39).

McKeown, K. R., Barzilay, R., Evans, D., Hatzivassiloglou, V., Klavans, J. L., Nenkova, A., … Sigelman, S. (2002). Tracking and Summarizing News on a Daily Basis with Columbia's Newsblaster. In *Proceedings of the 2nd International Conference on Human Language Technology Research* (S. 280–285).

Millan, M., Sánchez, D. und Moreno, A. (2008). Unsupervised Web-based Automatic Annotation. In *Proceedings of the 4th STAIRS Conference: Starting AI Researchers' Symposium.*

Miller, G. A. (1995, November). WordNet: A Lexical Database for English. *Communications of the ACM*, *38* (11), 39–41.

Mitchell, T. M. (1997). *Machine Learning.* McGraw-Hill.

Nadeau, D. (2007). *Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision* (Dissertation). Ottawa-Carleton Institute for Computer Science, School of Information Technology and Engineering, University of Ottawa.

Nadeau, D. und Sekine, S. (2007, Januar). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30* (1), 3–26.

Ng, A. Y. und Jordan, M. I. (2002). On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*

(Bd. 14). MIT Press.

Nichols, J., Mahmud, J. und Drews, C. (2012). Summarizing Sporting Events Using Twitter. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces.*

OpenStreetMap. (2013). Zugriff am 23.07.2013 auf `http://www.openstreetmap.org`

Opitz, D. und Maclin, R. (1999). Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, *11*, 169–198.

Overell, S. und Rüger, S. (2008). Using co-occurrence models for placename disambiguation. *International Journal of Geographical Information Science*, *22* (3), 265–287.

Page, L., Brin, S., Motwani, R. und Winograd, T. (1999). The PageRank Citation Ranking: Bringing Order to the Web. In *Proceedings of the 7th International World Wide Web Conference* (S. 161–172).

Parsons, E. (2012, November). *Ed Parsons at Google PinPoint London 2012.* `https://plus.google.com/110553637244873297610/posts/8SYwR5Ze6nB`.

Pearson Scott Foresman. (2007a). *Line art drawing of earth with latitdue parallels.* `http://en.wikipedia.org/wiki/File:Latitude_(PSF).png`.

Pearson Scott Foresman. (2007b). *Line art drawing of earth with longitudes.* `http://en.wikipedia.org/wiki/File:Longitude_(PSF).png`.

Penatti, O. A. B., Li, L. T., Almeida, J. und da S. Torres, R. (2012). A Visual Approach for Video Geocoding using Bag-of-Scenes. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval* (S. 53:1–53:8).

Peng, Y., He, D. und Mao, M. (2006). Geographic Named Entity Disambiguation with Automatic Profile Generation. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence* (S. 522–525).

Peregrino, F. S., Tomás, D. und Llopis, F. (2013). Every Move You Make I'll Be Watching You: Geographical Focus Detection on Twitter. In *Proceedings of the 7th Workshop on Geographic Information Retrieval* (S. 1–8).

Porter, M. F. (2001, 10). *Snowball: A language for stemming algorithms.* Zugriff am 23.10.2013 auf `http://snowball.tartarus.org/texts/introduction.html`

Pouliquen, B., Kimler, M., Steinberger, R., Ignat, C., Oellinger, T., Blackler, K., … Best, C. (2006, Mai). Geocoding Multilingual Texts: Recognition, Disambiguation and Visualisation. In *Proceedings of the 5th International Conference on Language Resources and Evaluation* (S. 53–58).

Priedhorsky, R., Culotta, A. und Valle, S. Y. D. (2014). Inferring the Origin Locations of Tweets with Quantitative Confidence. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing* (S. 1523–1536).

Psallidas, F., Becker, H., Naaman, M. und Gravano, L. (2013). Effective Event Identification in Social Media. *IEEE Data Engineering Bulleting*, *36* (3), 42–50.

Qin, T., Xiao, R., Fang, L., Xie, X. und Zhang, L. (2010). An Efficient Location Extraction Algorithm by Leveraging Web Contextual Information. In *Proceedings of the 18th SIGSPATIAL International*

Conference on Advances in Geographic Information Systems (S. 53–60).

Quinlan, J. R. (1986). Induction of Decison Trees. *Machine Learning*, *1*, 81–106.

Quinlan, J. R. (1987, September). Simplifying Decision Trees. *International Journal of Man-Machine Studies – Special Issue: Knowledge Acquisition for Knowledge-based Systems*, *27* (3), 221–234.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Ratinov, L. und Roth, D. (2009). Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the 13th Conference on Computational Natural Language Learning* (S. 147–155).

Rauch, E., Bukatin, M. und Baker, K. (2003). A confidence-based framework for disambiguating geographic terms. In *Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References of Geographic References* (Bd. 1, S. 50–54).

Reichert, S. (2012). *Analyse und Vorhersage der Aktualisierungen von Web-Feeds* (Dissertation). Technische Universität Dresden, Faculty of Computer Science.

Reichert, S., Urbansky, D., Muthmann, K., Katz, P., Wauer, M. und Schill, A. (2011). Feeding the World: A Comprehensive Dataset and Analysis of a Real World Snapshot of Web Feeds. In *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services* (S. 44–51).

Rennie, J. D. M., Shih, L., Teevan, J. und Karger, D. R. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Proceedings of the 20th International Conference on Machine Learning* (S. 616–623).

Rijsbergen, C. J. V. (1979). *Information Retrieval* (2nd Aufl.). Butterworth-Heinemann.

Roberts, K., Bejan, C. A. und Harabagiu, S. (2010). Toponym Disambiguation Using Events. In *Proceedings of the 23rd International Florida Artificial Intelligence Research Society Conference.*

Roller, S., Speriosu, M., Rallapalli, S., Wing, B. und Baldridge, J. (2012). Supervised Text-based Geolocation Using Language Models on an Adaptive Grid. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (S. 1500–1510).

Sakaki, T., Okazaki, M. und Matsuo, Y. (2010). Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In *Proceedings of the 19th International Conference on World Wide Web* (S. 851–860).

Samet, H., Adelfio, M. D., Fruin, B. C., Lieberman, M. D. und Sankaranarayanan, J. (2013). PhotoStand: A Map Query Interface for a Database of News Photos. In *Proceedings of the 39th Conference on Very Large Data Bases.*

Sanderson, M. und Kohler, J. (2004). Analyzing geographic queries. In *Workshop on Geographic Information Retrieval SIGIR.*

Sang, E. F. T. K. und Meulder, F. D. (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003* (Bd. 4, S. 142–147).

Sankaranarayanan, J., Samet, H., Teitler, B. E., Lieberman, M. D. und Sperling, J. (2009). TwitterStand: News in Tweets. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (S. 42–51).

Serdyukov, P., Murdock, V. und van Zwol, R. (2009). Placing Flickr Photos on a Map. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (S. 484–491).

Sinnott, R. W. (1984). Virtues of the Haversine. *Sky and Telescope*, *68* (2), 159.

Smith, D. A. und Crane, G. (2001). Disambiguating Geographic Names in a Historical Digital Library. In *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries* (S. 127–136).

Smith, D. A. und Mann, G. S. (2003). Bootstrapping toponym classifiers. In *Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References* (Bd. 1, S. 45–49).

Sørensen, T. (1948). A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and its Application to Analyses of the Vegetation on Danish Commons. *Det Kongelige Danske Videnskabernes Selskab, Biologiske Skrifter*, *5* (4), 1–34.

Starbird, K., Palen, L., Hughes, A. L. und Vieweg, S. (2010). Chatter on The Red: What Hazards Threat Reveals about the Social Life of Microblogged Information. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work.*

Steinberger, R., Pouliquen, B. und van der Goot, E. (2009). An introduction to the Europe Media Monitor family of applications. In *Proceedings of the SIGIR 2009 Workshop* (S. 1–8).

Steinhaus, H. (1957). Sur la division des corp matériels en parties. In *Bulletin de l'Académie Polonaise des Sciences* (Bd. 4, S. 801–804).

Teitler, B. E., Lieberman, M. D., Panozzo, D., Sankaranarayanan, J., Samet, H. und Sperling, J. (2008). NewsStand: A New View on News. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.*

Tobin, R., Grover, C., Byrne, K., Reid, J. und Walsh, J. (2010). Evaluation of Georeferencing. In *Proceedings of the 6th Workshop on Geographic Information Retrieval.*

Trevisiol, M., Jégou, H., Delhumeau, J. und Gravier, G. (2013). Retrieving Geo-Location of Videos with a Divide & Conquer Hierarchical Multimodal Approach. In *Proceedings of the 3rd ACM International Conference on Multimedia Retrieval* (S. 1–8).

Tsagkias, M., de Rijke, M. und Weerkamp, W. (2011). Linking online news and social media. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining* (S. 565–574).

Urbansky, D. (2012). *Automatic Extraction and Assessment of Entities from the Web* (Dissertation). Technische Universität Dresden, Faculty of Computer Science.

Urbansky, D., Muthmann, K., Katz, P. und Reichert, S. (2012, 02). *TUD Palladian Overview.*

Vincenty, T. (1975, April). Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations. *Survey Review*, *XXIII* (176), 88–93.

Wang, C., Xie, X., Wang, L., Lu, Y. und Ma, W.-Y. (2005a). Detecting Geographic Locations from

Web Resources. In *Proceedings of the 2005 Workshop on Geographic Information Retrieval* (S. 17–24).

Wang, C., Xie, X., Wang, L., Lu, Y. und Ma, W.-Y. (2005b). Web Resource Geographic Location Classification and Detection. In *Proceedings of the 2005 Workshop on Geographic Information Retrieval.*

Watanabe, K., Ochi, M., Okabe, M. und Onai, R. (2011). Jasmine: A Real-time Local-event Detection System based on Geolocation Information Propagated to Microblogs. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management* (S. 2541–2544).

Weber, A. (1909). *Über den Standort der Industrien, Erster Teil: Reine Theorie des Standortes*. J. C. B. Mohr.

Wing, B. P. und Baldridge, J. (2011). Simple Supervised Document Geolocation with Geodesic Grids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Bd. 1, S. 955–964).

Witten, I. H., Frank, E. und Hall, M. A. (2011). *Data Mining* (Third Edition Aufl.). Morgan Kaufmann.

Zong, W., Wu, D., Sun, A., Lim, E.-P. und Goh, D. H.-L. (2005, Juni). On Assigning Place Names to Geography Related Web Pages. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries* (S. 354–362).